Ph.D. Thesis

UNDERWATER TARGET CLASSIFIER WITH IMPROVED SUCCESS RATE USING META-OPTIMAL SUPPORT VECTOR MACHINES

Submitted to

THE COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

in partial fulfilment of the requirement for the award of the degree of

Doctor of Philosophy

by

SHERIN B. M.

Under the guidance of

Prof. (Dr.) Supriya M. H.

DEPARTMENT OF ELECTRONICS
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
COCHIN – 682 022, INDIA

DECEMBER 2017

UNDERWATER TARGET CLASSIFIER WITH IMPROVED SUCCESS RATE USING META-OPTIMAL SUPPORT VECTOR MACHINES

Ph. D. Thesis in the field of Ocean Electronics

Author

Ms. Sherin B. M.
Research Scholar
Department of Electronics
Cochin University of Science & Technology
Cochin – 682 022, India
e-mail: Sherin.Ben@gmail.com

Research Advisor

Dr. Supriya M. H.
Professor
Department of Electronics
Cochin University of Science And Technology
Cochin,– 682 022, India
e-mail: supriyadoe@gmail.com

December, 2017

Dedicated to.....

My Parents, Husband & Son

Cochin University of Science and Technology

DEPARTMENT OF ELECTRONICS

CERTIFICATE

This is to certify that this thesis entitled, ***Underwater Target Classifier with Improved Success Rate using Meta-Optimal Support Vector Machines*** is a bonafide record of the research work carried out by Ms. Sherin B. M., under my supervision in the Department of Electronics, Cochin University of Science and Technology. The result presented in this thesis or parts of it have not been presented for any other degree(s).

Cochin – 682 022
28th December 2017

Prof. (Dr.) Supriya M. H.
Supervising Guide
Department of Electronics
Cochin University of Science and Technology

DECLARATION

I hereby declare that the work presented in this thesis entitled ***Underwater Target Classifier with Improved Success Rate using Meta-Optimal Support Vector Machines*** is a bonafide record of the research work carried out by me under the supervision of Dr. Supriya M. H., Professor, in the Department of Electronics, Cochin University of Science and Technology. The result presented in this thesis or parts of it have not been presented for other degree(s).

Cochin – 682 022
28th December 2017

SHERIN B. M.
Department of Electronics
Cochin University of Science
And Technology

Cochin University of Science And Technology

DEPARTMENT OF ELECTRONICS

CERTIFICATE

This is to certify that this thesis entitled, ***Underwater Target Classifier with Improved Success Rate using Meta-Optimal Support Vector Machines*** has been modified to effect all the relevant corrections suggested by the Doctoral Committee and the audience during the Pre-synopsis Seminar.

Cochin – 682 022
28th December 2017

Prof. (Dr.) Supriya M. H.
Supervising Guide
Department of Electronics
Cochin University of Science
And Technology

Acknowledgements

*I would like to express my deepest sense of gratitude to my research guide, **Prof. (Dr.) Supriya M. H.**, Professor, Department of Electronics, Cochin University of Science and Technology for her excellent guidance and incessant encouragement. It has been a great pleasure and privilege to work under her and she was always there when I needed help.*

*I am deeply grateful to **Prof. (Dr.) P. R. S. Pillai**, Former Professor & Head, Department of Electronics, Cochin University of Science and Technology, for his wholehearted support and motivation. I would like to express my sincere thanks to Prof. (Dr). K. Vasudevan, Prof. (Dr). P. Mohanan, Prof. (Dr.) Tessamma Thomas, Prof. (Dr.) C. .K. Aanandan and Prof. (Dr.) James Kurian, of the Department of Electronics for rendering me consistent support and encouragement.*

*I take this opportunity to thank all my teachers, mentors and professors at all stages of my education who have inspired and impacted me throughout my career. I take this opportunity also to thank **Dr. Suresh Das**, Executive Vice President, KSCSTE, **Dr. Pradeep Kumar**, Member Secretary, KSCSTE, **Dr. G. M. Nair**, Advisor, Kerala Biotechnology Commission, for their support and kind words of encouragement. I especially remember with gratitude the support and encouragement rendered by **Dr. V. Ajit Prabhu**, Chief Scientist, KSCSTE during my thesis work. A special word of thanks to **Mr. Subash Chandrabose M.R.**, Group Director, Transducer Group, NPOL for his invaluable support during my thesis preparation.*

*I express my deep sense of acknowledgement and gratitude to all the research fellows at the **ASPIRE** lab, Mr. Mohan Kumar K., Mrs. Rithu James, Mr. Suraj Kamal, Mr. Satheesh Chandran, Mr. Kurian Thomas and Mr. Alex Raj for rendering unconditional support and valuable suggestions in my PhD work. A word of thanks is also deserved by my friends, Mrs. Anju K. R., Mrs. Prabha C., and Mr. Adrine Correya and the faculties Mr. Arun A. Balakrishnan, Dr. Bijoy Antony Jose, Mr. Mithun Haridas T.P. and Ms. Limna T.J. for their good wishes and support.*

I thankfully bear in mind the sincere co-operation and support I received from the library and administrative staff of the Department.

I take this opportunity to thank my friends and cousins at my hometown, Trivandrum, especially, Sherine, Remya, Priyanka, Safikh, Sarika, Arjun, Divya, Maya, Greeshma, Naveen, and Prem for their tremendous support and kind words of encouragement.

I owe a lot to my parents ***Mr. Benraj P. S.*** *and* ***Mrs. Mary Hose****, who inspired and supported me at every stage of my personal and academic life, and longed to see this achievement come true. I also gratefully remember my grandmother Late.* ***Mrs. Lizzie Johnson*** *whose prayers has guided me throughout my personal and academic life.*

I also thank my parents-in-law ***Mr. & Mrs. Winslow*** *for their love, prayers and moral support. Thanks are also due to my brother in law and co-sister,* ***Mr & Mrs. Bineesh*** *for their love and good wishes.*

I express my deep gratitude to my uncle and aunt, ***Mr. & Mrs Anil George*** *for providing me comfortable stay at their home, at Cochin during my research period. Thanks are also due to their daughters,* ***Mekha*** *and* ***Varsha****. I was lucky to enjoy their deep love and care, while at Cochin.*

It is beyond words to express my appreciation and gratitude to my husband ***Mr. Aneesh W.*** *who supported me in every possible manner throughout my PhD work. A sweet, lovely bunch of thanks to my little one,* ***Abel S. Aneesh****, whose happiness is indispensable to keep my spirits high.*

Finally, but most importantly, countless and endless thanks to Almighty, the source of all knowledge, the supreme guide, without whose blessings, I would not have written this last note.

Who, like Thyself, my Guide and stay can be ?
Through cloud and sunshine, Lord Abide with me. – Amen

Thank you, Lord, for blessing me with my family, research supervisor, lab mates and friends without whose co-operation and support, I could not have completed this thesis.

SHERIN B.M.

28th December 2017

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Underwater target recognition and classification is a demanding task owing to the dynamic nature of the ocean. Underwater target recognition is accomplished through hydro-acoustic monitoring in which hydrophones detect underwater acoustic signals which are then analysed to get information about the various activities and phenomena occurring in the sea. The dynamic behaviour of the ocean, due to a myriad of factors including the changing acoustic channel properties amidst heavy background noise, alters the characteristics of hydrophone received echoes in a random manner. Nevertheless, the detection and classification of underwater targets of interest concealed and masked by heavy ocean noise is very important, particularly in strategic sectors.

The source-specific features of the acoustic patterns emanating from the targets are patterned by suitable pattern recognition algorithms to generate characteristic acoustic signatures of the targets, commonly referred to as features. Classification algorithms, which are pattern matching techniques, identify individual targets of interest from the hydrophone captured acoustic mixture making use of their individual features. Although the classification task is quite straight-forward, the underwater target classifier is imposed with challenge of classifying data embedded with varying background noise.

Most underwater target recognition applications, especially strategic applications, necessitate fast and accurate underwater target classifiers with little or no manual intervention. This work aims to improve the performance of an Support Vector Machine (SVM) based multi-class underwater target classifier. An aim towards performance improvement of the classifier

necessitates the creation of an adequately descriptive feature vector. Lot many acoustic feature extraction techniques exist for a wide range of applications. It has been found that simple time and spectral features cannot aid in reliable classification in the presence of composite ambient noise and varying oceanic parameters. Nonlinear cepstral techniques are known to be capable of yielding promising features of signals heavily laden with noise. Different time domain, spectral and cepstral based feature extraction techniques are evaluated. A composite feature vector is formed from the different features extracted by applying appropriate feature selection algorithms.

The parameters of the classification algorithms have a profound impact on the classifier performance. The performance of an SVM based classifier is particularly dependent on the kernel function adopted and the kernel parameters. Therefore, an attempt to improve the performance of the SVM based classifier must invariably focus on selecting the optimal algorithmic parameters. Underwater target classification with dynamically changing constraints, necessitates the dynamic selection of the optimal algorithmic parameters, kernel function and kernel parameters. The optimal parameters which give the best performance can be found from the search space using meta–heuristic algorithms. Five meta-heuristic algorithms have been implemented to automatically find the optimal parameters of the classifier, which has shown to improve the classifier performance. It has been found that the Symbiotic Organisms Search (SOS) algorithm gives better performance compared to other evaluated techniques. Certain modifications are proposed to the SOS algorithm, and has been named as *m*-SOS algorithm. The proposed *m*-SOS algorithm has been found to give superior performance, compared to other algorithms evaluated.

CHAPTER 1

INTRODUCTION

SONAR, short for SOund Navigation And Ranging, are devices that use sound propagation to navigate, communicate, detect and locate objects underwater. Since its introduction during the early half of the 20th century, it has been undergoing various evolutionary stages and underwater research has remained as one of the priority areas of research. They are used to explore and gain insight into the underwater activities and hence play an integral role in ocean research. Research on underwater target recognition has remained a priority in many countries since the early half of 20th century when sonars were introduced, due to its strategic importance. However, underwater target recognition is complicated by a myriad of factors including the varying underwater channel characteristics and the dynamic noise background of the ocean. This chapter briefly describes sonars and various noise sources in the ocean. The chapter also throws light on the applications of underwater target recognition and the challenges associated with it. The underlying principle of operation of the proposed classifier including optimisation of the classifier for improved performance is also presented.

1.1 *Background*

India is a country with its southern region projecting onto the Indian Ocean and has an appreciably large coastline of over 7,500 kilometres. India's maritime security has a large bearing on the National Security.

Research for improving the efficiency and performance of acoustic target classifiers to detect any potential enemy vessel in the waters are thus very important in the National perspective.

The sound is the most reliable medium which reveals to us the deep dark ocean. Non-acoustic waves like electromagnetic waves have reserved underwater applications, as they are heavily attenuated due to the high conductivity of water. Sound, on the other hand, depending on its frequency can propagate several kilometres underwater. Hence, hydro-acoustic monitoring is used to gather information on the hidden activities in the sea. Underwater hydrophones detect underwater acoustic signals which may then be analyzed to get insight into the ocean noise, natural underwater biological phenomenon and also for strategic applications. SONAR, an acronym for **SO**und **NA**vigation and **R**anging, are systems employing sound propagation to communicate and unveil underwater biological and man-made phenomena and systems utilising this principle referred to as sonar systems [1]. In the strategic sector, sonars are used for detection, classification, localisation of submarines and mines as well as for communication and navigation purposes. In commercial applications, sonars are used in fish finders, material inspection and seismic exploration. Sonar systems can be broadly categorized into active and passive sonar.

1.2 *Active Sonar*

Sonar systems are said to be active, when sound waves generated by an acoustic projector is reflected back as sonar echoes by a target object, to a hydrophone, which converts sound into electricity. The projector consists of an electro-acoustic transducer with associated modules such as a signal generator and power amplifier, and the hydrophone is a piezoelectric transducer which captures sound and converts to electricity. Sonar systems

can be configured either in the monostatic or bistatic configuration. In the monostatic configuration, the transmitter and receiver are co-located whereas in the bistatic configuration, the transmitter and receiver are spatially separated by a distance comparable to the range of the target. An extension of the bistatic configuration is multistatic configuration, which is a multi-node system with more than one projector, receiver or both. Most sonars are configured monostatically, with the same transducer array being used for both transmission and reception. Sonar systems for long range surveillance and buried object detection are configured multistatically.

The fundamental principle of an active sonar is shown in Figure 1.1. A sonar projector emits pulses of sound often referred to as 'ping' which on hitting a target, is reflected back as echoes. A beamformer is usually employed, to concentrate the acoustic power in the return echoes into a beam, which is then listened to and analysed, to detect the presence of a target. The analysis may also be extended to determine the range and bearing of an underwater target.

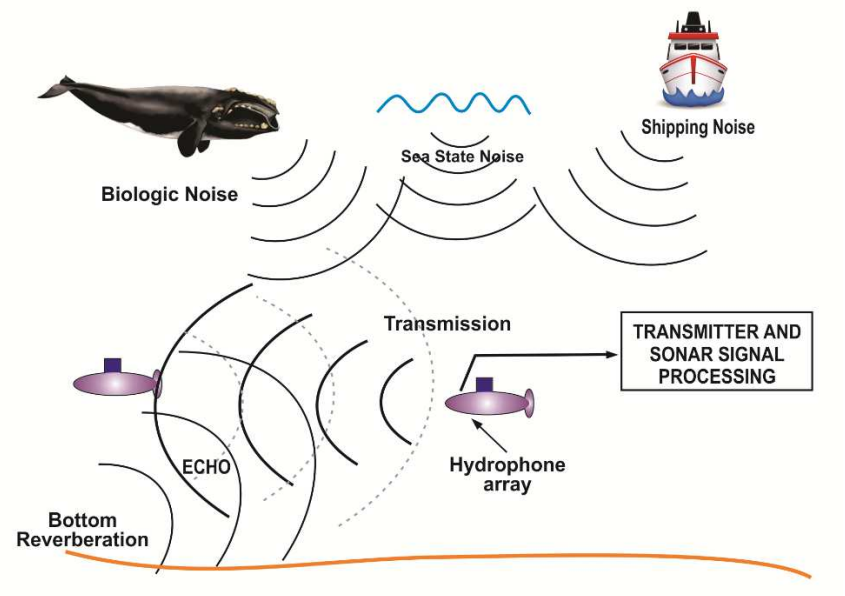

Fig.1.1 Principle of an active sonar

To measure the range of a target, the time '*t*', measured from transmission of a pulse to reception is measured and converted into a range *R*,

$$R = ct/2 \qquad 1.1$$

where *c* is the velocity of light and t is time of echo arrival at the receiver array. The received echo usually is usually shifted in frequency due to Doppler components. The total shift is a sum of four doppler components as follows and is depicted in Figure 1.2.

- A Doppler shift to the transmitted pulse, equal to $S_t cos\theta_t$
- A Doppler shift to the pulse received by the target, $Scos\Phi$
- An equal doppler shift to the echo transmitted by the target, $Scos\Phi$
- A Doppler shift to the echo received by the platform, $S_r cos\theta_r$

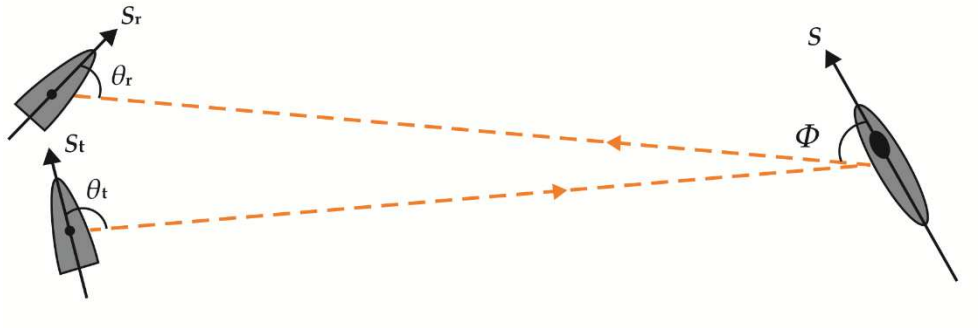

Fig.1.2 Components of doppler shift

The first and fourth components are normally removed by own doppler nullification (ODN). The total frequency shift due to target doppler, $Scos\emptyset$ for an operating frequency '*f*' is given by

$$\Delta f = \frac{2Scos\emptyset}{f/c} \qquad 1.2$$

To determine the bearing (direction of arrival) of the target, an array of hydrophones is employed in which; each element is delayed to steer the response of the array in different directions. The direction of maximum return is estimated and corresponds to the bearing of the target.

Active sonars employ two broad classes of pulse types – continuous wave (CW) and frequency modulation (FM). In CW, a pulse of constant frequency and duration *T* seconds is employed and in FM, the frequency of the pulse changes during the *T* seconds duration of the pulse. CW pulses have good doppler frequency resolution characterises but poor range resolution and poor background reverberation suppression. Conventional active CW systems employ a bank of analog filters, each matched to the bandwidth of the CW pulse. Adequate filters were provided to cover the expected Doppler frequency shifts. In modern CW systems, analog filters are replaced by an FFT processor.

Poor doppler resolution characterises FM pulses but they have high range resolution and reverberation suppression. FM processing replaces the FFT by a process where the output of the beamformer is correlated with a replica of the transmitted pulse.

Active sonars employ matched filters at the receiver for detection, and localisation, as they have *a–priori* information on the signal to be detected. The target signal, together with noise is passed through various signal processing units, before being passed through a decision device, which may be an operator with headphones or display, or in more sophisticated sonars this function may be carried out by software. Further processes may be carried out to classify the target and localise it, as well as to measure its velocity.

1.3 *Passive Sonar*

Passive sonar also referred to as listening sonar, listens to the sound radiated by a target using hydrophones, and unlike their active counterpart, do not require a projector to generate sound waves. They are well suited for battery-powered operations, as their power consumption is very low compared to their active counterpart [2]. They are particularly used in military applications for early warning of ships, submarines, torpedoes etc. and also to detect seismic events. Recently, passive sonars have also been employed for the detection of marine creatures that emit their own characteristic sound, as it has been observed that the pings of active sonars are harmful to them which results in their behavioural changes including vocalisations and foraging behaviour.

The basic principle of passive sonar is depicted in Fig. 1.3. Passive sonars, generally have a much greater detection range than active systems and is used for detection, estimation of range and bearing and tracking and localisation of targets. But, unlike active systems which transmit and receive in a set direction, the passive system must listen to all angles at all times which requires a wide beamwidth. However, a narrow beamwidth is desirable for locating the source amongst the ambient noise. These two conflicting interests are achieved simultaneously by the passive beamforming processor. The beamformer is a spatial filter, which processes the data obtained from an array of sensors in a manner that serves to enhance the amplitude of the desired signal wavefront, relative to background noise and interference. The constructive combination enhances the signals from a particular angle or a set of angles, and destructive interference rejects noise from other angles. Spatial discrimination capability depends on the size of

the spatial aperture, and as the aperture increases, the discrimination improves.

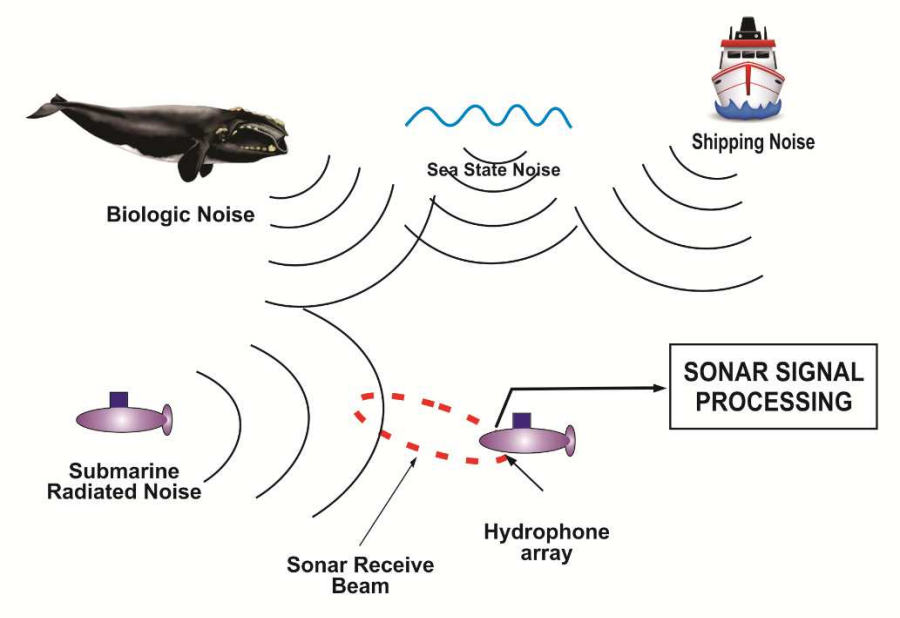

Fig.1.3 Principle of a passive sonar

Passive sonar systems can be made directional to determine the bearing of a target. They estimate range by the following methods.

i. **Triangulation:** measuring the bearing of a target from two well-separated arrays.

ii. **Horizontal Direct Passive Ranging (HDPR):** based on the measurement of wavefront curvature using three well-separated arrays.

iii. **Vertical Direct Passive Ranging (VDPR):** measuring the vertical arrival angles of signals arriving at the same array via multiple paths as well as measuring the time differences between them.

Passive sonars are also used to determine a target's trajectory by a technique referred to as Target Motion Analysis (TMA), which will provide the target's range, course, and speed. TMA is done by identifying from which direction the sound comes at different times using beamforming technique

and utilises the data from the hydrophone arrays to arrive at some solutions. A TMA operator must analyse these solutions for identifying the most probable or hazardous solutions.

1.4 *Active and Passive Sonar Equations*

The sonar equations can be expressed in terms of the sonar parameters, which are expressed in logarithmic scale in dB relative to standard reference levels. These parameters can be determined by the equipment, medium and the target as detailed below.

- Parameters determined by the equipment
  - Projector Source Level (SL), Self Noise Level (NL), Receiving Directivity Index (DI) and Detection Threshold (DT)
- Parameters determined by the medium
  - Transmission Loss (TL), Reverberation Level (RL) and Ambient Noise Level (NL)
- Parameters determined by the target
  - Target Strength (TS), Target Source Level (SL)

The source level (SL) is a measure of the acoustic intensity of the signal measured at unit distance from the source, whether it be the projector or the target. This parameter assumes that the acoustic energy spreads omnidirectionally outwards away from the source. However, most acoustic sources are designed to focus the acoustic energy into a narrower beam to improve efficiency. This effect is accounted for in the sonar equations by the directivity index (DI), a measure of focusing. The detection threshold (DT) is a parameter defined by the system. If the observed signal to noise ration exceeds the detection threshold, a target is deemed to be present.

The intensity of an acoustic signal reduces with range due to propagation effects such as spreading and attenuation and is accounted for by the parameter transmission loss (TL). The target strength (TS) is a measure of how good an acoustic reflector is a target. The echo level will increase with target strength. The noise level (NL) indicates the background noise present and is assumed to be isotropic. The reverberation level (RL) indicates the level of the backscattered sound.

The active sonar equation for the monostatic case in which the projector and the receiving hydrophone are coincident and in which the acoustic return of the target is directed back towards the source is described as follows

$$SL - 2TL + TS - (NL - DI) = DT \qquad 1.3$$

In case of bistatic sonars with projector and receiver separated by a distance, the transmission loss to and from the target is not the same. Also in some modern sonar, it is not possible to distinguish between DI and DT, and it becomes appropriate to refer to DI-DT as the increase in signal-to-background noise ratio generated by the entire receiving system. In case of limited reverberation background, the term NL – DI is replaced by equivalent plane-wave reverberation level RL observed at the hydrophone terminals. The active-sonar equation for the bistatic case then becomes

$$SL - 2TL + TS = RL + DT \qquad 1.4$$

In the case of passive sonars which detect signals radiated by the target, the parameter source level refers to the level of radiated noise of the target at unit distance. Also, the parameter target strength becomes irrelevant, and as only one-way transmission loss is incurred, the passive sonar equation becomes

$$SL - TL = NL - DI + DT \qquad 1.5$$

1.5 *Acoustic Propagation in the Underwater Environment*

Physical properties of the ocean, as well as channel characteristics, influence generation and propagation of sound in the marine environment.

1.5.1 Physical Properties of the Ocean

The ocean being a heterogeneous environment has different levels of pressure, temperature and salinity, all of which varies with geographical locations as well as time. The air-ocean interface at the sea surface is primarily influenced by wind and is strikingly dynamic. The ocean-seafloor interface is mostly stable over time except for occasional disturbances caused by seismic activities and local water currents. Internal boundaries can also rise from within the column of water due to variations in temperature and salinity. Many empirical equations have been derived to determine the speed of sound with varying precision, considering a wide range of parameters which influences its speed in sea water. As expressed in the equation proposed by Leroy, Eq.(1.6), various parameters alters the propagation characteristics of the sound in the ocean.

$$c = 1492.9 + 3(T-10) - \frac{6(T-10)^2}{10^3} - \frac{4(T-18)^2}{10^2} + 1.2(S-35) - \frac{(T-18)(S-35)}{10^2} + \frac{Z}{61} \qquad 1.6$$

where c is the sound velocity in meters per second, T is the temperature in degrees centigrade, S is the salinity in parts per thousand and Z is the depth of meters. An increase in temperature and salinity will increase the speed of sound in water.

1.5.2 Underwater Channel Characteristics

Underwater propagation of sound is affected by channel characteristics like absorption, refraction, reflection, scattering and reverberation.

1.5.2.1 Absorption

The ocean forms a dissipative medium which absorbs acoustic energy. The acoustic absorption by seawater depends on its properties such as temperature, salinity and acidity. The phenomenon is also frequency dependent, such that lower frequencies will reach longer than higher frequencies. For frequencies below 100 kHz, the dominant cause of absorption is the ionic relaxation of magnesium sulphate ($MgSO_4$) molecules in sea water. Other causes of underwater acoustic absorption can be attributed to pressure gradation, the presence of boric acid, shear as well as volume viscosity, and changes in temperature, salinity and pH of the water [3].

1.5.2.2 Refraction

The velocity of acoustic waves in the ocean varies temporally as well as spatially with depth, due to gradations of temperature and pressure. The ocean is stratified into different layers in which, the velocity of sound is highly influenced by temperature in the upper layers, and by pressure in the deeper layers. The acoustic waves are refracted depending on the velocity variations confronted. A sustained up and down refraction may create a SOFAR (Sound Fixing And Ranging Channel) channel or Deep Sound Channel (DSC), which acts as a waveguide for sound to travel thousands of miles without the signal losing considerable energy [1].

1.5.3 Reflection

The ocean being stratified into different layers with differing acoustic impedances causes reflection at the boundaries. The unwanted reflections at the interfaces result in multiple echoes of the same signal. This can result in a series of replicas of the transmitted signal, with the difference between

arrival times and propagation losses along the possible paths [4]. The phenomenon is called multipath and is illustrated in Figure 1.4.

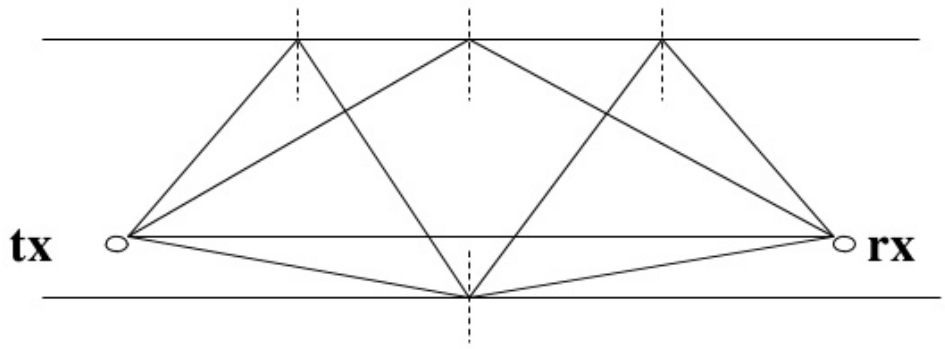

Fig.1.4 Illustration of multipath propagation

1.5.3.1 Scattering and reverberation

The sea is laden with a myriad of inhomogeneities within itself as well as on its boundaries, which form discontinuities in the physical properties of the medium. These discontinuities intercept and echo back a portion of acoustic energy incident upon them. Reverberation is a combination of a large number of echoes, formed due to scattering of sound at the discontinuities of the ocean [5]. The targets producing these echoes may be air bubbles, suspended solid matter, organic matter such as plankton or fish schools, or minute inhomogeneities in the thermal structure. The sea boundaries - the sea surface and bottom, also contributes appreciably to reverberation.

The location of the scatterers determines the type of reverberation, typically like volume, sea surface and bottom reverberation, produced in the sea. In volume reverberation, the scatterers like biological organisms, marine life, suspended matter occur in the volume of the sea. Sea surface reverberation is produced by scatterers located on or near the sea surface, and bottom reverberation originates at scatterers on or near the sea bottom.

1.6 *Noise Sources of the Ocean*

The underwater world is teeming with noise emanating from multiple sources. Man-made noises, natural noises due to wind, waves, currents and rain along with those emitted by underwater mammals and other organisms, create a natural, permanent backdrop that can differ greatly in intensity from place to place. The noises in the sea can be categorised as follows.

- Ambient Noise
- Radiated Noise
- Self Noise

1.6.1 Ambient Noise

The noise background of the sea, posed by the various noise sources, in general, can be termed as the ambient noise. Ambient noise surrounds the hydrophone on all sides unequally over a wide frequency range in a non-isotropic manner. The major sources of ambient noise are as follows.

- **Hydrostatic pressure effects**: Hydrostatic pressure sources include a large number of sources which generate noise due to various physical phenomena, including movement of water itself due to winds, tides, currents etc. and contributes to low frequency noise. Surface waves are also a predominant source of hydrostatic pressure change effect noise and originate mainly due to wind action and contribute to the low frequency noise spectrum. The bubbles are yet another source from which hydrodynamic noise originates.

- **Ocean turbulence**: Turbulence, in the form of irregular water currents of varying magnitudes are capable of generating ambient noise background primarily in three ways. First, such currents may shake or

rattle the hydrophone and its mounting to produce a form of self–noise. Second, the pressure changes associated with the turbulence may be radiated to a distance, and so appear as part of the background at places distant from the turbulence. Thirdly, the turbulence gives rise to varying dynamic pressures that cause voltage fluctuations at the piezoelectric hydrophones.

- **Shipping activity**: Shipping is a dominant source of shallow water ambient noise. Distant ship traffic also contributes to ambient noise at frequencies around 100 Hz.

- **Surface waves**: The sea surface is a major contributor to ambient noise in the frequency range 500 Hz to 25 kHz. This can be due to a combination of different phenomena such as crashing noise produced by breaking whitecaps, flow noise produced by wind blowing over the rough sea surface causing pressure changes at the hydrophones which translate as noise, wave generating action of the wind on the sea surface producing wind noise.

- **Thermal noise**: Thermal noise is the noise generated by the random motion of water molecules. It dominates at frequencies above 100kHz and places a limit on hydrophone sensitivity at those frequencies.
- **Wind noise:** Wind speed appears to determine the noise levels over a wide frequency range in coastal waters such as continental shelves. Measurements made by Piggot [6] in Scotian shelf waters of about 150 ft in depth showed that the ambient noise increased by 7.2 dB per wind speed doubled. This suggests an increase of noise intensity slightly greater than the square of wind speed.
- **Biological Noise:** The sounds produced by biological organisms are diverse and span a broad spectrum. Three groups of marine animals

are known to make a sound: certain kinds of fish, certain kinds of shellfish and marine mammals. Shellfishes such as snapping shrimps contribute to a broad spectrum of noise between 500 Hz and 20 kHz. Marine mammals such as whales and porpoises, create sound by blowing air through the larynx. Biological noise forms an erratic and an unpredictable part of the ambient noise background.

1.6.2 Radiated Noise

Radiated noise is the noise generated by ships, submarines and torpedoes. The machinery involved in these vessels vibrate, which appears as noise at a distant hydrophone. The radiated noise of vessels consists of a mixture of broadband noise and tonal noise and may be characterised as having a continuous spectrum containing superposed line components. The sources of noise on ships, submarines and torpedoes can be grouped into three major classes as follows.

- **Machinery noise** - Machinery noise is the noise caused by the ship's machinery, due to mechanical vibration of different parts of the vessel, which is coupled to the waters via the hull of the vessel.
- **Propeller noise** - Propeller noise is produced by mechanisms such as cavitation at the tips of the blades or cavitation on the blades themselves or mechanical vibration of the blades. When a propeller rotates, regions of negative pressure are created at the tips and on the surface of the propeller blades, which leads to physical rupture of water and cavities in the form of minute bubbles are produced. The production and collapse of cavities formed by the action of the propeller are called propeller cavitation and contributes to noise at the high-frequency end of the spectrum of ship noise. Because cavitation

noise consists of a large number of small random bursts caused by bubble collapse, it has a continuous spectrum.

- **Hydrodynamic noise** - Hydrodynamic noise originates in the irregular and fluctuating flow of fluid past the moving vessel. The excitation and re-radiation of sound by various structures of the vessel are important sources of hydrodynamic noise. The roar of the breaking bow and stern waves of a moving vessel and the noise originating at the intake and exhaust of the main circulating water system are also sources of hydrodynamic noise [1]. The pressure fluctuations associated with the fluctuating irregular flow may be radiated directly as sound to a distance and may even excite parts of the vessel into vibration. Normally, hydrodynamic noise is masked by the machinery and propeller noises and is only a minor contributor to radiated noise. However, when a structural member or cavity is excited into a resonant source of line component noise, hydrodynamic noise becomes a dominant noise source in the region of the spectrum in which it occurs.

1.6.3 Self-noise

Self-noise is the noise due to the sources located in the immediate proximity of the hydrophones. The major contributors to self-noise are machinery noise, propeller noise and hydrodynamic noise. The mechanical vibrations passed on to the transducer by the platform structure due to the manner of mounting, and the electronic noise radiated by high-power electrical devices towards the sonar's circuitry also contributes to self-noise [7]. The vibrations generated by each of these kinds of noise reach the hydrophone in a variety of different acoustic paths, such as through the water column, reflections from the seafloor, backscatter from particulates and also

through the superstructure of the vessel. Self-noise increases with the speed of the vessel, probably due to dome cavitation and other sources of noise. At moderate speeds, flow noise is an important contributor of self noise.

1.7 *Underwater target recognition and classification*

1.7.1 Challenges in Underwater Target Classification

Underwater target classification is a demanding task, owing to the heavy background noise embedded with the targets of interest. The dynamically varying channel characteristics and the ever-changing oceanic environment adds to the complexity, and the signals obtained from the same source often has variable temporal and spectral characteristics. Information masked in the return echoes from the targets are to be interpreted precisely without any ambiguity, for their recognition. However, since the targets are often buried in the ambient noise background of the sea, the recognition and classification task is difficult.

The hydrophones receive an acoustic mixture of requisite signals and noise. Each type of target signal has its unique characteristics and conventionally was identified by human experts either by listening to or by looking at the spectrograms of the processed signals. However, manual processing of received data is often erroneous and time-consuming due to fatigue, boredom, distraction and disinterest caused due to repetitive tasks. Furthermore, most underwater target recognition applications necessitate accurate and online information processing capabilities which are often beyond human capacity. Hence, automatic underwater target classifiers centred around pattern recognition algorithms are to be developed for increased effectiveness and efficiency of military systems for target recognition and surveillance. Automated classifiers exploit computational

power to detect features beyond the human capability and can aid in saving resources (time and human effort) and can also deliver repeatable performance. Efficient pattern recognition algorithms refined by suitable signal processing techniques have been adapted for developing automated underwater target classifiers. Active ongoing research to improve the prediction performance and processing capabilities of automated underwater classifiers are being undertaken by countries across the globe owing to its strategic importance.

1.7.2 Underwater Target Detection

Detecting the presence of a target amidst the background noise essentially involves a decision. The decision is made based on the observations of the received signal at several aspect angles and can be arrived at by comparing the level of certain statistics of the received signal with an estimated statistic. The estimation is based on a criterion depending the level of accuracy needed. Tracking of the target also aids in the recognition and classification tasks as it gives information on target dynamics, which turns out to be useful in certain situations, such as to identify a school of fish from a freighter target or submarine.

The commonly used statistics for the detection purpose is the maximum likelihood ratio, defined as the ratio of the conditional probability density vector of the received data vector when the signal is present to that when the signal is absent. In active sonar, for a stationary white Gaussian noise background, the optimum processor is a 'matched filter' which employs correlation operator for detection. In passive sonar, the signal is to be detected from a noise masking background, and the signal contains both coherent and incoherent components. If the signal and noise are Gaussian

random processes with known spectra, the optimum processor is some form of energy detector.

The considerable variability in the temporal and spectral characteristics of the received signal necessitates some form of signal processing techniques to be applied to the detected signals emphasising the need for a pre-processing stage. A variety of signal processing methods can be employed, to extract reasonable estimates of the target of interest. While doing so, it is a usual practice to analyse short-term data records as most of the noise signals, or processes of interest will be very short in duration and may have time-varying spectra, which can be considered reasonably stable only for short-term records. A typical example is when a noisy moving target or a moving receiver platform cause time-varying spectral responses due to Doppler effects.

1.7.3 Underwater Target Classification

One of the most important purposes of sonar is the timely recognition and classification of underwater targets. However, classification of marine acoustic data has always been a challenging task due to the substantial impediments imposed by the marine environment. The targets of interest are identified by acoustic echoes emitted by them. The reflecting target renders its characteristics to the echoes and the acoustic signals emitted, depending on the target strength which is a collective score of size, geometry, aspect, transmission pulse width and surface reflectivity of the target. The classification problem in most cases is to have a distribution free learning as the designer is unaware of the dynamically changing underlying probability distribution and channel characteristics.

The recognition and classification of underwater targets are centred on the extraction of signature features pertaining to the acoustic targets of

interest. The process of feature extraction can be carried out using various signal processing techniques so that the raw data is transformed into a new data set that can be used by the classifier for system identification. Recognition and classification of signals with fixed signatures in stationary backgrounds is a straightforward task, for which numerous simple and effective temporal and spectral feature extraction techniques exist. But as in the case of the oceanic environment, when the target signature background varies in an unpredictable manner, feature extraction based on simple temporal and spectral methods are insufficient. Moreover, in many cases, underwater acoustic target recognition and identification is of strategic interest and has to be performed accurately on a real-time basis with little or no manual intervention. Hence underwater target recognition necessitates feature extraction techniques that are adequately descriptive of the signal and also robust to variations in the environment. Chapter four covers in detail the feature extraction and selection techniques adopted for acoustic target classification.

The extraction of features provides clues to the classifier, and a useful representative feature corresponds to a strong classification clue. The underlying classifier which works on the feature set to classify the data has to be fast and robust while not compromising on the accuracy. The classifier essentially employs a pattern recognition algorithm to assign each new signal to a set of pre-defined classes according to the observed features. Many pattern recognition algorithms exist which includes classical statistical methods such as Discriminant analysis, Mixture models, Naive Bayes classifiers, Decision tree and Rule-based methods. Modern techniques include Artificial Neural Networks (ANN) and Support Vector Machines (SVM) based classification. Most underwater target classifiers reported in literature use supervised learning approaches, wherein the classifier is trained

with already available data, so as to make decisions on new data. The choice of a classifier depends on the characteristics of data, such as the nature and quantum of data and knowledge about the statistics of the data. It also depends upon the user requirements and on attributes such as the prediction speed, memory usage and accuracy.

Underwater target classifier models based on parametric as well as non-parametric models are reported in the literature. While parametric models refer to learning models with a set of parameters of fixed size for summarisation of data, which is independent of the number of training data, non-parametric models are not be characterised by a bounded set of parameters [8]. Parametric classifier methods include regression techniques [9] and artificial neural networks[10]. Neural network based classifiers have been widely employed for classification of acoustics well as non-acoustic underwater data. Classifiers based on non-parametric classification techniques such as SVM[11] and decision trees[12] have also been developed. In particular, SVMs which require a significantly lower amount of training data is an attractive choice and hence is adopted in this work. The techniques and details of SVM are described in chapter four.

1.7.4 Applications of Underwater target recognition and classification

Robust underwater target recognition and classification systems have application in both military and civilian domain. The navies around the world employ underwater target recognition and classification systems for protecting the strategic security of its oceans and coastline. Non-military applications of underwater target recognition include identification of different species of aquatic fauna.

Most underwater acoustics research efforts have been linked to military applications. Underwater target recognition systems are used for

identifying submarines, underwater mines and other military targets. The acoustic emissions from ships and similar vessels are mainly due to its machinery and cavitation noise and are relatively easy to be detected by an efficient recognition system. However, military targets such as modern-day submarines are protected by low probability detection technology which makes it extremely difficult to be detected. With low probability detection technology, the hull of the sonar is protected by anti-sonar protection tiles that reduces its chances of detection and hence, the submarine noise level is often comparable to the background noise level of the ocean. Thus the development of efficient recognition systems which can reliably extract underwater targets of interest is highly essential.

Fishes and marine mammals use acoustic signals for communication, territorial defence and mating calls. Underwater recognition and classification tools are hence invaluable tools to extract relevant biological information and study the spatiotemporal distribution of the marine ecosystem [13]. Automated acoustic detection techniques have also been used to locate non-communicatory fish sounds of shad fish and snapping shrimps [14]. Acoustic classifiers can also be used for identification and taxonomic classification of fish, based on their acoustic signals[15]. Acoustic recognition techniques can also be extended to identify dangerous marine species such as sharks, so that warning can be issued to marine divers who are in the peripheral region of the species.

Underwater target recognition and classification systems have been employed worldwide to locate targets of interest whether it is of strategic or non-strategic importance. Research efforts in this area has concentrated on improving the performance of the classifier in the complex natural environment imposed by the ocean.

1.8 *Optimisation of the Underwater target classifier*

A robust automatic underwater target recognition is required in most applications particularly for defence applications of strategic importance. However, owing to the changing complexity of the oceanic environment which makes the recognition and classification of underwater targets cumbersome, no single rigid classifier can be an ideal one. Or in other words, a rigid, static classifier will not be able to meet the requirements of the dynamic underwater scenario satisfyingly. Therefore, classifiers for underwater target recognition must be designed to have maximum performance with changing underwater conditions.

Performance of an underwater target classifier profoundly depends on two factors, *viz* the underlying classification algorithm and the feature extraction method used. Performance improvement can thus be achieved by selecting the best suitable algorithm for the problem at hand and also using the most representative feature extraction technique. The parameters of the underlying classifier influence the performance output of the classifier. Fine tuning the parameters of the algorithm to identify the optimal parameters with respect to the changing background conditions will improve the performance of classifiers further, and this can be accomplished through optimisation algorithms. The task of finding the optimal parameter values of the algorithm is referred to as parameter tuning or parameter optimisation and is crucial for maximising the performance of classifiers especially those who are doomed to work in dynamic settings.

Parameter tuning has several important aspects which include interdependency of the parameter values and also the size and dimensionality of the search space. The parameter values are usually interdependent, and hence, it is not sufficient to optimise multiple parameters independently.

Also, the search space can be large if the range of the parameters is very large with multiple dimensions depending on the number of parameters.

Many probabilistic, deterministic and meta-heuristic approaches with varying complexities and worst case guarantees can be used for parameter optimisation of classifiers. A widely used deterministic approach is the grid search, because it is simple and delivers good results. However, it is an exhaustive method and requires a large number of classifier evaluations, and hence not feasible for applications in which the optimisation runtime is important or when the data set is very large or with multidimensional data. Other popular approaches for finding the optimal parameter set include random search, gradient-based optimisation, and meta-heuristic algorithms. The choice of an optimisation algorithm is largely a trade-off between accuracy and algorithm speed. The complexity of the algorithm also determines the choice of an algorithm for optimisation. Chapter five details the different optimisation algorithms and strategies adopted in this work.

While designing a classifier, feature extraction and feature selection methods also play a significant role in determining its performance. Both feature extraction and feature selection are capable of improving the performance, by lowering the computational complexity and building better generalizable models, with reduced storage requirements. Feature extraction maps the original feature space to a new feature space of lower dimension. The feature vector in the new space comprises of features that are distinctive properties of the input pattern intended to be informative and helps in differentiating between different categories, thus facilitating the process of classification. Simple temporal and spectral features may not be representative enough for underwater target recognition amidst the noise imposed by the ocean. Cepstral and features based on higher order statistics have been found to have a high degree of representativeness in spite of heavy

noise background. Different feature extraction techniques are further detailed in Chapter Four.

Feature selection aims to select a subset of highly discriminant features from amongst the extracted features. The relevance of a feature to be selected by a feature selector is assessed by its capability to distinguish between different classes. Feature selection is particularly important when constructing a classifier for underwater targets because, the optimum set of features changes with the dynamic background. There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods. In filter methods, features are selected by their scores in various statistical tests. Wrapper methods consider the selection of a set of features as a search problem, where different combinations are evaluated and compared to other combinations. A predictive model is then used to evaluate several combinations of features and assign scores based on their performance. Embedded methods learn which features best contribute to the accuracy of the model, while the model is being created. The most common type of embedded feature selection methods is regularisation method. Different feature selection techniques are further detailed in Chapter Four.

1.9 *Structural diagram*

Underwater target classification, in essence, is a pattern recognition problem with increased complexity due to dynamically varying channel and background noise characteristics compounded by the fact of not having adequate training data that can represent the extreme variability of the real underwater scenario. The main challenges of an automatic underwater target classifier is to have

- High target classifier accuracy in spite of the combinatorial explosion of different kinds of background noise, target signature variations due to varying channel characteristics, and environmental variations.
- Low false alarm rates even in varying and complex backgrounds.
- Real-time operation necessitated by the nature of applications of the underwater acoustic target classifier.

The basic block diagram of an automatic underwater target recognition system is as shown in Figure 1.5. The input to the classifier are the acoustic noise of 11 classes of targets. A discriminative feature vector is computed for each of the sources through feature extraction and selection algorithms. The feature vector is fed to the classifier which then distinguishes between the different classes. To improve the performance of the classifier, which depends on the classifier parameters, meta-heuristic algorithms are employed to automatically select the optimal parameters.

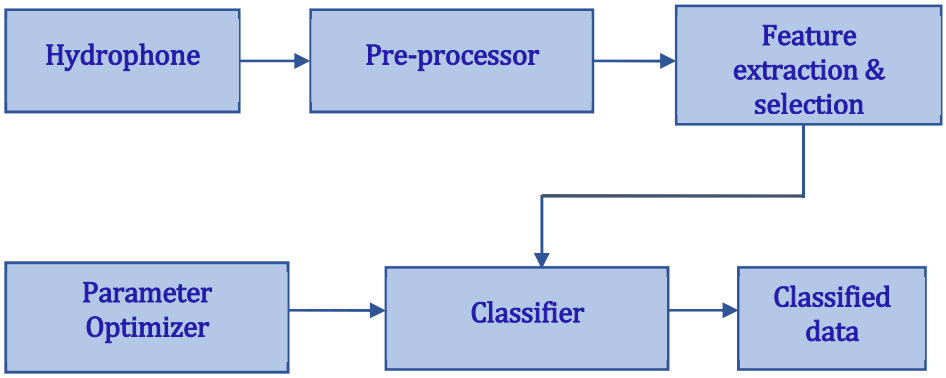

Fig.1.5 Block diagram of an Underwater target recognition system

1.10 *Summary*

This chapter throws light on the salient operational and functional features of sonar systems and the various noise sources in the ocean, highlighting the challenges and impediments in the recognition and

classification of underwater targets. The chapter gives a brief introduction to the underlying principle of the proposed classifier including feature extraction and feature selection techniques. The chapter also throws light on the requirement of an optimisation method to be embedded in the underlying classifier.

CHAPTER 2

REVIEW OF PAST WORK

This chapter touches upon the review of research work reported in the open literature in the areas of underwater acoustic environment, acoustic feature extraction methods, classification algorithms, support vector machines and classifier optimisation. Underwater target classification posses a crucial theoretic and application value and is a very important topic in the world. The detection and classification involve the computation of various statistics and employing suitable classification algorithms to achieve the best possible performance. Support vector machine (SVM) based algorithms, a recent advance in the pattern recognition field, which has proven to be very efficient is presented in this section. Clustering techniques are used to accomplish an effective and valid organisation of data, upon which classification algorithms can be implemented efficiently with ease.

2.1 *Introduction*

This chapter touches upon the review of research work reported in the open literature in the areas of underwater acoustic environment, acoustic feature extraction methods, classification algorithms, support vector machines, and classifier optimisation. Methods of dynamic feature selection, kernel function and its parameter selection are highlighted in this chapter. This chapter is organised into five sections.

Underwater target classification possesses a crucial theoretic and application value and is a very important topic in the world. The detection of the underwater targets from sonar returns is a difficult task due to the complex and variable background of the ocean. The detection and classification involve the computation of various statistics and employing suitable classification algorithms to achieve the best possible performance. Knowledge of the background noises generated by the ocean is crucial for this. Section 2.2 deals with the underwater acoustic environment.

Classification of the received acoustic echoes is to be done using a suitable classification algorithm. A typical signature of these echoes is formed using feature extraction methods. The classification performance of the chosen classifier depends greatly on the credibility of the chosen features. A survey of the state-of-the-art acoustic features is dealt in section 2.3. Recently developed biologically inspired features and comparison of different feature extraction methods are also presented in this section. Feature selection algorithms to select the best representative features from the extracted features are discussed in section 2.4.

Various acoustic classification algorithms and their pros and cons are dealt in section 2.5. Support vector machine (SVM) based algorithms, which has proven to be very efficient is also presented in section 2.5 and is adopted in this work. To overcome specific disadvantages in SVM, variants of SVM such as weighted-SVM, least squares-SVM, reduced-SVM, fuzzy-SVM etc. which have been proposed in the literature are also presented.

SVM is reported to be one of the most efficient learning algorithm in the field of pattern recognition. For classification, SVM tackles the problem of nonlinear separability of classes by mapping the data to a higher dimensional space using kernel functions. The data thus mapped to the higher

dimensional space will be linearly separable. The choice of kernel functions thus turns out to be of critical importance in SVMs. Also, the algorithmic parameters of the SVM classifier profoundly influence its performance. Section 2.7 touches upon the literature survey of various optimisation techniques for selecting the right kernel and its parameters and also for fine tuning the parameters of the SVM classifier.

2.2 *Underwater Acoustic Environment*

Underwater acoustic channels are recognised as the most difficult communication channel in use today. The hydrophones receive a mixture of acoustic events, in which the target signal may be deeply corrupted by the noise imposed by the sea. The noise background of the sea is dominated by ambient noise followed by radiated noise and self noise. Ocean ambient noise results from both anthropogenic and natural sources. The National Research Council's 2003 report, Ocean Noise and Marine Mammals define ambient noise as "The noise associated with the background din emanating from a myriad of unidentified sources. Its distinguishing features are that it is due to multiple sources, individual sources are not identified (although the type of noise source—e.g., shipping, wind—may be known), and no one source dominates the received field" [16]. This definition excludes the anthropogenic noise due to individual sources more localised in both time and space. Surface waves, internal turbulence, fluctuation in sound speed and other small-scale phenomena contribute to random signal variations. This section presents a literature study of various noise sources in the ocean.

Urick [1] discusses various underwater noise sources and their characteristics. Ambient noise, radiated noise of ships, submarines, and torpedoes, self noise of ships, submarines and torpedoes, and their spectral characteristics are described. Dahl *et al.* present a detailed study of

underwater ambient noise [16]. The major anthropogenic and natural constituents of the spectrum are itemised, and two spectra, corresponding to nominal high and low ambient noise levels, is presented to illustrate the dynamic range of underwater ambient noise. These spectra are then compared with field measurements, and some historical trends in field measurements are also presented.

Bradley and Stern [17] discusses the characteristics of the ocean environment and the various underwater acoustic noise sources. Natural (biotic and abiotic) and anthropogenic sources of noise are studied in detail. Hildebrand [18] discusses various anthropogenic and natural sources of ambient noise in the ocean. Different noise sources are dominant in each of 3 frequency bands: low (10 to 500 Hz), medium (500 Hz to 25 kHz) and high (>25 kHz). The low-frequency band is dominated by anthropogenic sources: primarily, commercial shipping and, secondarily, seismic exploration. Shipping and seismic sources contribute to ambient noise across ocean basins, since low-frequency sound experiences little attenuation, allowing for long-range propagation. Ambient noise in the mid-frequency band is primarily due to sea-surface agitation: breaking waves, spray, bubble formation and collapse, and rainfall. Various sonars as well as small vessels, contribute to anthropogenic noise at mid-frequencies. At high frequencies, acoustic attenuation becomes extreme, and all noise sources are confined to an area close to the receiver. Thermal noise, the result of the Brownian motion of water molecules near the hydrophone, is the dominant noise source above about 60 kHz.

Ross [19] discusses the increase in ambient noise level as a result of the rapid increase in shipping across the world. During the past ten years, there has been a virtual revolution in the sizes and speeds of merchant ships, resulting in an increase in low-frequency ambient noise at an average rate of

about 1/2 dB per year. McDonald *et al.* [20] have also correlated the increase in low frequency ambient noise in the Northeast Pacific with the rapid increase in shipping activity. Increase in commercial shipping is accounted for the observed low-frequency ambient noise level increase, and it is reported that low frequency ambient noise within the North Pacific deep sound channel has increased by at least 15 dB since 1950.

Oceanic ambient noise at frequencies above about a kilohertz is strongly correlated to the wind speed. The wind dependence of underwater ambient noise is discussed in [21] which suggests that the ambient noise is increasingly sensitive to the wind speed at low frequency levels. Vijayabaskar and Rajendran, [22] have studied the wind dependence of ambient noise in the shallow waters of the Arabian sea. The study has reported that the noise level increases with wind speeds above 2.5 m/s for frequency above 500 Hz, at a higher rate than that predicted by Knudsen curves which are derived from deep water, because surface winds more pronounce shallow water ambient noise.

Some marine creatures produce sounds which, under certain conditions, completely dominate the ambient noise in the sea. Shrimp noise caused by snapping shrimps often corrupts a sonar ping signal. Shrimp noise is likely to be found around the world in tropical and sub-tropical waters less than 55 meters deep co-existing with ambient noise such as shipping and wind noise. The characteristics and spectrum of underwater noise caused by snapping shrimps are discussed in [23]. Everest *et al*. [24] analyse acoustic noise of snapping shrimps, based upon measurements made off the southwestern and south eastern coasts of the United States, the Hawaiian Islands, and several islands in the Southwest Pacific. Over a shrimp bed, the noise spectrum is found to be roughly independent of frequency from 2 to 24 kHz (the upper limit of measurement), whereas the ambient noise normally

present in the deep sea decreases with frequency. It has been observed that there is a broad peak in the shrimp spectrum somewhere between 2 and 15 kc. Over a shrimp bed the noise at 20 kc is about 25db above the ambient noise for a sea state of 2 (waves less than a meter high, not including swell). The maximum diurnal variation of shrimp noise is from 3 to 6 dB, the noise being greater at night.

Iqbal *et al*. [25] present a statistical characterisation and sensitivity analysis of underwater ambient noise model. The nondeterministic behaviour of ambient noise is modelled through stochastic distribution instead of a deterministic mathematical representation. Ambient noise is approximated as Gaussian distribution in quiet deep sea whereas in case of shallow water it is modelled through asymmetric lognormal distribution. Sound originated from various intrusive sound sources are convoluted with background ambient noise and sensitivity analysis is applied to study the change that takes place in the probability density function model of background ambient noise due to an intrusive sound source.

Bouvet and Schwartz [26] discuss statistical modelling of underwater noises using a Gaussian–Gaussian mixture (GGM) for three different underwater noise data sets. It is shown that one of them can be adequately described by a Gaussian–Gaussian mixture, one is very close to a Gaussian model and is described by a mixture with a very small perturbating term, whereas the third one seems closer to a non-stationary version of the Middleton, class-A model. It is also shown that the classical noise power estimate, calculating the *L*2 norm of the observation vector, is a good approximation of the square of the maximum likelihood estimator of the noise amplitude for the Gaussian–Gaussian mixture.

2.3 *Acoustic Feature Extraction*

Feature extraction and analysis form an important part of acoustic pattern recognition. Features are distinctive patterns extracted from a signal. These characteristic patterns are given to the classifier as classification cues to categorise them to different classes. Extracting and selecting the most distinctive features from the acoustic data significantly improves the classifier performance. This section furnishes a literature review of the various methods of acoustic feature extraction.

2.3.1 Time & Frequency Domain Features

Song *et al*. [27] analyse audio features such as short-time energy, zero-crossing rate, bandwidth, low short-time energy ratio, high zero-crossing rate ratio, and noise rate. The experimental results show that the features are effective in a decision tree based audio classification system for news video, and gives reasonable classification accuracy. Zhouyu *et al.* [28] discuss different spectral based features such as spectral centroid, spectral roll-off, spectral flux, spectral bandwidth etc. A discussion on feature combination and classifier fusion are presented.

A summary of several modern spectral estimation methods is presented by Marple in [29]. Most of the methods are explained in the context of parametric time series modelling. Few non-parametric methods are also discussed. The techniques discussed include classical spectral estimation, autoregressive (Maximum entropy), ARMA, Prony, Maximum Likelihood, Pisarenko and MUSIC methods. Many of these techniques can be implemented with fast computational `algorithms, making these methods viable for real time applications. [30] also suggests various spectral estimation techniques.

Igor Luzin [31] suggests a high-resolution spectrum estimating algorithm. The proposed algorithm is based on an approximation of maximum likelihood criterion, which leads to the two-channel structure of estimation filter. The first channel corresponds to measuring channel like one of Capon's algorithm, and the second one is for taking into account the compensation of non-coherent white noise component. The developed algorithm is a compromise solution of the simplicity of DSP-realisation of Capon's or ARMA-models algorithms and the effectiveness of methods based on covariance matrix singular decomposition.

Empirical feature analysis for audio environment characterisation is discussed by Chu *et al*. in [32]. The author proposes using a matching pursuit (MP) algorithm to obtain effective time-frequency features. The MP-based method utilises a dictionary of atoms for feature selection, resulting in a flexible, intuitive and physically interpretable set of features. The MP-based feature is adopted to supplement the MFCC features to yield higher recognition accuracy for environmental sounds. Extensive experiments conducted demonstrated the effectiveness of these joint features for unstructured environmental sound classification, including listening tests to study human recognition capabilities.

Asefi *et al*. [33] propose pole modelling-based features for audio classification to achieve higher classification performance. A suitable pole modelling computation method is investigated and evaluated with an audio database of 40 human speech samples, and 40 non- human audio signals including aircraft, helicopter, drum, flutes, and piano sounds. An accuracy rate of 85% is achieved using the pole modelling features and linear discriminant analysis (LDA). Comparison with Autoregressive (AR), and Mel-frequency Cepstral Coefficients (MFCCs) indicate that pole modelling is an appropriate tool for real-time audio scene analysis.

2.3.2 Miscellaneous features

Fractional Fourier transforms (FRFT) based feature extraction of audio data for speaker recognition is presented by Jinfang *et al.* in [34]. It has been shown that the features derived from fractional Fourier transform, when applied on GMM based classifiers for speaker recognition, achieve excellent success rate, with acceptable levels of computational efficiency. Li *et al.* [35] attempted classification of buried sea mines by FRFT based acoustic features and has reported good recognition and classification rates. Jleed *et al.* [36] present new feature based on discrete Hartley transform which, require only real arithmetic computations.

Wang *et al*. [37] discuss several compressed-domain features used in audio-visual indexing and analysis. For each feature, the extraction methods, computational complexity, potential effectiveness in applications, and possible limitations caused by compressed-domain approaches are described. The possibilities of extracting some important MPEG-7 visual and audio descriptors directly from the compressed domain is also discussed.

Jian *et al*. [38] discuss underwater target recognition based on wavelet packet transform (WPT) and SVM. The WPT and the wavelet packets energy spectrum feature of sub-frequency-bands of ship radiated noise are extracted after pre-processing and filtering, which is fed to an SVM based classifier. The extracted features have proven to give reasonable classification performance and have the advantage of low-dimension, convenient computing, small storage and fast operation.

Shi *et al*. [39] have attempted using multi-time slice demodulation line spectrum feature of radiated noise for underwater acoustic target recognition. The multi-time slice target recognition method based on the multi-time slice demodulation line spectrum is researched by using the

information entropy evaluation criterion. The comparative result shows that the multi-time slice target recognition could improve effectively the recognition rate of underwater acoustic targets.

2.3.3 Cepstral features

Childers *et al*. [40], has presented a tutorial review of the Cepstrum concepts on data processing. The concepts of power, complex, and phase Cepstrum and relations between them have been discussed. The advantages and disadvantages of windowing the sampled data sequence, taking log spectrum and complex Cepstrum are presented. The issues associated with phase unwrapping, linear phase components, spectrum notching, aliasing, oversampling and zero padding are also discussed. Various forms of cepstral liftering have been presented, and their effects are also analysed. The paper also surveys applications of Cepstrum signal processing procedures in areas such as speech, seismology hydroacoustics etc. Kemerait and Childers in [41] have presented signal detection and extraction using a cepstral technique which involves using the power Cepstrum and complex Cepstrum techniques for decomposing a composite signal of multiple unknown wavelets overlapping in time. The proposed technique makes use of the property of the power Cepstrum for efficiently recognising wavelet arrival times and amplitudes. The property of complex Cepstrum is used to estimate the form of the basic wavelet and its distorted echoes.

A novel method for warping the frequency axis of Cepstrum coefficients in a way analogous to the pre-processing performed by the human ear is discussed by Menve and Preez in [42]. The corresponding equations are derived, and historical background relating to different warping scales is discussed. In the proposed method, the bilinear transform is used to represent the LPC coefficients on a warped frequency scale in which, the

degree of transformation is determined by a warping constant, which is represented in an ARMA representation of the filter transfer function. The second step is to determine recursively the Cepstrum coefficients corresponding to this ARMA transfer function.

Cecilia [43] discusses cepstral synthesis on the Mel frequency scale. They have also proposed an adaptive algorithm for cepstral analysis on the Mel frequency scale. In [44] Molau *et al.* presents a method to derive Mel-frequency cepstral coefficients directly from the power spectrum of a speech signal without the filter bank. Their study has shown that omitting the filter bank in signal analysis does not affect the word error rate. It avoids possible interpolation and discretisation problems, and results in a compact implementation, and simplifies the speech recogniser's front end by merging subsequent signal analysis steps into a single one. Another novel algorithm for extracting MFCC is attempted by Han *et al*. [45]. The algorithm reduces the computation power by 53% compared to the conventional algorithm with a slight compromise of about 1.5%inaccuracy. However, the number of logic gates required to implement the new algorithm is about half of the MFCC algorithm, which makes the new algorithm very efficient for hardware implementation.

Murat *et al*. [46] performed feature extraction using Mel-Frequency Cepstral Coefficients (MFCC), and Linear Predictive Coding derived Cepstral Coefficients (LPCCC) for underwater acoustic signal recognition. Their experiments indicate that MFCC is quite stable under noisy conditions. Andreas [47] attempts the problem of instrument classification from rough audio data. The features are characterised using Mel Frequency Cepstral Coefficients (MFCCs), and Harmonic Representation (HR). Experiments on a large database of real instrument recordings showed that MFCC offered a

more satisfactory characterisation, and therefore the authors comment that MFCCs should be preferred to HR for instrument modelling/classification.

Utpal [48] investigates the performance of LPCC and MFCC for the recognition of Assamese phonemes. A multilayer perceptron-based baseline phoneme recogniser has been built, and all the experiments have been carried out using that recogniser. Observations indicate that the performance of LPCC based system degrades more rapidly compared to MFCC based system with increasing environmental noise.

A comparison of different spectral analysis model for speech recognition using neural networks is presented by Zebulum *et al*. [49]. The performance of a neural network-based recognition system when using different spectral analysis models is compared. Different sets of coefficients, such as Autocorrelation and Mel Cepstrum, are extracted from the speech utterances. Experiments using different sets of coefficients as the neural network inputs are presented. A hybrid system is developed, combining two different sets of coefficients. The results indicate that the hybrid approach outperforms other models.

Dirk *et al*. [50] propose a new type of audio feature (HFCC-ENS) as well as an unsupervised method for detecting short sequences of spoken words (key-phrases) within long speech recordings. In this novel feature, bandwidth-adapted filter banks are used instead of classical MFCC-style filters in the feature extraction step and the time resolution of the resulting features is adapted to account for the temporal characteristics of the spoken phrases. Muhammad *et al*. [51] propose a system for environment recognition using selected MPEG-7 audio low-level descriptors together with conventional Mel-frequency cepstral coefficients (MFCC).

Hynek [52] introduces the perceptual linear predictive (PLP) technique for speech analysis. This technique uses three concepts from the psychophysics of hearing to derive an estimate of the auditory spectrum: (1) the critical-band spectral resolution, (2) the equal-loudness curve and (3) the intensity-loudness power law. An autoregressive all-pole model then approximates the auditory spectrum. A 5th-order all-pole model is effective in suppressing speaker-dependent details of the auditory spectrum. In comparison with conventional linear prediction (LP) analysis, PLP analysis is found to be more consistent with human hearing. PLP analysis is computationally efficient and yields a low-dimensional representation of speech. These properties are found to be useful in speaker-independent automatic-speech recognition. Florian *et al.* [53] develop acoustic features that combine the advantages of MFCC and PLP. A new variant of PLP is proposed by improving the filter-bank, the equal-loudness pre-emphasis and the input for the linear prediction, which has shown improved performance for certain applications over a wide range of clean and noisy acoustic conditions. Hermansky *et al*. [54] introduce PLP-RASTA, a novel extension of PLP, which is more robust to linear spectral distortions and steady-state spectral factors in speech.

Conventional speaker recognition systems have degraded performance under noisy conditions, and hence it becomes essential to have a set of highly representative and discriminative features. Valero and Alias in [55] introduce gammatone frequency cepstral coefficient, a novel feature based on human auditory periphery model and shows its effectiveness in capturing salient speaker characteristics. Taking MFCC computation as the basis scheme, Gammatone cepstral coefficients (GTCCs) are developed as a biologically inspired modification employing Gammatone filters with equivalent rectangular bandwidth bands. The new feature is shown to

perform substantially better than conventional speaker features under noisy conditions.

Jun *et al*. [56] present a novel implementation of the Gammatone filter- based feature with purely time domain filter implementation. The time-domain implementation avoids the approximation introduced by short-time spectral analysis and hence is more precise; and also, avoids the complex spectral computation simplifying its hardware realisation. Their results indicate significant performance improvement under various noise conditions when compared with the widely used MFCC and PLP features.

Valero and Alias [55] have discussed the effectiveness of GTCCs for non-speech audio classification. The performance is evaluated on two audio corpora of 4 hours each (general sounds and audio scenes), following two cross-validation schemes and four machine learning methods. According to the results, classification accuracies are significantly higher when employing GTCC rather than other state-of-the-art audio features. Their analysis shows that, with a similar computational cost, the GTCCs are more effective than MFCCs, in representing the spectral characteristics of non-speech audio signals, especially at low frequencies. Tazi *et al*. [57] evaluates GTCC and concludes that they achieve higher accuracy than MFCC for speech recognition in high SNR conditions. Shao *et al*. [58] analyses the effectiveness of GTCC and proves that they perform considerably better than conventional acoustic features for speech recognition. They extend their work by integrating GTCC with computational auditory scene analysis, which yields very promising recognition performance.

Jerry M. Mendel [59] presents a tutorial on Higher-Order Statistics which describe bispectrum and its applications. Richardson and Hodgkiss [60] give insight into Bispectral features, estimation of the bispectrum and

bicoherence of underwater acoustic signals. Properties of the bispectrum and bicoherence have been detailed. The paper throws light on how the bispectrum estimate can be used to detect non-Gaussianity, non-linearity, and harmonic coupling. The general reasons behind using the bispectrum in signal processing are discussed by Nikias and Raghuveer [61]. These are to extract information due to deviations from normality, to estimate the phase of parametric signals, and to detect and characterise the properties of nonlinear mechanisms that generate time series. Conventional Bispectral estimators and their properties are also discussed. Parametric models of Bispectral estimation and properties of bispectrum are also described.

2.4 *Feature Selection*

Feature subset selection constitutes an important pre-processing phase in classification. The main objective of feature selection techniques is to identify relevant features and remove redundant features, leading to dimensionality reduction and reduced computational complexity, without compromising on the performance. To select a reduced set of relevant features from a set of all features, various feature selection techniques are employed. The main aim of these techniques is to generate an optimal subset of features leading to better classification quality while spending less computational cost compared to maintaining the whole initial feature set.

Shroff and Maheta in [62] have presented a comparative survey of different approaches to feature selection search strategies and evaluation criteria. Complete search techniques give an optimal solution but take exponential time to process dataset that makes it difficult to use. Random search has shown good performance on some real-life problems. Sequential search has low time complexity, but it can fall into local optima problem.

The relative merits and demerits of filter, wrapper and hybrid approaches are also presented.

Pramokchon and Piamsa [63] presents a filter based algorithm to select a subset of features by using outlier cut-offs of relevance between features and targeted categories, which are specified by statistical techniques. The technique is fast as it does not require iterative empirical experiments and also has the advantage that it does not depend on the type of learning machine. The experimental results show that the proposed algorithm can select a small and effective feature subset from multi-class high dimensional data sets.

An overview of wrapper methods of feature selection is presented by Aboudi and Benhlima in [64] pointing out their weaknesses and their strengths. They have discussed various wrapper based approaches for feature selection such as exhaustive search methods, population-based approaches and sequential selection strategies. The paper also discusses the challenges to be addressed while selecting and designing a wrapper based feature selection approach in different application scenarios. Wald *et al.* [65] discusses the stability aspects of wrapper-based feature subset selection and have shown that wrapper based methods are less stable than filter based approaches.

Huang *et al.* in [66] have adopted a wrapper based approach, which utilizes an improved estimation of the conditional mutual information as an independent measure for feature ranking in the local search operations. The mutual information between the predictive labels of a trained classifier and the true classes is used as the fitness function in the global search for the best subset of features. Thus, the local and global searches consist of a hybrid genetic algorithm for feature selection. Experimental results demonstrate

both parsimonious feature selection and excellent classification accuracy of the method on a range of benchmark datasets.

Ververidis and Kotropoulos [67] has presented a novel method to control the number of cross-validation repetitions in sequential forward feature selection algorithms. The criterion for selecting a feature is the probability of correct classification achieved by the Bayes classifier when the class feature probability density function is modelled by a single multivariate Gaussian density. The proposed method is twice faster than the sequential forward selection algorithm that uses a fixed number of cross-validation repetitions, and it maintains the performance of the sequential floating forward selection algorithm.

Zhu and Yang in [68] have proposed a fast sequential feature selection algorithm based on affinity propagation clustering for application in high dimensional data sets. This scheme applies sequential feature selection in the sub-spaces created by a clustering algorithm and collects all features together. Experimental results on several benchmark datasets indicate that the proposed scheme can be implemented much faster than sequential feature selection with comparable performance.

Laanaya *et al*. [69] present feature selection using genetic algorithm for sonar images classification with support vector machines. Genetic algorithms (GA) are used to select the best subset of features to be used by the classification system. The classification results on sonar images are encouraging and indicate significant improvements with the presented approach. Xia *et al.* [70], have proposed an improved GA algorithm (IGA) to reduce the dimensionality of the feature space. Experimental results on several UCI datasets indicate that a better feature subset is obtained with proposed method compared to GA based feature selection. Zawbaa *et al.*

[71], employed social spider optimisation (SSO) as a search method to find optimal feature set. SSO mimics the mechanism of cooperative behaviour of social spiders in nature. The proposed system when evaluated on different datasets, show good classification performance compared to particle swarm optimisation (PSO), and genetic algorithm (GA) based search.

2.5 *Classification Algorithms*

Classification is the problem of identifying to which of a set of categories a new observation belongs, by a training a classifier model with data (or instances) whose category membership is known. This section provides a survey of different classification algorithms. Various state-of-the-art algorithms are discussed first, followed by a survey of literature on SVM which is adopted in this work for classifying underwater targets.

2.5.1 Statistical Classifiers

Shapo and Bethel in [72], proposes CPDF, or "cell probability density function," a new statistical detecting and tracking algorithm suitable for imaging arrays such as sonar arrays. The algorithm consists of pre-processing, detection and tracking phases. The input to the algorithm is the 20 array of intensity levels in all beams as a function of time. CPDF has proven to be very successful in detecting and tracking targets on broadband data collected by sonar arrays and has excelled in especially challenging scenarios with high bearing rates and multiple crossing targets.

Binesh *et al*. [73] have implemented an HMM-based underwater target recognition and classification system using discrete sine transform based features. For the underwater target signals corrupted by moderate levels of white Gaussian noise, the HMM-based classifier gives unambiguous recognition for the set of inputs, except for Beluga and Blue-

Grunt noises. The authors have commented that the system can be augmented by utilising DST features along with other suitable characteristics for robust underwater target classification. Kim and Bae in [74] have implemented a multiaspect HMM-based underwater target classification using synthesised active sonar signals, employing a matching pursuit algorithm for feature extraction from the synthesised sonar signals. Soroosh *et al.* [67] attempted a classification of chaotic signals using HMM classifiers. Results indicated good performance for HMM based classification schemes.

Parada and López in [75] have attempted using Gaussian mixture models (GMM) to detect sounds in recordings and classifies them as background noise, whistles, pulses, and combined whistles and pulses. Two parameters computed using the multiple signal classification algorithm and an unpredictability measure were included in the classifier for improving its performance, which resulted in an increased detection rate and reduced classification error rate. The authors have also explored the potential of multiple signal classification algorithm and unpredictability measure for estimating whistle contours and classifying cetacean species which yielded promising results. The performance of GMM classifier with other classification schemes, for classifying pathological voice has been studied by Wang and Jo in [58]. The comparison of the average classification rate of GMM, Artificial Neural Networks (ANN) and HMM indicated highest classification rate for GMM followed by ANN and then HMM.

Patrikar and Baker [76] have attempted improving the performance of GMM by incorporating additional discriminative training method. The Moore-Penrose pseudo-inverse algorithm, often used in extreme learning machines, is applied to the GMM classifier first trained with the expectation-maximization algorithm. It is shown that on some benchmark pattern classification problems, the proposed method improves the accuracy of the

GMM classifier significantly and produces results that are comparable to the SVM or extreme learning machines. The advantages of the proposed method are that there are no tunable parameters and the training is straightforward and fast.

Cipli *et al.* in [77] have attempted multi-class acoustic event classification of hydrophone data, using an improved HMM-GMM classifier. In this work, the authors have employed an adaptive MFCC with adaptive window length and, a B-spline approximation to the generated Gaussians parameters of the multi-model HMM-GMM classifier to enhance the separation of the decision region. Experimental results for the real recorded hydrophone data indicate high mean classification accuracy (96%), sensitivity (95%), and specificity (97%).

2.5.2 Lazy learning algorithms

Reese *et al.* [78] have presented an efficient recognition technique for mine-like objects using nearest-neighbour classification. The technique employs a nearest-neighbour classifier in conjunction with a non-metric similarity function and synthetic augmentation of the training data, which capitalises on the strong echo highlights produced by broadband sonars while combating the complexity of high-dimensional data. The author's comment that this approach has the potential to be used with forward- or side-looking sonars, as well as with synthetic aperture systems in a side-looking configuration. A novel evidence K-nearest neighbour recognition algorithm based on Dezert-Smarandache Theory (DSmT) has been presented by Zhang *et al.* in [79]. In the new method, the basic belief assignments (BBA) are determined by using the feature similarity between the object and its *k* nearest neighbours in each class, and then the *k*BBA is discounted according to the distance of the *k* nearest neighbours, which are combined using DSmT rule.

The mean of these combined results in each training class is used for recognition of the object. The experimental results show higher recognition rate of at least 2% for *k* values above 2 in the proposed algorithm compared with conventional K-nearest neighbour algorithm.

2.5.3 Decision tree classifiers

Rokach and Maimon [80] describe decision tree based classifiers. Algorithmic frameworks for decision trees and representation of decision trees have been discussed. Analysis of univariate and multivariate splitting criteria is also reported. Various pruning methods and stopping criteria have also been reviewed. Issues related to decision trees and state-of-art decision tree inducing schemes have also been described. Meir T. *et al.* [12], has implemented a decision tree acoustic classifier for classifying various boat types such as ferry, sailboat and speedboat.

2.5.4 Neural network classifiers

Raul Rojas [81] gives a systematic introduction to neural networks. Ben Krose [82] gives an insight into the biological paradigm of neural networks, perceptron learning, unsupervised learning and clustering, back propagation algorithm, self-organising maps, associative networks, Stochastic networks, Kohonen networks, genetic algorithms and other concepts related to neural networks. Some of the most important developments in neural network classification research are summarised by Zhang in [83]. Specifically, the issues of posterior probability estimation, the link between neural and conventional classifiers, learning and generalisation trade-off in classification, the feature variable selection, as well as the effect of misclassification costs have been investigated.

Abdel Allim,*et al.* [84] describes a neural network system which can recognise different types of sonar signals. The parameters that affect the shape of the echoes are identified and compared using fourteen echo signals from three different types of military targets. The results indicate that neural network techniques are potential implementation solutions for recognition functions with targets of complex geometrical shape. Fonseca and Correia [85] employed neural networks for identification of ship noise without any *a-priori* knowledge on the environmental characteristics. The use of neural nets to accomplish this task produced good results, and the authors comment that the system can be improved by teaching the neural nets off-line in more powerful machines and then loaded into the real-time execution system.

Sadjadi *et al.* in [86] developed a new sub band-based classification scheme, for classifying underwater mines and mine-like targets from the acoustic backscattered signals, using a back-propagation neural network classifier. The system consists of a feature extractor using wavelet packets in conjunction with linear predictive coding (LPC). The receiver operating characteristic (ROC) curve of the classifier generated based on the results, demonstrated excellent classification performance of the system. Khotanzad *et al.*[87] have developed a neural-network (NN) based system for the passive detection of underwater acoustic target signals. The proposed system consists of an auto-associative memory whose function is to eliminate the noise and reconstruct the received signal, the output of which is fed to a multilayer perceptron (MLP) classifier trained using the backpropagation algorithm. The MLP outputs a decision regarding the presence or absence of targets. Results clearly indicate promising classification performance.

Farrokhrooz and Karimi in [88] employ a probabilistic neural network (PNN) classifier for classifying ships into three separate classes – heavy ships, medium ships and boats. The acoustic radiated noise of ships is

modelled by an AR(Auto Regressive) model with appropriate order, and the corresponding coefficients are used as classification cues. The performance is examined by using a bank of real data files and the results show that the method has high probability of correct decisions. Madrid *et al.* in [89] describe a target classification system which uses the measured Doppler signatures to excite a neural network. The paper throws light into multilayer perceptron-based neural network and its training using backpropagation algorithm, pointing out the advantages of neural networks.

2.6 *Support vector machines (SVM)*

Support Vector Machine (SVM) is a machine learning tool that is based on the idea of large margin data classification. SVM has strong theoretical foundations based on statistical learning theory and linear and non-linear pattern classification algorithms using SVMs have been reported to have good generalisation performance. A literature review on theory and various practical aspects of SVM is discussed below.

Vapnik in [90] gives an overview of statistical learning theory and introduces SVM. He describes the shortcomings with Empirical Risk Minimization (ERM) induction principle and describes the Structural Risk Minimization (SRM) induction principle that led to the development of SVM. Vapnik in [91] describes in detail the nature of statistical learning theory including the pattern recognition problem setting, risk minimisation principles and methods of pattern recognition.

Huang *et al.* [92] discuss the suitability of SVM for classification and regression. The main differences between SVM and other classification methods such as neural networks are outlined. The advantages of SVM and how SVM achieves these advantages are also presented. Regression by

support vector machines is also covered in detail. The implementation issues of SVM are also discussed.

Christopher Burges [93] presents a tutorial on Support Vector Machines for Pattern Recognition. Practical implementation of support vector training is described. The kernel mapping technique and the concept of VC dimensions are discussed. While very high VC dimension would normally bode ill for generalisation performance, and while at present there exists no theory which shows that good generalisation performance is guaranteed for SVMs, there are several arguments which support the observed high accuracy of SVMs. Results of some experiments which were inspired by these arguments are also presented. Proofs of most of the key theorems are also provided.

SVM is originally designed as a binary classifier. However, most practical pattern recognition tasks necessitate multiclass learning which requires assigning labels to instances where the labels are drawn from a finite set of elements. The classical approach to solving multiclass problems is to consider the problem as a set of binary classification problems.

Knerr *et al.* [94] proposes a stepwise building procedure with single layer training, as an alternative to multilayer neural networks. The procedure is called one-against-one classification (1-a-1) which has been adapted to solve multi-class SVM problems and works by creating binary classifiers for all possible pair of classes. Another popular approach called one-against-all (1-a-a) classification proposed by Vapnik [95] constructs K-separate binary classifiers for a K class problem. Chih-Wei Hsu and Chih-Jen Lin [96] compare the different approaches adopted to solve multi-class problems. Weston and Watkins [97] propose a formulation of SVM that enables a multiclass pattern recognition problem to be solved in a single optimisation.

The authors report that the experiments using benchmark datasets achieved a reduction in the number of support vectors and kernel calculations needed. Crammer and Singer [98] describe the algorithmic implementation of multiclass kernel-based vector machines.

One of the design issues in SVM classifier is reducing the number of support vectors without compromising the classification accuracy. J. Manikandan [99] proposes a novel technique, called modified one-against-all SVM, which requires only a subset of the support vectors. The subset is suitably chosen by including only those support vectors for which Lagrange multiplier is greater than a threshold. Results indicate a reduction in the number of support vectors with very little or no compromise in recognition accuracy.

SVM based classifier was applied to classification of underwater acoustic targets in [100] by Xinhua *et al.*. The experimental data consisted of real recorded signals radiated from four classes of underwater acoustic targets and 16-dimensional target feature vector was extracted using auditory predictive model. The classification performance of SVM was compared with KNN classifier and NN classifier respectively. Results indicate that the average classification rate of the SVM classifier is distinctly 1.12% higher than that of the KNN and 7.54% higher than that of the NN.

Li *et al.* [101] have examined the performance of four different classification algorithms, namely multivariate Gaussian, evidential K-NN, probabilistic neural network (PNN), and SVM on wide-band 80-kHz acoustic backscattered data from six different objects. The performance of these classifiers was then compared together and with those of the back-propagation neural network (BPNN). The robustness and statistical confidence of these results were studied on a large number of trials. SVM

exhibited superior performance over others, and the PNN and multivariate Gaussian exhibited the least performance. Robotham *et al.* [102] also compared different classification approaches namely classification trees (CART), SVM, and multilayer perceptron NN (MLP-NN) in classifying pelagic fish species such as common sardine, anchovy and jack mackerel. Results showed that MLP-NN and SVM exhibited similar performance and CART had poor performance.

Babu and Pradeepa in [103] address the multiclass underwater classification problem using SVM. The paper attempts first to classify ship noise and submarine noise, and then proceed to within ship class or within submarine class problems. Three methods all-against-all, all-against-all Hierarchical, one-against-all compared through different performance metrics. Results indicated that one-against-all gives better performance, requiring less computation compared to other methods when using Gaussian kernel.

Lian *et al*. in [104] use Support Vector Machine (SVM) to classify different ship noises. They have proposed Modified Gammatone Frequency Cepstral Coefficients (MGFCC) features as classification cues to the classifier. Results indicate promising performance with MGFCC features and SVM classifier. Li *et al.* [105] propose an underwater acoustic signal classification method based on wavelet packets-fractal and SVM. The feature vector of ship radiated noise is extracted through wavelet packets-fractal, and SVM realizes multi-class classification of underwater targets. The experiment result shows that the method based on wavelet fractal and SVM for underwater target recognition have good classification rate, which has practical application value. Yang *et al.*, in [106] employs an ensemble of SVM classifiers to classify ship radiated noise. The authors have proposed a novel AdaBoost method based on weighted sample and feature selection

algorithm to reduce the computational complexity associated with ensemble classifiers.

2.6.1 SVM variants

In many practical applications, the obtained training data with which the SVM classifier would be trained is often contaminated by noises. Furthermore, some points in the training data set are misplaced far away from the main body, or even on the wrong side in feature space. These atypical points called outliers tend to become support vectors with large Lagrangian coefficients during the training process. Since the classifier obtained by SVM only depends on the support vectors, these outliers cause SVM training algorithm to make the decision boundary deviate severely from the optimal hyperplane, such that, the SVM is very sensitive to outliers. Xulei *et al*. [107] and [108] presents a weighted support vector machine (WSVM) to improve the outlier sensitivity problem of standard support vector machine (SVM) for two-class data classification. Different weights are assigned to different data points such that the WSVM training algorithm learns the decision surface according to the relative importance of data points in the training dataset.

The least squares support vector machine (LS-SVM) is an interesting variant of the SVM. It performs structural risk through margin-maximisation and has the excellent power of generalisation. Suykens *et al*. [109] discuss a least squares version of support vector machine (SVM) classifiers. Due to equality type constraints in the formulation, the solution follows from solving a set of linear equations instead of quadratic programming for classical SVMs. Reduced Support Vector Machine (RSVM) was proposed as an alternative to standard SVM in dealing with large datasets for the practical objectives, to overcome computational difficulties and to reduce the model complexity. Lee and Huang [110] studies RSVM from the viewpoint of

sampling design, its robustness, and the spectral analysis of the reduced kernel. Experimental results indicate that reduced approximation kernels can retain most of the relevant information for learning tasks in the full kernel.

Olvi *et al*. [111] propose a new approach to SVM classification, wherein each of two data sets is proximal to one of two distinct planes that are not parallel to each other. Classification by proximity to two distinct nonlinear surfaces generated by a nonlinear kernel leads to two simple generalised eigen value problems. Tests demonstrate the effectiveness of the proposed method on simple examples as well as on some public datasets. These examples show the advantages of the proposed approach in both computation time and test set correctness.

Tang and Zhang [112] proposes a multiclass proximal support vector machine (MPSVM) classifier, which extends the binary proximal SVM to the multiclass case. An efficient algorithm to implement the MPSVM by solving a system of linear equations is suggested and results confirm that the proposed algorithm requires much less computational effort than solving the standard SVM, which often requires quadratic programming and can be slow for large problems. Zhuang *et al*. [113] extend the original proximal SVM by learning a weight for each training error. Experimental results prove that the classification algorithm based on this model is capable of handling high dimensional and unbalanced data.

2.7 *Classifier Optimisation*

The SVM classifier has good learning and generalisation ability. However, the parameter setting and choice of kernel function largely influences its performance, and hence parameter optimisation is significant to improve classification performance of an SVM-based classifier. Various

algorithms ranging from exhaustive search techniques to meta-heuristic search techniques can be used to search for the optimal parameters from the parameter space. This section throws light into the literature review conducted on parameter optimisation of an SVM classifier as well as on different optimisation algorithms.

Antoniou and Lu [114] gives an introduction to the optimisation problem and discusses the general structure and properties of optimisation algorithms, principles and applications of constrained optimisation and nonlinear optimisation. Stochastic search and optimisation techniques have been discussed in detail by Spall in [115]. The book apart from giving a detailed introduction on stochastic optimisation also details deterministic search and optimisation procedures such as steepest-descent and Newton-Raphson search, annealing type algorithms such as simulated annealing, evolutionary optimisation techniques such as genetic algorithm (GA), and statistical methods for optimisation.

Jason Brownlee [116] discusses in detail several nature-inspired optimisation algorithms including stochastic algorithms such as hill climbing, scatter search; evolutionary algorithms; physical algorithms such as simulated annealing, harmony search; probabilistic algorithms such as Bayesian optimisation algorithm; and swarm-based algorithms such as particle swarm optimisation, ant colony optimisation, bees algorithm. Various nature-inspired optimisation algorithms such as simulated annealing, genetic algorithm, differential evolution, firefly algorithm, cuckoo search algorithm, BAT algorithm etc. have been described by Yang in [117]. The author has also presented a detailed analysis of the algorithms and has also discussed the salient concepts in optimisation such as no free lunch (NFL) theorem, and the balance between exploration and exploitation aspects of the algorithm.

Eitrich and Lang[118] have proposed a derivative-free numerical optimizer for optimising the algorithmic parameters of an SVM classifier. The authors have also proposed a new sensitive quality measure based on generalised F-measure. Numerical tests on well-known dataset show that the proposed approach can produce support vector machines that are well tuned to their classification tasks.

Phan *et al.* in [119] propose a GA-SVM model for feature weighting and parameter optimisation. The GA part in the GA-SVM model is designed with a special direction based crossover operator. The experimental results show that the GA-SVM model achieves significant improvement in the performance of classification on all the datasets in comparison with traditionally followed grid search method. Zhou *et al.* in [120] have described that the use of the standard genetic algorithm for parameter selection in SVM may cause premature convergence which limits the accuracy of SVM. The authors have proposed a new genetic algorithm with improved genetic operators to optimise the SVM classifier parameters. Experimental results show that the parameters obtained by this method can greatly improve the classification performance of SVM. Ayat *et al.*, in [121] employ quasi-Newton techniques for optimising the kernel parameters in the SVM, by minimising an empirical error estimate. The results show that the quasi-Newton approach proved to converge much faster than the straightforward gradient descent.

Cho and Hoang [122] has attempted feature selection and parameter optimisation of an SVM designed for fault classification in power distribution systems. Particle swarm optimisation (PSO) has been used as an optimiser to improve the performance of SVM classifier by selecting an appropriate feature subset and kernel parameters. The results indicate that the system achieved high accuracy rate in classifying fault types. Yan *et al.*, in

[123] propose a 'dynamic' artificial bee colony (D-ABC) algorithm in which a dynamic activity factor is introduced, for parameter optimisation of a soft margin SVM classifier. Experiments demonstrate that D-ABC algorithm has better performance regarding classification accuracy than traditional methods.

Czarnecki *et al.*, in [124] investigated the Bayesian-based and random search based optimisation of SVM hyperparameters for classifying bioactive compounds. The effectiveness of these strategies was compared with grid search and heuristic optimisation procedures. The results showed that random search optimisation of hyperparameters leads to significantly better performance than grid search and heuristic-based approaches. However, the Bayesian optimisation not only provides better, more efficient classification but is also much faster. Moreover, for the Bayesian approach, the choice of parameters in subsequent iterations is directed and justified; therefore, the results obtained by using it are constantly improved and the range of hyperparameters tested provides the best overall performance of Support Vector Machine.

2.8 *Summary*

An attempt has been made in this chapter to present a state-of-the-art literature in the topic covered by the thesis highlighting the characteristic signatures of typical ocean noise as well as the classes of features that have been considered for realizing the various types of classifiers as reported in open literature. Support vector machine (SVM) based algorithms, a recent advance in the pattern recognition field, which has proven to be very efficient is also presented in this section. Clustering techniques which are used to accomplish an effective and valid organisation of data is also covered.

CHAPTER 3

METHODOLOGY

This chapter describes the methods adopted to realise the proposed underwater target classifier. This work attempts the classification of the eleven target types using 1-a-a (one-against-all) approach based multiclass SVM. The methodology involves the formation of an appropriate feature vector through suitable feature extraction and feature selection techniques. A knowledge base can be constituted by labelling the feature vector with the corresponding class label. The classifier learns from the knowledge base to compare the feature vector of an unknown signal for the closest match in the database to make predictions. Nature-inspired meta-heuristic algorithms have been employed for optimisation of the parameters of the underlying classifier, considering the impact of choosing the right classifier parameters.

3.1 *Background*

The underwater target classification is a highly demanding task often of strategic importance, and requires classifiers with high target recognition rate. Conventionally underwater target recognition was done manually, by analysing the spectrograms of the signals that have been processed by suitable signal processing algorithms in the receiver front-end. However, the manual analysis is often error-prone due to factors such as lack of expertise of the operator, operator lethargy due to repetitious work nature and operator fatigue due to continuous hours of work, which necessitates automatic target classifiers to aid the classification task. The classifiers must also be robust

enough to separate targets of interest amidst the heavy noise background of the ocean. This thesis concentrates on improving the performance of an automatic underwater target classifier.

Highly distinguishing features can be utilised for improving of the performance of a generic classifier for a particular application and also by optimising the classifier for the specific application domain. In this work, various feature extraction and selection methods have been explored for forming the feature vector for the target classifier. Meta-heuristic algorithms are utilised for optimising the parameters of the underlying classifier.

3.2 *Performance Measures*

There are several evaluation criterions for assessing the performance of the classifier. The classifier model must be interpretable regarding the level of understanding and insight provided by the model and also scalable so that it works well even with the larger amount of data.

The most widely used performance metric to analyze the predictive power of the classifier design is accuracy. Accuracy is the proportion of correct results to the total number of cases evaluated by the classifier. However, the accuracy paradox suggests that accuracy alone is not a good measure of the predictive power of a classifier [125]. The accuracy paradox can be stated as follows - predictive models with a given level of accuracy may have greater predictive power than models with higher accuracy. Therefore, optimising classification accuracy alone may fail to capture crucial information in the classification task. Hence, other metrics such as precision and recall also have to be considered, while optimising the classifier for improved performance. Precision, also called positive predictive value is the ratio of the number of positive predictions to the total

number of positive class values predicted. In other words, it is the number of true positives divided by the number of true positives and false positives. Precision is a measure of the classifiers exactness, a lower value of which indicates a higher number of false positives [126]. Recall, also called sensitivity or true positive rate is the ratio of the number of positive predictions to the total number of positive class values in the test data set. In other words, it is the number of true positives divided by the number of true positives and the number of false negatives. The recall is a measure of the classifiers completeness, a lower value of which indicates a higher number of false negatives [126]. Specificity also called true negative rate measures the proportion of true negatives in the negative class values of the test data set. F-measure or F-score, a weighted harmonic mean of precision and recall is often used as a performance metric for classifier evaluation and provides more insight into the functionality of the classifier than the accuracy metric [127]. F-measure defined as in equation 3.1, conveys a balance between precision and recall.

$$\mathrm{F-measure} = 2 \times \frac{\mathrm{precison} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \qquad 3.1$$

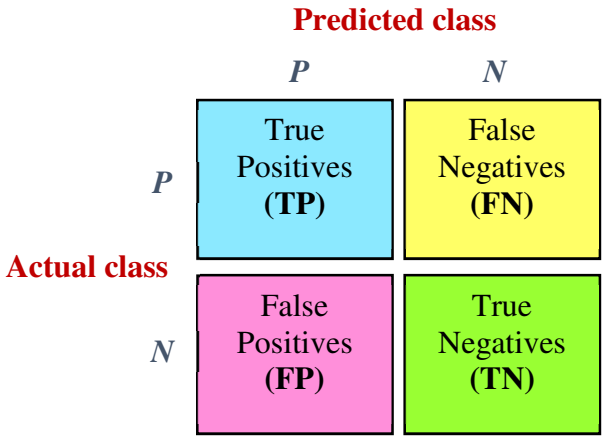

Fig. 3.1 Confusion Matrix

Confusion matrix, proposed by Kohavi and Provost [128] is a matrix that is often used to describe the performance of the classifier. It contains information about actual and predicted classifications done by a classification system. Confusion matrix of a binary classifier is depicted in Figure 3.1. The main diagonal elements represent the correctly classified instances. Performance metrics such as accuracy, precision, recall, specificity and F-score can be calculated from the specific entries of the confusion matrix.

The various performance measures that can be calculated from the confusion matrix are as follows

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3.2}$$

$$\text{Precision} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{3.5}$$

$$\text{F} - \text{score} = 2 \times \frac{\text{precison} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{3.6}$$

In the proposed multi-target classifier, F-score is selected as the performance metric to evaluate the classifier. F-scores of individual classes are caluculated from the confusion matrix. The average value of F-scores of all classes is used as the fitness measure to evaluate the classifier. An example of confusion matrix of the proposed 11-class target classifier is depicted in Figure 3.2.

65	5	4	1	1	4	0	1	0	5	3
5	**53**	5	1	0	2	1	1	4	4	4
4	4	**52**	1	0	5	1	1	4	5	4
2	2	2	**66**	1	2	3	1	1	2	1
3	0	4	2	**67**	2	0	2	2	3	2
5	3	3	0	0	**55**	1	0	3	4	2
0	2	2	3	2	1	**64**	3	2	3	0
2	0	2	2	1	2	0	**65**	1	2	1
5	3	3	0	0	3	0	0	**62**	4	5
6	2	3	2	0	5	0	0	4	**63**	3
4	3	4	1	0	2	1	1	3	4	**57**

Fig. 3.2 An example confusion matrix obtained with the proposed classifier for 11 targets

The overall accuracy, which is the ratio of correctly classified instances to a total number of instances, calculated from the above matrix is 73.6%. The F-scores of individual classed obtained from the above confusion matrix is tabulated in Table

Table 3-1 F-scores of individual targets calculated from confusion matrix in Fig. 3.2

Class	F-score
Class 1	0.68
Class 2	0.68
Class 3	0.63
Class 4	0.81
Class 5	0.84
Class 6	0.69
Class 7	0.83
Class 8	0.85
Class 9	0.73
Class 10	0.67
Class 11	0.70

3.3 *Block Diagram*

A simplified block diagram of the proposed classifier system is shown in Figure 3.2. As discussed, the performance improvement of the classifier is attempted through the selection of appropriate features for the classification process and also through parameter tuning of the classifier. The underlying classification algorithm used is support vector machines, and the performance is assessed.

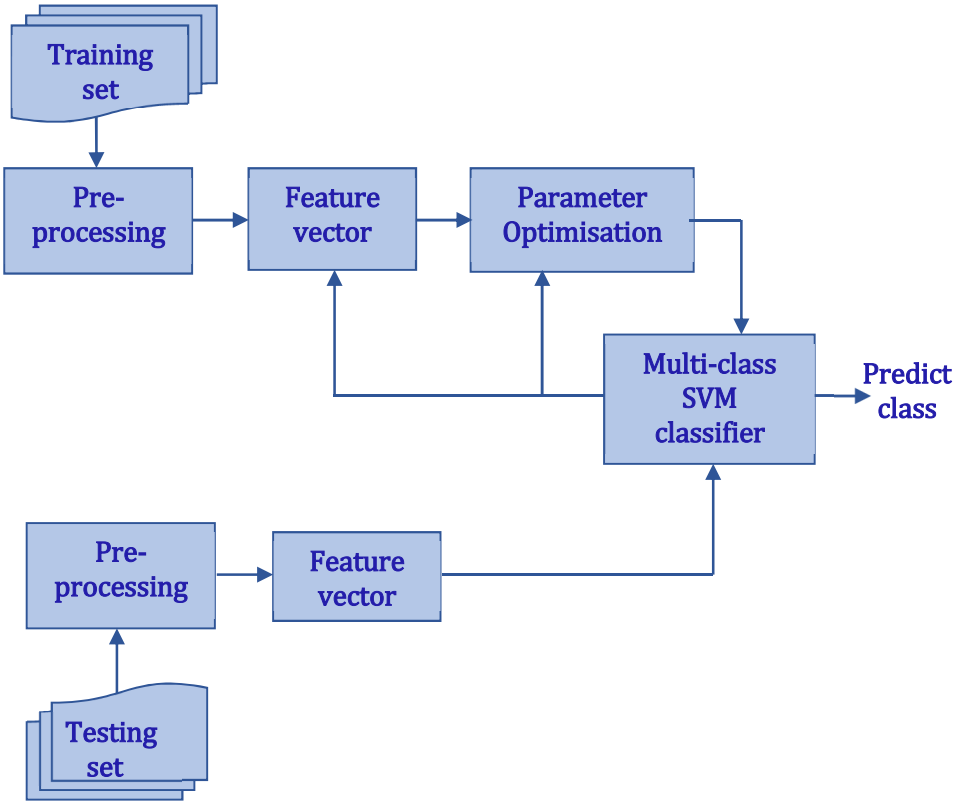

Fig.3.3 Block diagram of the proposed classifier

3.4 *Target Characteristics*

The database used in this thesis consists of acoustic signals from eleven classes of acoustic targets. The eleven classes of acoustic targets and

their associated class labels are tabulated in Table 3.2. Three targets namely, humpback whale, sea lion and snapping shrimp are biological sources of noise, while the rest are of mechanical origin.

Table 3-2 Acoustic targets and their corresponding class labels

Class	Target
Class 1	Humpback whale
Class 2	Ship 1
Class 3	Ship 2
Class 4	Ship 3
Class 5	Ship 4
Class 6	Boat 1
Class 7	Boat 2
Class 8	Boat 3
Class 9	Boat 4
Class 10	Sealion
Class 11	Snapping Shrimp

Water traffic is the most significant source of underwater noise and all vessels from sailing ships to rowboats are a major cause of the anthropogenic underwater noise. Small boats are often a dominant noise source in coastal regions and are characterised by a broader frequency content ranging to several tens of KHz and with fundamental frequencies from hundreds of Hz up to 5-6 kHz [129]. Ship traffic is the principal source of noise in the frequency range 50 to 500 Hz. Ship-generated noise is largely due to propeller action, propulsion machinery and hydraulic flow over the hull. Peak spectral densities for individual ships have been reported in literature to be 195 dB re $\mu Pa^2/Hz$ @ 1 m for fast moving super tankers, to 140 dB re $\mu Pa^2/Hz$ @ 1 m for small fishing vessels [18].

Humpback whales are medium-sized baleen whales which have shown to produce complex vocalisations that can last for many hours. the humpback whales may produce structured series of vocalisations termed 'song' as mating calls as well as 'social sounds' while on their low latitude wintering grounds. 'Songs' are repeated, continuous rhythmic sound patterns that can reach upto 30 minutes duration. Songs are produced by mature males and are thought to be associated with mating functions. The overall frequency range of these songs has been estimated to be 20-1900 Hz [130] with a source level of 140 to 170 dB re 1Pa [131]. Humpback whales, both male and female also produce unpatterned sounds associated with their social and feeding behaviour. The social sounds are likely to be a way for individual humpback whales to signal to others about their location, identity and size. Certain vocalisations also appear to be coordinated for organised feeding activity which are found to span the frequency range 40 Hz – 1250 Hz [130]. Maximum source levels for non-song sounds reported in the literature are 162– 171 dB for low-frequency pulse trains from a feeding whale, 179– 181 dB for blowhole shrieks, and 181– 185 dB for trumpet-like horn blasts[131].

Sea lions are marine mammals characterised by external ear flaps and long fore-flippers which rely on echo-ranging for communication. Sea lion sounds can be divided into two categories [131], whistles and clicks. The whistlers have a long-time duration ranging from several dozens of milliseconds to more than one second. The clicks are a series of pulses with varied pulse shapes. The average time of an estimated pulse reported in literature is about 122 milliseconds with peak frequency lying between 500Hz to 5kHz [132].

Snapping shrimp produce sound by snapping of their claws. They are found in shallow tropical and subtropical waters wherever rock, coral or

other material on the bottom provides interstices in which the shrimp thrive. The sound produced by a shrimp colony can be so loud that the sonars may miss other nearby targets. Over a shrimp bed, the noise spectrum, ranging from 2 to 24 kHz, is found to be roughly independent of frequency. The noise at 20 kHz is about 25dB in a sea state of 2. The maximum diurnal variations of the shrimp noise, has been observed to be ranging from 3-6 dB, the noise being greater at night [24].

3.5 *Multiclass SVM based target classification*

Support vector machines (SVM), is the underlying classification algorithm used in the proposed underwater target classifier. SVM's are robust classifiers rooted in statistical learning theory (SLT), which have been widely applied to a variety of machine learning applications such as image classification [133], speech recognition, character recognition [134],[135] etc. SVM's have stemmed from structural risk minimisation theory, which describes a general model of capacity control, matches the model capacity of SVM with the training data complexity, thus resolving problems like over-fitting and under-fitting. The SVM creates a model with minimised VC dimension, resulting in a low expected probability of error and thus good generalisation performance. A particular advantage of SVM which led to its selection in this thesis is that it requires a lesser amount of training data [90],[91].

SVM relies on three key ideas like mapping data, using patterns and finding hyperplane.

- Map the data to a high dimensional space using kernel functions, where complex classification problems are converted into simpler

problems and employ linear classifiers in this high dimensional feature space.

- Using only the training patterns referred to as Support Vectors that are near the decision surface for classification.
- Finding the hyperplane referred to as the Optimal Separating Hyperplane (OSH) that separates the data with the largest margin, so that the resulting maximal margin classifier will possess good generalisation characteristics

SVM's were originally designed as binary classifiers but have been extended to solve multiclass problems by decomposing the multiclass problem into multiple binary class problems. Two common approaches adopted for multiclass SVM classification are one-against-one (1-a-1) and one-against-all (1-a-a) approach which is detailed in chapter four This work has adopted 1-a-a approach as it is faster than its 1-a-1 counterpart and also has lower computational complexity as only K classifiers needs to be built for a K-class problem.

3.6 *Training and Testing Phase*

The classifier requires training as well as testing phase. In the training phase, the classifier is trained to select the most deterministic features as well as suitable parameters to the algorithm. In the testing phase, the classifier works with previously unseen data with its underlying parameters matched for the problem at hand.

The data is divided into the training and testing phase in the ratio 70:30. It is well known that an intensive training phase will certainly improve the classifier performance. The training data set is again divided into the ratio 70:30. The major 70% training data is fed to the classifier for selection of

deterministic features and parameter tuning. The remaining 30% training data called as validation data is used to calculate the fitness measure of the solutions for validation during the training phase. The distribution of data in the training and testing phase is depicted in Figure 3.4.

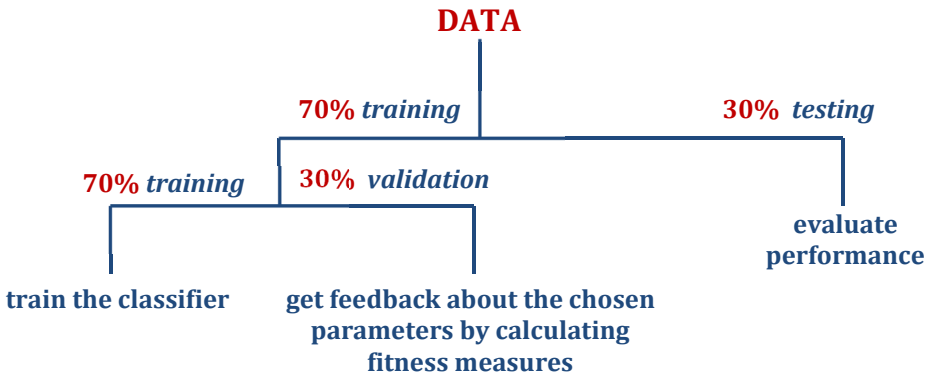

Fig.3.4 Data division for the training and testing phase of the classifier

3.7 *Formation of feature vector*

Formation of the feature vector of the acoustic inputs, which are to be fed to the classifier comprises of two operations, namely feature extraction and feature selection. The purpose of forming the feature vector is to form the acoustic signature for representing the significant characteristics of the data concisely and optimally. The formation of the feature vector is a key problem in pattern recognition, and the optimum feature vector for one classifier may not be optimum for the others [136]. The feature vector must be highly representative to preserve the classification information contained in the original acoustic input data. Hence, for the proposed underwater target classifier, it is essential to design an optimum feature vector suited to the classifier as well as application requirements. There are no exact methods to determine which features are good for a particular application; the only

method is to test and decide whether a particular feature is good for the classification task at hand.

The feature extraction techniques are aimed to generate unique descriptions of the acoustic data which has high distinguishability between different classes. The various feature extraction techniques attempted in this thesis is detailed in chapter four. The classifier is evaluated for the different feature extraction techniques, and a feature vector is formed by combining the features vectors from different feature extraction algorithms. Feature selection techniques are employed to select from the combined multi-featured feature vector, a small subset of features, which is sufficient to predict the target labels. The different feature selection techniques employed are also described in chapter four. Feature selection removes as much irrelevant and redundant information as possible and is mainly resorted to reduce the dimensionality of the feature vector and thus reduce the complexity of the classifier. The overall process of formation of the feature vector is depicted in Fig 3.5.

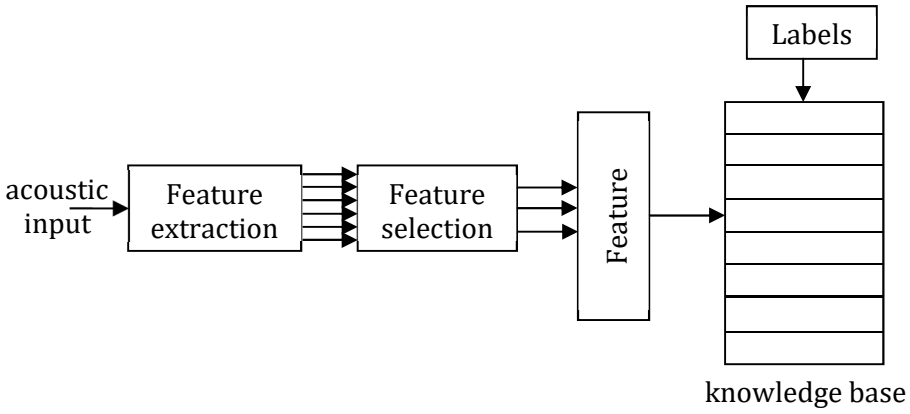

Fig.3.5 Formation of feature vector and knowledge base

The feature vectors formed are given appropriate labelling and a knowledge base is created for the classifier. The classifier uses the labelled

signals in the knowledge base to compare against the feature vector of an unknown signal, based on which the system performs the decision-making process.

3.8 *Parameter Optimisation*

The parameter setting of a classifier profoundly impacts its performance. Therefore tuning the parameters of the classifier to best match with the classification task is important in improving the performance of the classifier. Particularly, for an SVM based classifier, which is acclaimed for its high generalisation capabilities, setting the right kernel parameters is a very determining factor in the classifier performance. Parameter optimisation algorithms are resorted for scanning the parameter space to determine suitable parameters of the classifier. Different parameter search strategies as detailed in chapter four and Meta-heuristic optimisation techniques, detailed in chapter five, have been adopted in this thesis for parameter optimisation..

The optimal parameters of the classifier including the kernel function parameters are found using meta-heuristic algorithms are tabulated in Table 3.3. Each solution encodes 8 parameters a_1, a_2, a_3, a_4, a_5, a_6, a_7, and a_8. The kernels employed are linear, quadratic, polynomial, multilayer perceptron and radial basis functions.

The fitness function of each solution during each iteration of parameter optimisation is calculated from using validation dataset. The fitness function employed is F-score. The candidate solutions are initialized randomly, and the performance of the algorithm is calculated for the varying population size of 15, 20, 25, and 30.

Table 3-3 SVM parameters

Attribute	Parameter and Range
a_1	Kernel (linear, quadratic, RBF, polynomial, MLP)
a_2	KKT violation level (0 to 1)
a_3	Soft Margin Value (0.01 to 100)
a_4	MLP kernel parameter p_1 (1 to 10^22)
a_5	MLP kernel parameter p_2 (–10^22 to –1)
a_6	Polynomial kernel parameter, polynomial order (2 to 12)
a_7	RBF kernel parameter, RBF sigma (0.1 to 10)
a_8	KKT tolerance level (10^-11 to 10^-1)

Meta-optimisation refers to employing an optimisation algorithm to optimise the parameters of another algorithm. The conceptual structure of meta-optimisation is depicted in Figure 3.6 , where a meta-level optimisation algorithm optimizes the parameters of a base level algorithm designed for solving a base level problem.

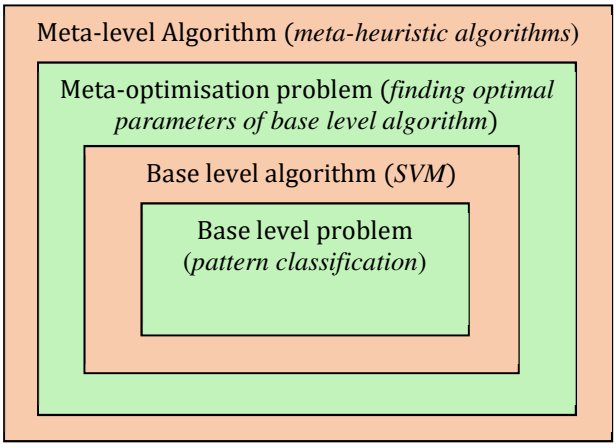

Fig.3.6 Meta-optimisation concept

In this work meta-optimal SVM based underwater target classifier is employed. Five meta-heuristic algorithms, namely, Genetic algorithm, BAT

algorithm, whale optimization algorithm, stochastic fractal search algorithm and symbiotic organisms search algorithm are adopted in this work to tune the parameters for improving the performance. A modified symbiotic organisms search algorithm is also proposed which is found to have better performance in classifying the underwater targets of interest.

3.9 *Summary*

The methodology adopted for the development of a target classifier with improved performance has been discussed in this chapter. The performance measure adopted and distribution of data to training and testing phase and formation of feature vector have been discussed. The target characteristics have also been briefly described in the chapter. An overview of the multiclass SVM classification and parameter estimation is also presented.

CHAPTER 4

TARGET CLASSIFIER

Features also called as attributes are characteristic signatures of the data that are given as input to the classifier. A highly representative and robust feature extraction technique will certainly improve the classifier performance. The chapter highlights the various state-of-the-art feature extraction and feature selection techniques that lead to the formation of the feature vector for the proposed classifier.

The chapter also gives a detailed description of the Support Vector Machine (SVM) based classifier adopted in this work. The chapter throws light on various machine learning concepts such as over-fitting and bias-variance trade-off, that led to the development of SVM. The performance achieved for the proposed classification task in a multi-class SVM based classifier with various feature extraction and feature selection methods are also discussed.

4.1 *Background*

Underwater target classification is a highly demanding task owing to the various noise impediments imposed by the underwater environment. This brings out the need for robust classifiers that can efficiently form non-linear decision boundaries. Furthermore, the extraction of target-specific features which carry the characteristic information about the underwater targets inflicted with noise, is also crucial in determining the performance of the classifier. In this work, an underwater target classifier based on Support Vector Machines (SVM) is adopted for classifying eleven classes of acoustic

targets. Suitable feature extraction techniques are employed to extract source-specific target features. Feature selection algorithms working on the extracted features, dynamically select the pertinent features, which gives the best classification performance.

4.2 *Feature Extraction*

Classification is a pattern recognition problem which identifies the set of categories to which an observation will belong. Different algorithms can be developed to implement the task of classification so that it can predict the class labels of previously unseen observations. The basic block diagram of a classifier is as depicted in Figure 4.1. Classification algorithms typically use features, often referred to as attributes, present in the underlying data as clues for the classification task. Features are the signature patterns that remove redundant information in the signal while representing it best. The classification algorithm operates on the labelled feature set to generate the decision surface in the classification task, and therefore the feature vector should be good predictors of the class membership.

Depending on the classification task, different features may be significant, and hence it is essential to have knowledge about the possible feature set and identify the best among them suited for the underlying classifier. The feature vector, y, that is composed of several sets of features should be as discriminative as possible, between the considered classes.

Various feature extraction algorithms working on the input signal can be used to extract the features. The broad classification of acoustic features includes time domain, frequency domain, cepstral domain and higher order spectral features. The choice of the specific features is the result of extensive

experimentation and conclusions that stem from them. The classifier design can be simplified using effectively chosen feature vector.

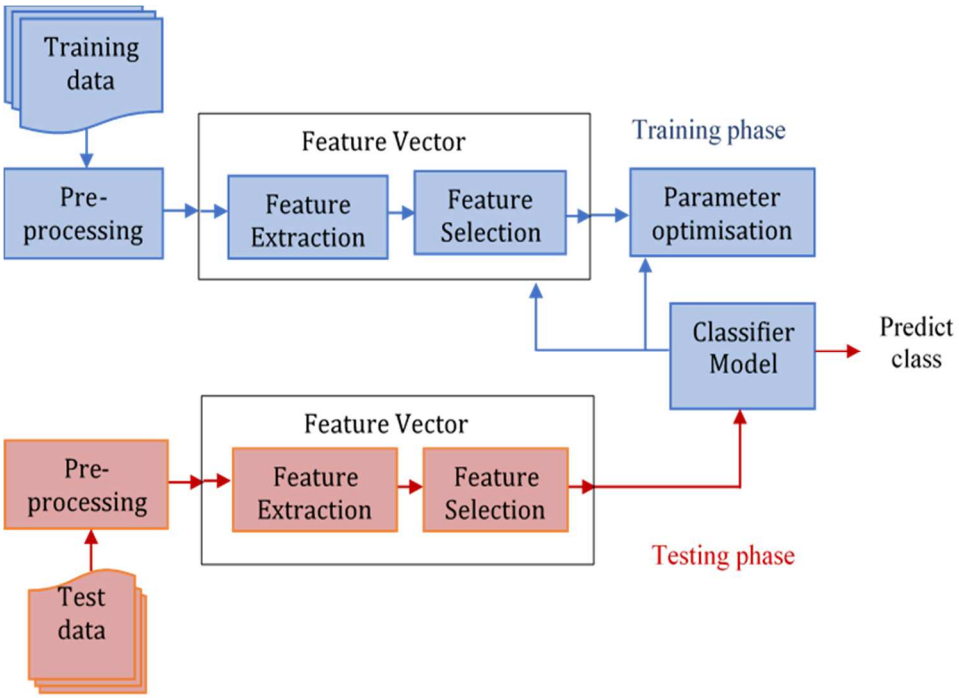

Fig.4.1 Basic block diagram of a classifier

Most underwater acoustic signals are non–stationary. Short term processing techniques are used to calculate characteristics of an acoustic signal. Let $x(n), n = 1, \ldots\ldots\ldots, L$ be samples of the acoustic signal and L be length of the signal. The acoustic signal is divided into short-term windows or frames in which the signal is assumed to be 'quasi–stationary' and the frames can either be overlapping or non-overlapping. The feature value f is calculated for each frame which results in an M-element array of feature values $F = f_j$, where $j = 1, \ldots\ldots\ldots, M$. The length of the feature array is equal to the number of frames; i.e. $M = [(L - S)/N] + 1$, where N, is the

window length or say the number of samples, S is the window step and L, is the total number of acoustic samples of the signal.

4.3 *Time Domain Features*

Time domain features are simple representations of the signal energy changes and can be directly extracted from the time domain. Acoustic signal discrimination based on energy differentiations like energy, energy entropy offer a simple way of acoustic analysis and can be used in combination with elements that contain frequency-related information.

4.3.1 Energy

Most marine acoustic sources are time varying in nature. The standard deviation of the energy sequence can be used to detect signals with large energy variations. Energy can be calculated by the equation 4.1.

$$E(i) = \frac{1}{N}\sum_{n=1}^{N}|x_i(n)|^2 \qquad 4.1$$

where $x_i(n)$-is acoustic sample on *i*-th frame of length *N*.

4.3.2 Energy entropy

Energy entropy is a measure of abrupt changes in the energy level of an acoustic signal. Each frame is further divided into k sub-frames of fixed duration. For each sub-frame j, the normalised energy e_j^2 is calculated, i.e., the energy of sub-frame divided by the corresponding short frame energy.

$$E(i) = \frac{1}{N}\sum_{n=1}^{N}|x_i(n)|^2 \qquad 4.2$$

$$e_j^2 = \frac{Esubframe_j}{Eshortframe_i} \qquad 4.3$$

The entropy of the sequence is then computed for each frame using the equation 4.4.

$$H(i) = -\sum_{j=1}^{k} e_j^2 \log_2\left(e_j^2\right) \qquad 4.4$$

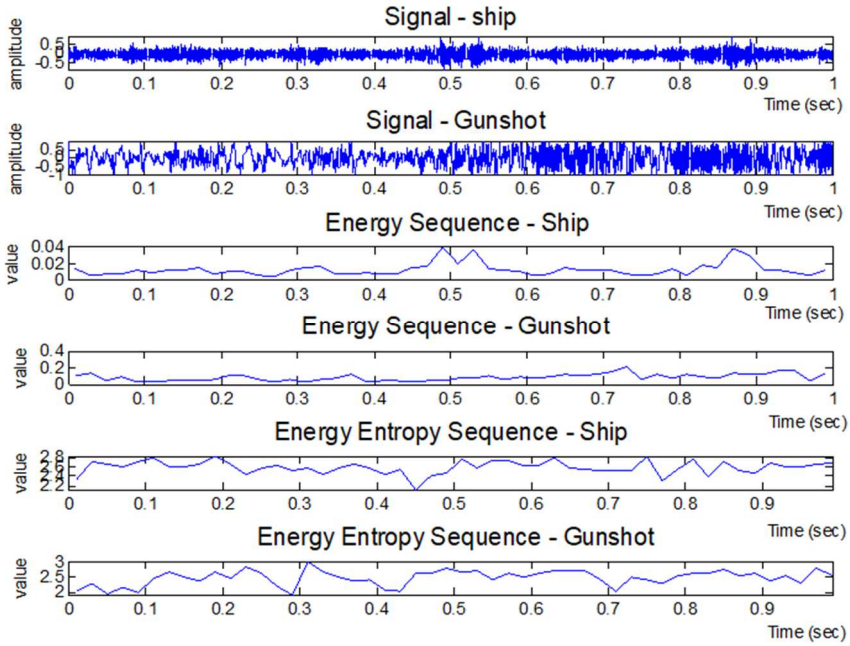

Fig.4.2 Energy and Energy entropy sequence of ship and gunshot noise

Any unexpected variations present in the structure of an acoustic signal can lead to lowering of the energy-entropy. Figure 4.2 shows the magnitude of energy and energy entropy obtained from a ship noise with seemingly lesser energy variations and a continuous series of gunshots which is a typical example of signals having large energy variations.

4.4 *Frequency Domain Features*

Short-Term Fourier transform or alternatively Short-time Fourier Transform, (STFT), of the acoustic signal computed using Discrete-Time Fourier Transform (DFT) helps in gaining inference on the spectral characteristics of the time-varying signal. The longer time signal is divided

into shorter segments of equal length and DTFT computed separately on each shorter segment.

$$X(w,n) = \sum_{m=-\infty}^{\infty} x(m)\, w(n-m) e^{-j\omega m} \qquad 4.5$$

where $w(n)$ is any window function suitable for Short Term Processing. This helps in determining the sinusoidal frequency and phase content of local sections of a signal as it changes over time.

4.4.1 Spectral Centroid

The spectral centroid, C_i, of the i-th frame is defined as the centre of gravity of its spectrum, i.e., the frequency at which the magnitude spectrum can be divided into two portions of approximately equal mass. The centroid of a spectral frame, which is the measure of the spectral shape, can be defined as the average frequency weighted by amplitudes, divided by the sum of the amplitudes.

$$C_i = \frac{\sum_{k=1}^{N} k X_i(k)}{\sum_{k=1}^{N} X_i(k)} \qquad 4.6$$

Centroid models the sharpness of sound and textures with high frequencies will possess higher centroid.

4.4.2 Spectral Roll–off

Another spectral feature, which gives a measure of the spectral shape, is the spectral roll off, RO, is measure of the amount of the right-skewedness of the power spectrum and is defined as the frequency below which 85% of the magnitude distribution of the signal is concentrated.

$$i.e.\quad RO = Minimum(R), \text{ such that} \qquad 4.7$$

$$\sum_{k=0}^{R} S_k \geq 0.85 \sum_{k=0}^{N-1} S_k$$

4.4.3 Spectral Flux

Spectral flux is a measure of the local spectral change between any two successive frames and a high value indicates rapid change of the power spectrum of signal. It is defined as the difference between squares of normalised magnitudes of the spectra of any two successive frames.

$$Fl_{(i,i-1)} = \sum_{k=1}^{N}(EN_i(k) - EN_{i-1}(k))^2 \qquad 4.8$$

where $EN_i(k) = \frac{X_i(k)}{\sum_{i=1}^{N} X_i(k)}$.

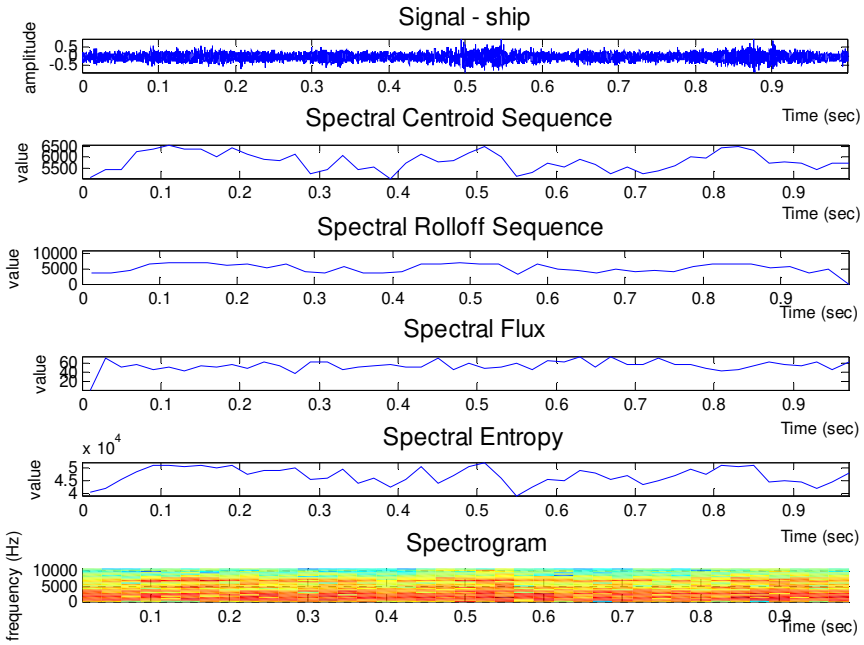

Fig.4.3 Frequency-domain features and spectrogram of ship noise

4.4.4 Spectral Entropy

Spectral entropy gives an estimate of the abruptness in the spectrum of a signal. It is computed as

$$H = -\sum_{f=0}^{L-1} n_f \log_2(n_f) \qquad 4.9$$

where n_f is the normalized spectral energy computed as

$n_f = \frac{E_f}{\sum_{f=0}^{L-1} E_f}$ where E_f is the energy of the f^{th} sub-band.

Figures 4.3 and 4.4 respectively shows the plots of frequency domain features of a ship noise and a continuous series of gunshots.

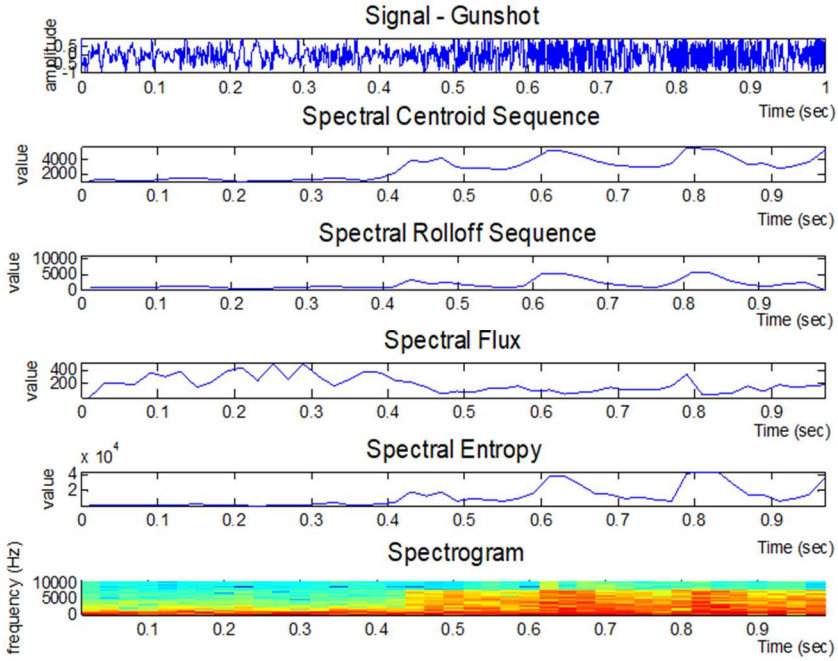

Fig.4.4 Frequency-domain features and spectrogram of gunshot noise

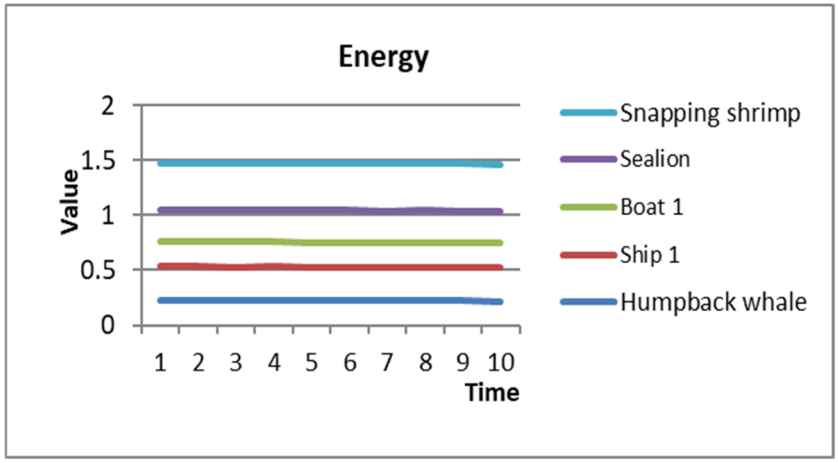

Fig.4.5 Plot of energy of different targets

The time and frequency features are often grouped into a feature vector which forms the signature pattern for classification. Fig.4.5 to 4.10 shows the plot of various time and frequency features computed for different types of target noises.

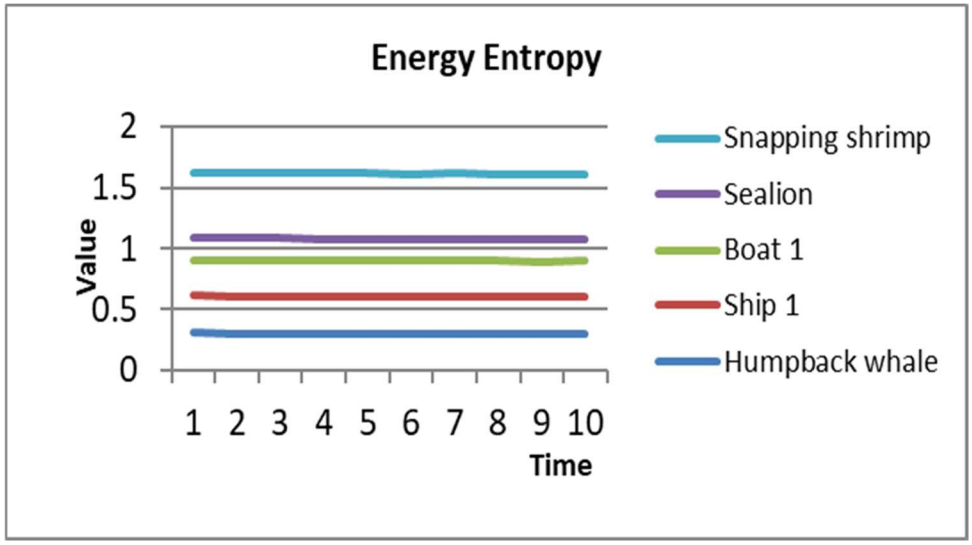

Fig.4.6 Plot of energy entropy of different targets

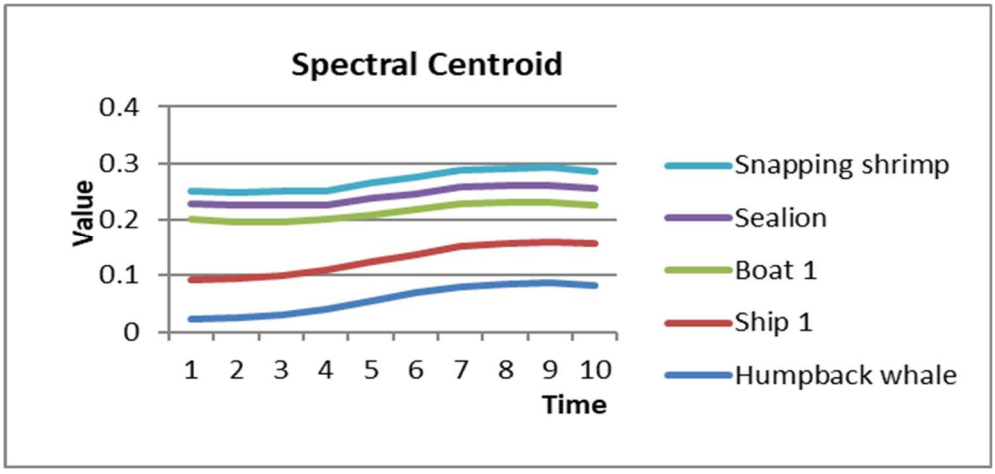

Fig.4.7 Plot of spectral centroid of different targets

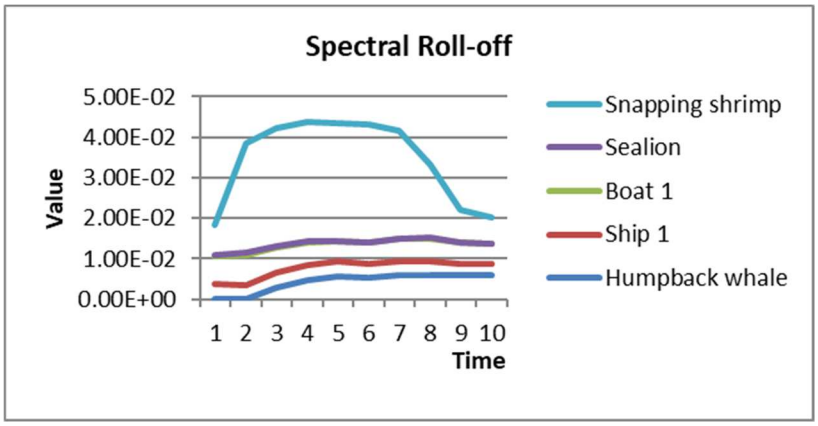

Fig.4.8 Plot of spectral roll-off of different targets

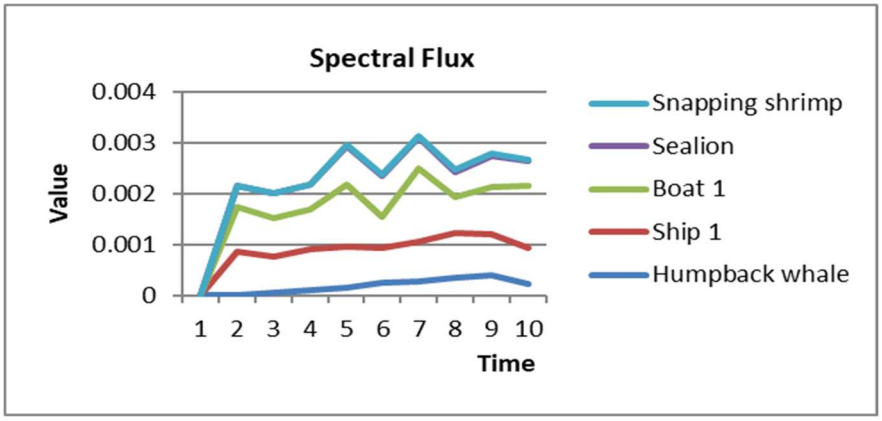

Fig.4.9 Plot of spectral flux of different targets

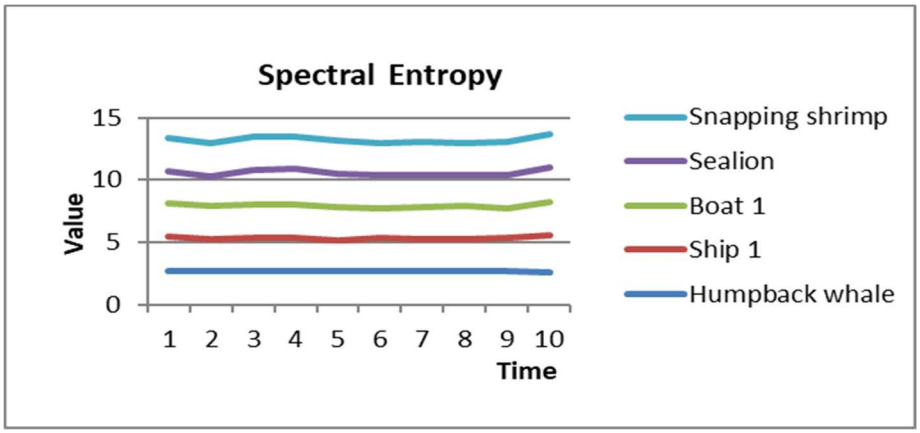

Fig.4.10 Plot of spectral entropy of different targets

4.5 *Cepstral Features*

Time domain and frequency domain features cannot always guarantee reliable classification in the presence of composite ambient noise and varying oceanic parameters. Nonlinear techniques such as cepstral analysis are capable of yielding potential features that can better aid in the process of classification of patterns heavily laden with noise. Cepstral analysis refers to a group of homomorphic signal processing methods that is frequently useful in decomposing non-linearly combined signals. The concept of homomorphic analysis, as a technique for non-linear signal processing was proposed by Alan V. Oppenheim [137]. The basic idea of homomorphic analysis is to use non-linearity to transform convolved or non-linearly related signals to additive signals which may then be processed by linear techniques.

The canonic representation of homomorphic systems consists of a cascade of three systems consisting of a system of forward and inverse operations with a linear operation sandwiched in between as shown in Fig.4.11.

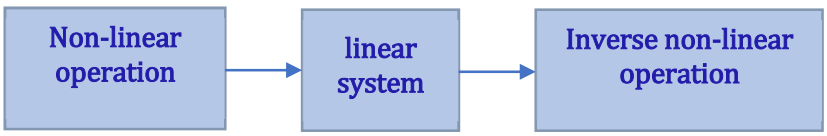

Fig.4.11 Canonic representation of a Homomorphic system

The first system consists of an invertible non-linear operation that maps a non-additive combination such as convolution into a simple additive combination. The second system is a linear system obeying additive superposition, and the third system is the inverse of the first nonlinear system. Thus, for signals combined by convolution, a homomorphic

deconvolution system maps convolution into addition, then addition into addition, and finally addition into convolution [138].

The spectrum of a signal can be decomposed into two components, the slowly varying part, referred to as the filter or spectral envelope and the rapidly varying part, referred to as the source or harmonic structure. Separation of these two components can be achieved by taking the Cepstrum, an anagram of the word spectrum, which is a homomorphic transform. Cepstrum is defined as the inverse Fourier transform of the log magnitude of the Fourier transform. The methodology of computation of Cepstrum is depicted in Fig.4.12. Cepstral analysis is capable of separating the filter and source components of the spectrum in a new domain called quefrency.

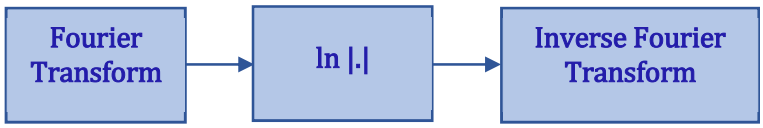

Fig.4.12 Methodology for computing Cepstrum

The convolution of any two signals in the time domain can be transformed into multiplication of the signals in the frequency domain through Fourier analysis. Further, on applying logarithm to the Fourier transform, convolution in the time domain can be transformed into sum of log-magnitude components in the frequency domain. Now applying an inverse Fourier transform to the log spectrum takes the function back into the time domain, and gives a measure of the rate of change of the spectral magnitudes.

Consider a signal $s(t)$ as a convolution of the two components, $x(t)$ and $y(t)$, so that

$$s(t) = x(t) * y(t) \qquad 4.10$$

Then, taking Fourier transforms of both sides,

$$s(\omega) = X(\omega)Y(\omega) \quad 4.11$$

The magnitude spectrum of the signal can be written as

$$|S(\omega)|=|X(\omega)||Y(\omega)| \quad 4.12$$

and taking the logarithms of both sides gives,

$$ln|S(\omega)| = ln|X(\omega)| + ln|Y(\omega)| \quad 4.13$$

Thus, convolution in time has been transformed into a sum of log-magnitude components in the frequency domain. The individual components may be separated from each other using a suitable cepstral filter referred to as lifter (anagram of filter). They may then be transformed back by applying Inverse Fourier Transform (IFT). The IFT takes the signal to a domain, similar to the frequency domain, called the quefrency (anagram of frequency) domain. However, the phase information will be lost as a result of applying the magnitude operation.

Applying an inverse Fourier transform to the log spectrum gives

$$F^{-1}\{\ln|S(\omega)|\} = F^{-1}\{\ln|X(\omega)|\} + F^{-1}\{\ln|Y(\omega)|\} \quad 4.14$$

For the signal $s(t) = x(t) * y(t)$ the cepstra is given by

$$c_s(n) = c_x(n) + c_y(n) \quad 4.15$$

$c_x(n)$ and $c_y(n)$ are the cepstra of the signals $x(t)$ and $y(t)$ respectively.

As the Cepstrum is derived from the power spectrum of the signal, it is always a real function of frequency. Because the log-magnitude spectrum is real and symmetrical for real signals, the final IFT can also be replaced with a Discrete Cosine Transform (DCT).

Cepstral feature extraction schemes have been developed for speech recognition application. However, they have also proved to be successful in other acoustic recognition applications such as audio forensics [139], audio watermark detection [140], acoustic environment identification [141] and underwater target recognition [142].

4.6 *Linear Prediction Coefficients*

Linear Prediction Coefficients (LPC) are conventional features used in speech processing. According to the speech synthesis model, speech can be modelled as the output of a linear time-invariant system. LP analysis provides a robust, reliable, and accurate method for estimating the parameters that characterise the linear time-varying system representing the vocal tract [143]. The basic idea of LP analysis is that, at a particular time k, the signal sample is represented as a linear sum of n previous samples

$$S_n = \sum_{k=1}^{p} a_k S_{n-k} \qquad 4.16$$

in which a_k is known as the predictor coefficients and p is the prediction order. The predictor coefficients provide a good estimate of the spectral properties of the speech signal and are often used as features. The two widely used methods for estimating the LP coefficients are the autocorrelation method and the covariance method described in section 4.6.1 and 4.6.2 respectively. Figure 4.13 shows the plot of log magnitude of frequency response of LPC coefficients obtained with autocorrelation method and covariance method.

Both methods assume that neither the vocal tract shape nor the glottal waveform changes, and determines the predictor coefficients in such a way that, the prediction error E, is minimised in the least squared sense.

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} (s(n) - \sum_{k=1}^{p} a_k s(n-k))^2 \qquad 4.17$$

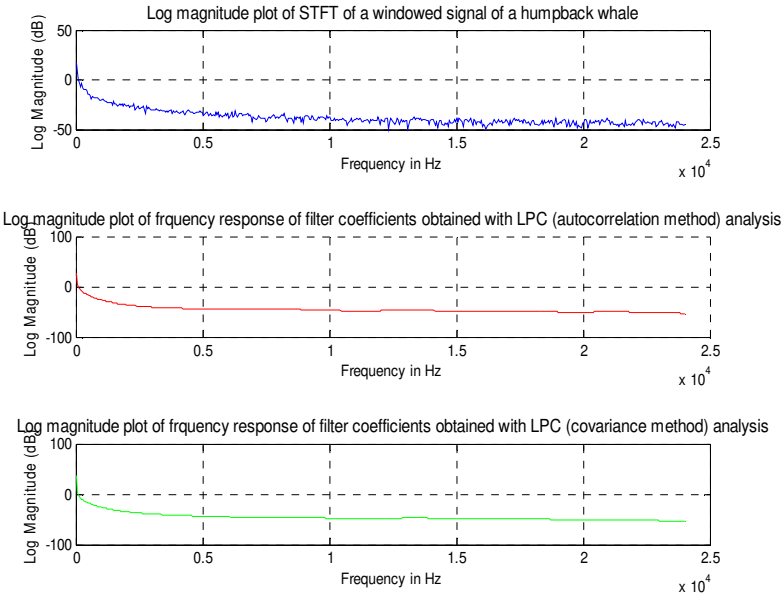

Fig.4.13 Plot of log magnitude of frequency response of filter coefficients obtained with LPC by autocorrelation method and covariance method

In the linear prediction model, the filtering action of the vocal tract, the radiation, and the glottal flow is represented by a discrete linear filter with p poles [144]. The transfer function of this filter in the complex z domain is related to the predictor coefficients by equation 4.18.

$$H(z) = 1 \Big/ \left[1 - \sum_{k=1}^{p} a_k z^{-k}\right] \qquad 4.18$$

The transfer function $H(z)$ is related to the samples of the impulse response of the filter by equation 4.19

$$H(z) = \sum_{n=0}^{\infty} h_n z^{-n} \qquad 4.19$$

On substituting equation 4.19 in equation 4.18, the relationship between predictor coefficients and the samples of the impulse response can be can be derived as in equation 4.20

$$h_n = \begin{cases} \sum_{k=1}^{p} a_k h_{n-k} & n > 0 \\ 1 & n = 0 \\ 0 & n < 0 \end{cases} \qquad 4.20$$

The first p samples $h_1, h_2, \ldots\ldots h_p$ are sufficient to determine the p predictor coefficients uniquely [144]. Thus the linear prediction characteristics of the speech wave can be represented by p numbers, $h_1, h_2, \ldots\ldots h_p$.

4.6.1 Autocorrelation Method

In this method the values of the predictor coefficients a_k that minimise E are found by assigning the partial derivatives of E of the windowed speech signal with respect to a_k to zeros.

$$\frac{\partial E}{\partial a_k} = 0 \text{ for } k = 1, \ldots\ldots\ldots, p \tag{4.21}$$

which yields p equations with p unknown variables as

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s(n-i)s(n-k) = \sum_{n=-\infty}^{\infty} s(n-i)\, s(n), \quad 1 \le i \le p \tag{4.22}$$

which can be expressed in terms of autocorrelation function as

$$\frac{\partial E}{\partial a_k} = 0 \text{ for } k = 1, \ldots\ldots\ldots, p \tag{4.23}$$

$$\sum_{k=1}^{p} R(|i-k|)\, a_k = R(i), \quad 1 \le i \le p \tag{4.24}$$

The set of linear equations given by equation 4.24 can be represented in the matrix form as

$$\begin{bmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(2) & \ldots & R(p-2) \\ \vdots & \vdots & \ldots & \vdots \\ R(p-1) & R(p-2) & \ldots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix} \tag{4.25}$$

The equation 4.25 can be represented as

$$Ra = r \tag{4.26}$$

The resulting matrix is a Toeplitz matrix where all elements along a given diagonal are equal. This allows the linear equations to be solved by the Levinson-Durbin algorithm.

4.6.2 Covariance Method

The covariance method is very similar to the autocorrelation method, with the difference that the covariance method windows the error signal instead of the original speech signal. The energy E of the windowed error signal is

$$E = \sum_{n=-\infty}^{\infty} e^2(n)w(n) \tag{4.27}$$

The p equations with p unknown variables in equation 4.22, obtained by setting the partial derivative of the error signal with respect to a_k to zero is expressed in terms of the covariance function as

$$\sum_{k=1}^{p} \varphi(i,k)a_k = \varphi(i,0), \qquad 1 \le i \le p \tag{4.28}$$

are coefficients of a linear predictive filter in which value of next sample is determined by a linear combination of previous samples.

The set of linear equations given by equation 4.28 can be represented in the matrix form as

$$\begin{bmatrix} \varphi(1,1) & \varphi(1,2) & \dots & \varphi(1,p) \\ \varphi(2,1) & \varphi(2,2) & \dots & \varphi(2,p) \\ \vdots & \vdots & \dots & \vdots \\ \varphi(p,1) & \varphi(p,2) & \dots & \varphi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \varphi(1,0) \\ \varphi(2,0) \\ \vdots \\ \varphi(p,0) \end{bmatrix} \tag{4.29}$$

The equation 4.29 can be represented as

$$\Phi a = \Psi \tag{4.30}$$

Since Φ is symmetric, this system of equations can be solved efficiently using Cholesky decomposition in (p^3).

Figures 4.14 and 4.15 shows the plots of LPCs of five records different target types, namely humpback whale, ship, boat, sealion and snapping shrimp.

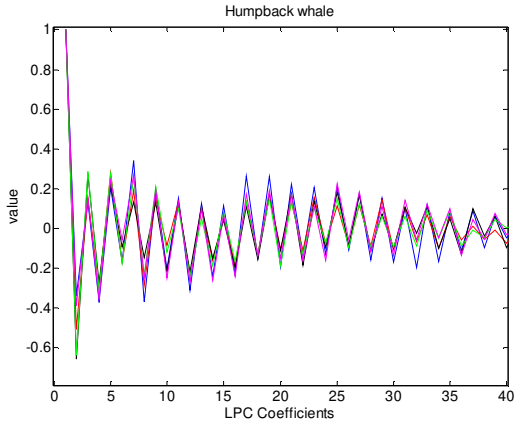

Fig.4.14 Plot of LPCs of Humpback whale noise

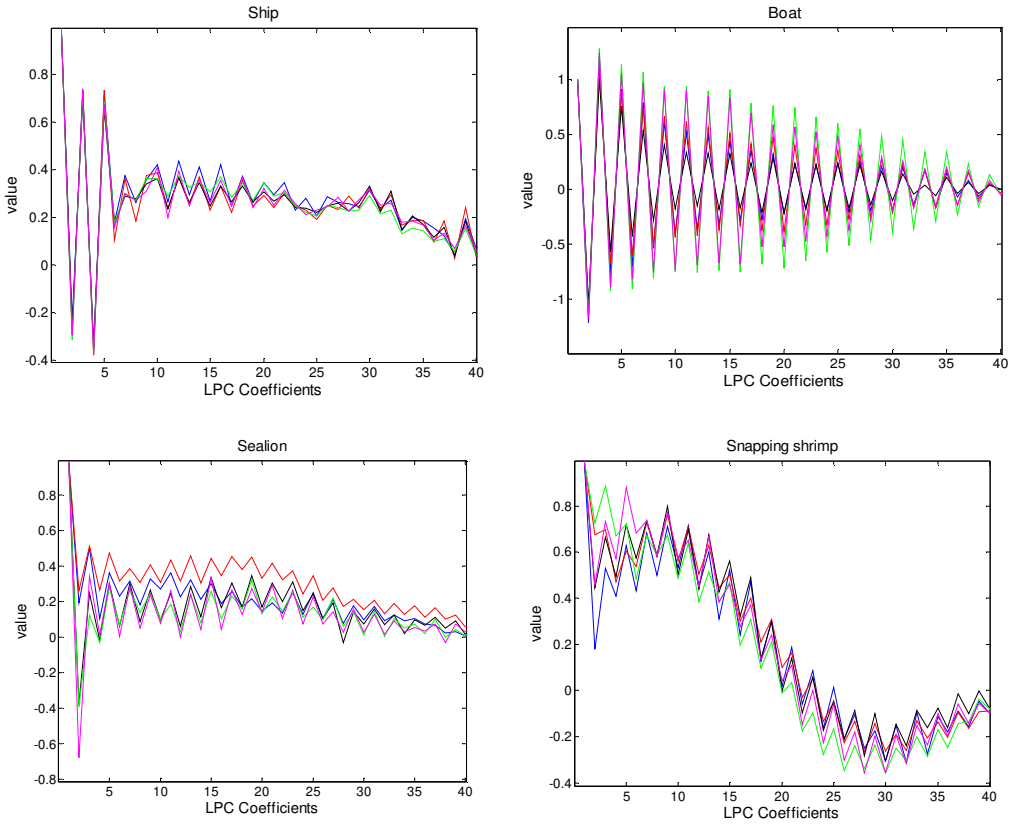

Fig.4.15 Plots of LPCs of different target types - ship, boat, sealion and snapping shrimp

4.7 *Linear Prediction Cepstral Coefficients*

Linear Prediction Cepstral Coefficients (LPCC) are an important representation of speech derived from linear prediction model and is obtained by considering the power series expansion of the logarithmic transfer function ln $H(z)$ in powers of z^{-1}. If all the poles of $H(z)$ are inside the unit circle, ln $H(z)$ can be expressed by equation 4.31.

$$\ln H(z) = C(z) = \sum_{n=1}^{\infty} c_n z^{-n} \qquad 4.31$$

A simple and unique relationship exist between the parameters c_n and a_n which is obtained by substituting equation 4.18 into equation 4.31 and taking derivatives on both sides with respect to z^{-1}.

$$\frac{d}{dz^{-1}} \ln\left[1/\left\{1 - \sum_{k=1}^{p} a_k z^{-k}\right\}\right] = \frac{d}{dz^{-1}} \sum_{n=1}^{\infty} c_n z^{-n} \qquad 4.32$$

which can be simplified to

$$\left\{\sum_{k=1}^{p} k a_k z^{-k+1}\right\} / \left\{1 - \sum_{k=1}^{p} a_k z^{-k}\right\} = \sum_{n=1}^{\infty} n c_n z^{-n+1} \qquad 4.33$$

and rearranged as

$$\sum_{k=1}^{p} k a_k z^{-k+1} = \left(1 - \sum_{k=1}^{p} a_k z^{-k}\right) \sum_{n=1}^{\infty} n c_n z^{-n+1} \qquad 4.34$$

Equating the constant terms and the powers of z^{-1} on the LHS and RHS, we get

$$\begin{aligned} c_1 &= a_1, \\ c_n &= \sum_{k=1}^{n-1} \left(1 - {}^{k}/_{n}\right) a_k c_{n-k} + a_n, \qquad 1 < n < p \\ c_n &= \sum_{k=1}^{n-1} \left(1 - {}^{k}/_{n}\right) a_k c_{n-k}, n > p \end{aligned} \qquad 4.35$$

The coefficients $c_n{}'s$ can be computed from the predictor coefficients and vice versa by equation 4.35. $c_n' s$ are the samples of the Cepstrum and are popularly known as Linear Predictive Cepstral Coefficients (LPCC). The Cepstrum is defined as the inverse Fourier transform of the log magnitude of the Fourier transform. However, for a transfer function with poles only, the

Cepstrum can be obtained directly from the impulse response samples h_n by equation 4.36 or by the predictor coefficients a_k by equation 4.35.

$$c_1 = h_1 \qquad 4.36$$
$$c_n = \sum_{k=1}^{n-1}\left(1 - {}^{k}/_{n}\right)h_k c_{n-k} + h_n, \quad 1 < n$$

LPC and LPCC were developed for speech recognition applications. However, LPCs and LPCCs in conjunction with other features have been used for other acoustic recognition applications, such as audio based event detection[145] and underwater target recognition [86].

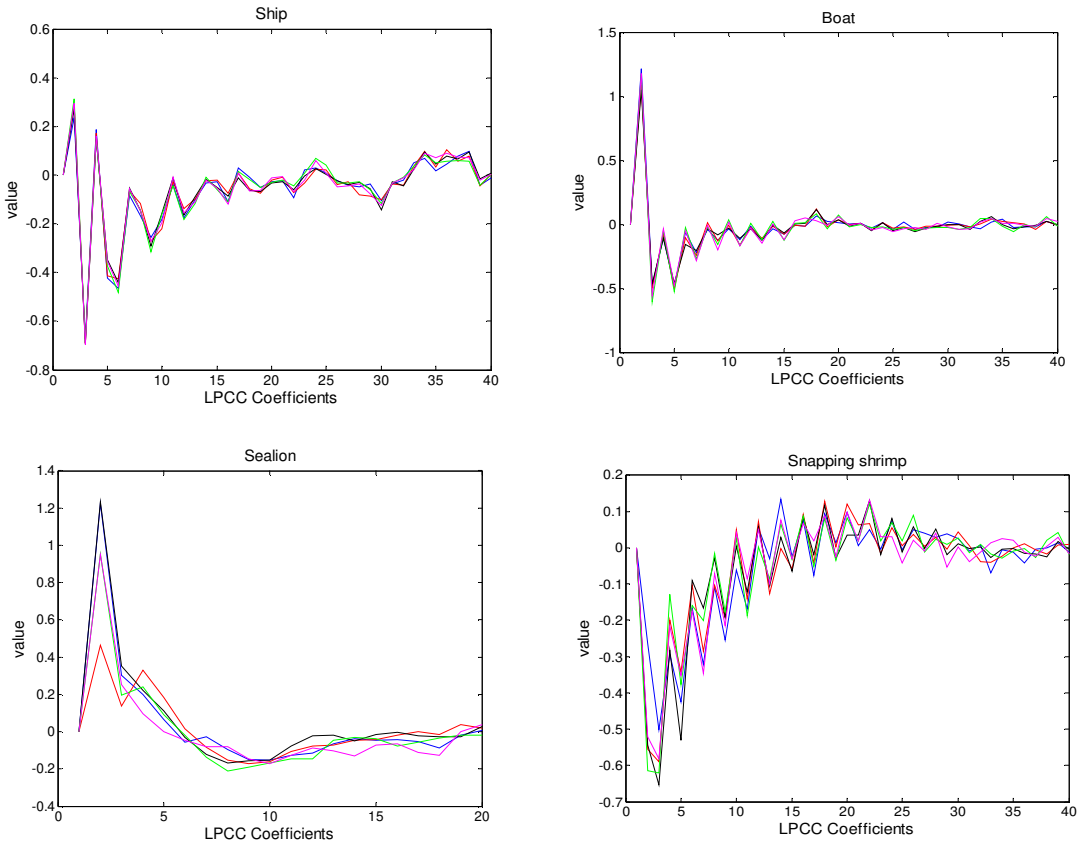

Fig.4.16 Plots of LPCCs of noise of different target types - ship, boat, sealion and snapping shrimp

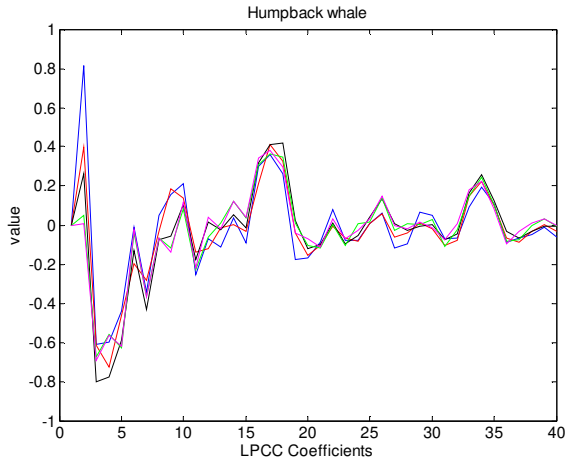

Fig.4.17 Plot of LPCCs of Humpback whale noise

Figures 4.16 and 4.17 shows the plots of LPCCs of five noise records of different target noises, namely ship, boat, sealion and snapping shrimp and humpback whale.

4.8 *Perceptual Linear Prediction (PLP) Cepstral Coefficients*

In Perceptual Linear Prediction (PLP) technique, the properties of human hearing are simulated by practical approximations, and the auditory like spectrum of speech is approximated by an autoregressive pole model. The steps for computing PLP cepstral coefficients is depicted in Figure 4.18.

The first stage in the PLP feature extraction process is to frame the signal and apply a suitable windowing technique. Typically, hamming window, which avoids discontinuities by shrinking the values of signal towards zero at the window boundaries, is applied. Hamming window can be represented by equation 4.37.

$$w(n) = \begin{cases} -0.54 - 0.46\cos\dfrac{2\pi n}{L}, & 0 \le n \le L \\ 0, & \text{Otherwise} \end{cases} \qquad 4.37$$

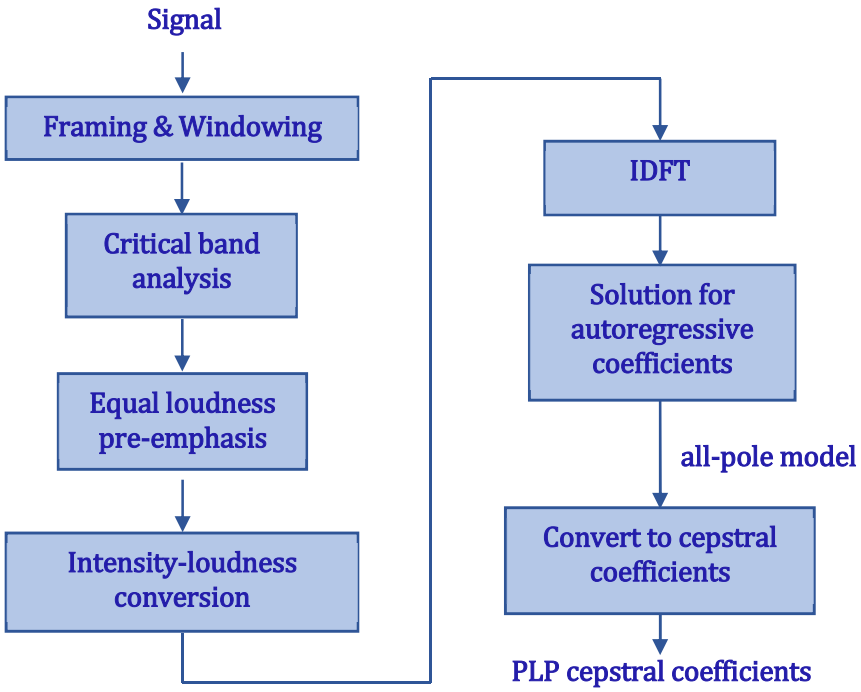

Fig.4.18 Steps in computing PLP cepstral coefficients

The spectral information of the windowed output is computed by taking Fourier transform (discrete Fourier transform or fast Fourier transform).

The next step in computing PLP coefficients is the critical band analysis. Critical band is a concept introduced by Harvey Fletcher [146] and describes the frequency bandwidth of the auditory filter created by the cochlea. Critical band models the change in hearing threshold around a sound as a filter. Psycho-acoustically, the critical bandwidth can be measured through the concept of masking. The critical band is the band of audio frequencies in which the narrow band of noise surrounding the tone causes auditory masking of the tone, when the power of the noise in this band (the critical band) is equal to the power in the tone. Critical bands are of great importance in understanding many auditory phenomena such as perception of loudness, pitch, and timbre.

Bark scale is a non-linear frequency scale which models the resolution of the human hearing system. The bark scale is defined so that the critical bands of human hearing each have a width of one Bark. The Bark scale ranges from 1 to 24 Barks, corresponding to the first 24 critical bands of hearing. The Bark-hertz transformation is given by

$$\Omega(\omega) = 6\ln\left\{\frac{\omega}{1200\pi} + \left[\left(\frac{\omega}{1200\pi}\right)^2 + 1\right]^{0.5}\right\} \qquad 4.38$$

where ω is the angular frequency in rad/s. The first step in the critical band analysis, is to warp the power spectrum calculated along the frequency axis ω into the Bark frequency Ω by equation 4.38. The relation between frequency in Hz, and Bark frequency is shown in Fig.4.19. The warped power spectrum is then convolved with the simulated critical band masking curve $\psi(\Omega)$ given by

$$\psi(\Omega) = \begin{cases} 0 & for \quad \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & for \quad -1.3 \leq \Omega \leq -0.5 \\ 1 & for \quad -0.5 < \Omega < 0.5 \\ 10^{-(\Omega-0.5)} & for \quad 0.5 \leq \Omega \leq 2.5 \\ 0 & for \quad \Omega > 2.5 \end{cases} \qquad 4.39$$

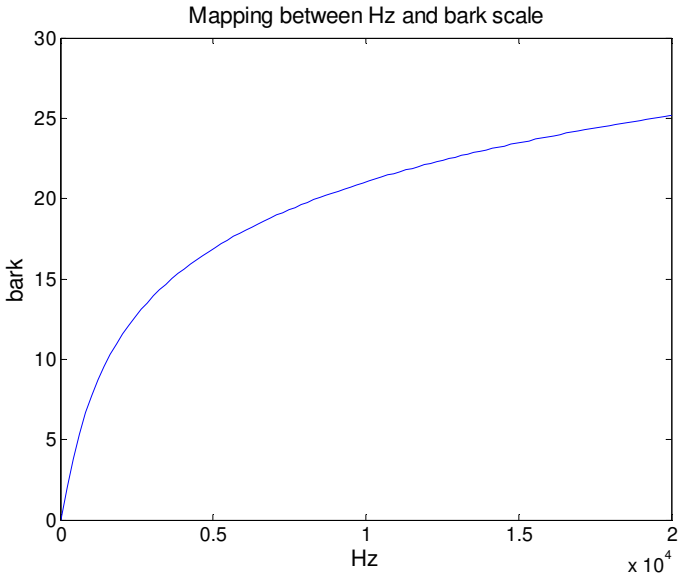

Fig.4.19 The Bark scale

In order to compensate the unequal sensitivity of human hearing at different frequencies, the next processing stage in PLP analysis pre-emphasizes the power spectrum processed by the critical band analysis, by the simulated equal loudness curve. The approximation is given by equation 4.40 and represents a transfer function of a filter with asymptotes of 12 dB/oct between 0 and 400 Hz, 0 dB/oct between 400 and 1200 Hz, 6 dB/oct between 1200 and 3100 Hz, and 0 dB/oct between 3100 Hz and the Nyquist frequency. For moderate sound levels, this approximation is reasonably good up to 5000 Hz.

$$E(\omega) = \frac{[(\omega^2 + 56.8 \times 10^6)\omega^4]}{[(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)]} \qquad 4.40$$

For applications requiring a higher Nyquist frequency, an additional term representing a steep (about - 18 dB/oct), decrease of the sensitivity to hearing for frequencies higher than 5000 Hz is incorporated. Equation 4.40 would then become

$$E(\omega) = \frac{[(\omega^2 + 56.8 \times 10^6)\omega^4]}{[(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)(\omega^6 + 9.58 \times 10^{26})]} \qquad 4.41$$

To simulate the non-linear relation between the intensity of sound and the human perception of loudness, the pre-emphasized signal is subjected to cubic root amplitude compression.

$$\Phi(\Omega) = \Xi(\Omega)^{0.33} \qquad 4.42$$

This operation is an approximation to the power law of hearing and also helps reducing the spectral amplitude variation of the critical-band spectrum so that the all-pole modelling can be done by a relatively low model order.

The next step in PLP analysis, is to approximate $\Phi(\Omega)$ by the spectrum of an all-pole model using the autocorrelation method of all-pole spectral

modelling. The auto regressive coefficients of the all-pole model gives the PLP coefficients. PLP coefficients, like linear prediction coefficients, themselves can be used as acoustic features. However, transforming the PLP coefficients into the cepstral domain yields more robust features. The PLP coefficients can be converted into cepstral coefficients by equation 4.43.

$$
\begin{aligned}
c_1 &= a_{1,} \\
c_n &= \sum_{k=1}^{n-1}\left(1 - {}^{k}/_{n}\right)a_k c_{n-k} + a_n, \qquad 1 < n < p \\
c_n &= \sum_{k=1}^{n-1}\left(1 - {}^{k}/_{n}\right)a_k c_{n-k,} \qquad\qquad n > p
\end{aligned}
\tag{4.43}
$$

Fig.4.21 and Fig. 4.21 shows the plots PLP cepstral coefficients of five records of different target noises, namely ship, boat, sealion, snapping shrimp and humpback whale.

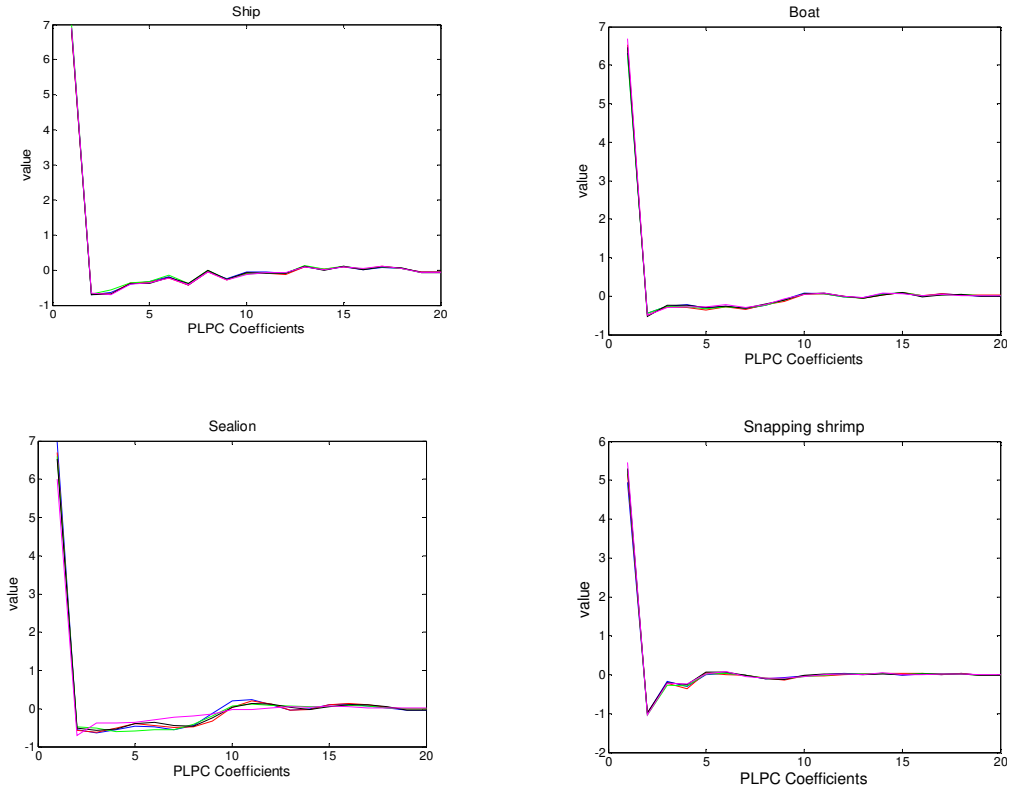

Fig.4.20 Plots of PLP cepstral coefficients of different target types - ship, boat, sealion and snapping shrimp

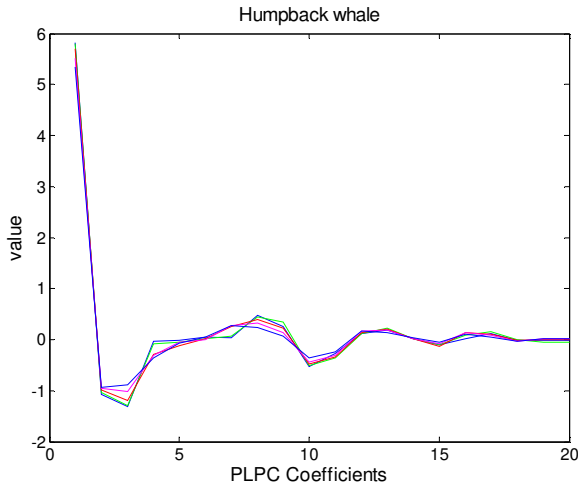

Fig.4.21 Plot of PLP cepstral coefficients of Humpback whale noise

4.9 *Mel Frequency Cepstral Coefficients*

Mel Frequency Cepstral Coefficients (MFCCs) are acoustic features widely used in automatic speech recognition systems. MFCCs are based on the human auditory system in which the perception sensitivity varies with frequency. Extensive studies on human inner ear have shown that basilar membrane, a portion of inner ear, stimulation of different areas of which are perceived as different pitches or tones, can be simulated using a bank of filters [147].

Mel, short for melody, is a psychoacoustic perceptual scale that provides the relation between pitch, perceived frequency, as a function of frequency [148]. The Mel scale, first formulated by Stevens *et al*., is a heuristically derived scale, and attempts to represent the psychological sensation of pitch of the human ear on a linear scale. Stevens *et al*. (1937) [149] organized experiments in which subjects were required to adjust the frequency of a stimulus tone to be half as high as that of a comparison tone. Based on the experimental results, the Mel scale was developed, but with the drawback of being subject dependent. Later in 1940, the original Mel scale

was revised in which differences among the test subjects were resolved. Equal increments in the Mel scale correspond to equal increments of perceived pitch of pure tone stimuli. The Hz to Mel transformation can be achieved by the formula as in equation 4.44 and is shown in Figure 4.22.

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \qquad 4.44$$

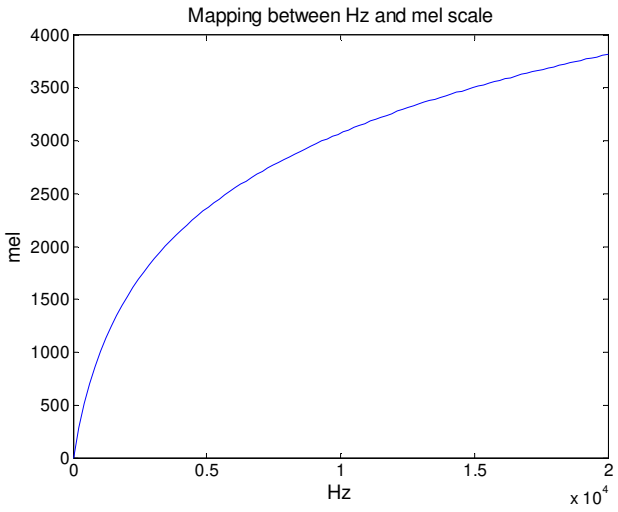

Fig.4.22 The Mel scale

MFCCs are systematically computed by taking the real Cepstrum of a windowed signal derived from the Fast Fourier Transform of Mel scaled signal. The discrete cosine transform of the real logarithm of the energy spectrum expressed on Mel scale gives the Mel Frequency Cepstral Coefficients. The steps for computing MFCCs are depicted in Figure 4.23.

The first stage in the MFCC feature extraction process is pre–emphasis filtering which boosts the energy in the higher frequencies. Typically, in case of speech signals for which MFCC were derived and is normally applied, a spectral tilt exists since more energy is concentrated in the lower frequencies than at the higher frequencies, which is caused by the

nature of excitation of glottal pulse. Therefore, boosting the high frequencies makes information from these higher formants more available to the acoustic model.

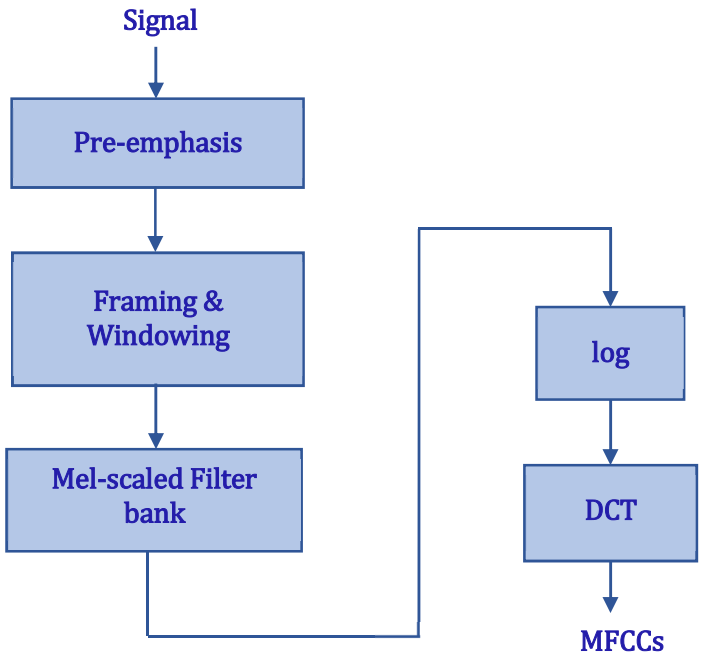

Fig.4.23 Steps in computing MFCCs

The next step in the computation of MFCC is to frame the signal and apply a suitable windowing technique. The spectral information of the windowed output is computed by taking Fourier Transform (discrete Fourier transform or fast Fourier transform) which is then passed through a Mel scaled filter bank. The filter bank is implemented as a set of triangular band-pass filters as shown in Figure 4.24, with spacing and bandwidth determined by a constant Mel-frequency interval. In the spectral domain, it corresponds to a set of non-uniformly spaced filters with more and narrow filters in the low frequency region and less and wide filters in the high frequency region

to account for more discriminative lower frequencies and less discriminative higher frequencies.

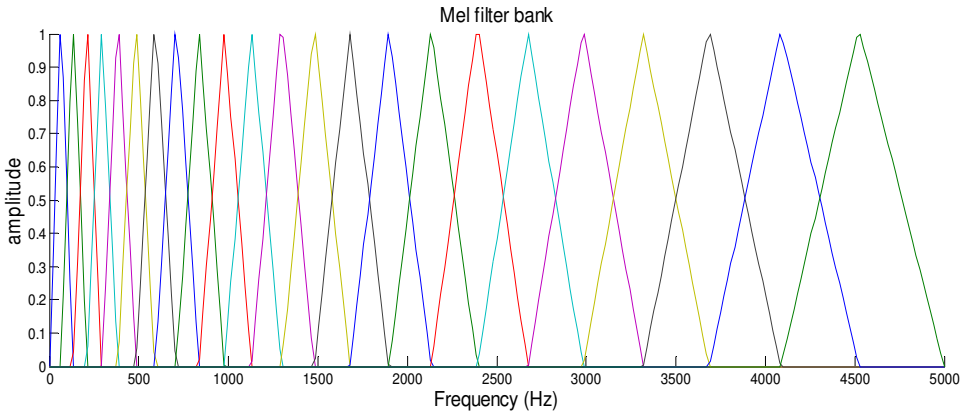

Fig.4.24 Mel filter bank

The next step in computing MFCC is to compute the logarithm of the square magnitude of the output of Mel-filter bank, which leads to compression of the dynamic range. Taking logarithm models the logarithmic sensitivity of human ear to sound amplitudes, as humans are less sensitive to slight differences in amplitude at high amplitudes than at low amplitudes. Therefore taking log of filter energies makes frequency estimates less sensitive to slight variations in input such as power variations.

The final stage in computation of MFCC is to take the Discrete Cosine Transform (DCT) of the log filter bank energies which transforms the log Mel spectrum back into the spatial domain. Even though DFT can also be used for the same purpose, DCT is preferred for its property to concentrate the information to a relatively fewer number of coefficients [150].

Figures 4.25 and 4.26 shows the plots MFCCs of five records of different target noises, namely humpback whale, ship, boat, sealion and snapping shrimp.

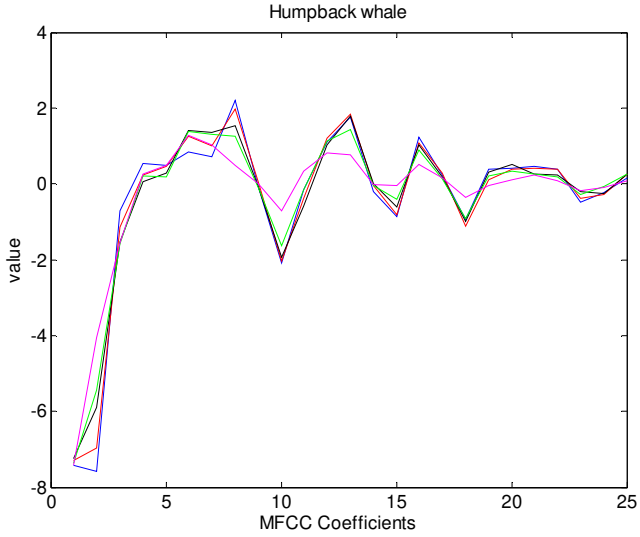

Fig.4.25 Plot of MFCCs of Humpback whale noise

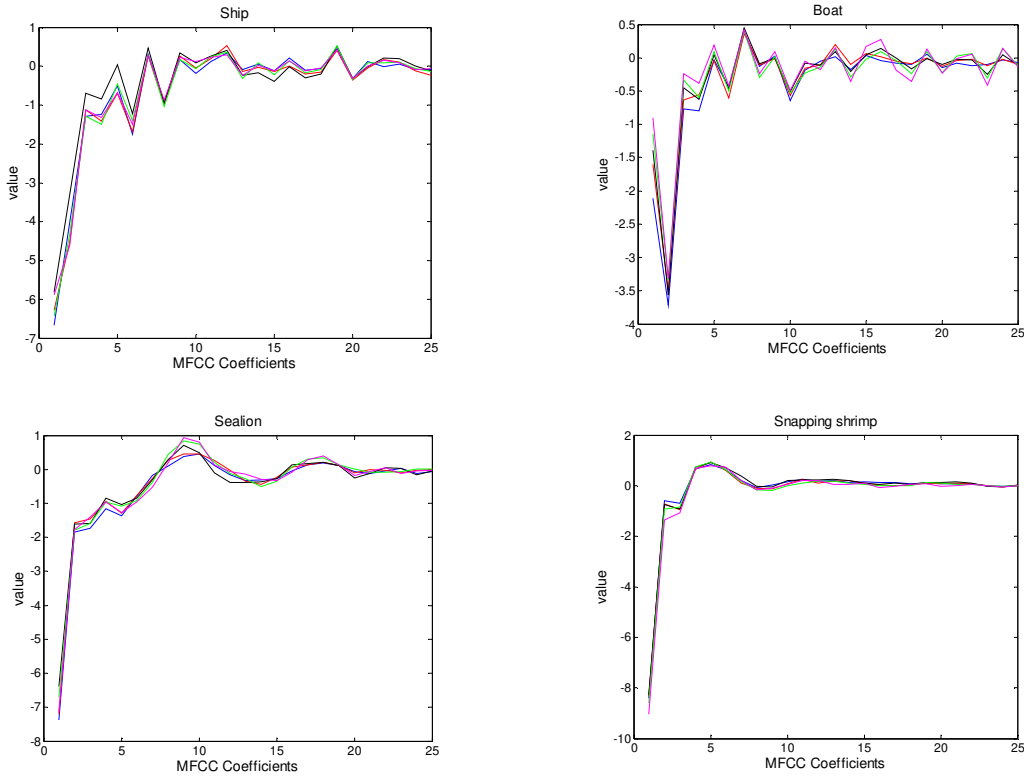

Fig.4.26 Plots of MFCCs of different target types - ship, boat, sealion and snapping shrimp

4.10 *Gammatone Cepstral Coefficients*

The Gammatone cepstral coefficients (GTCCs) are biologically inspired modification to MFCCs in which Gammatone filters equally spaced in the Equivalent Rectangular Bandwidth (ERB) scale is employed. ERB is a psychoacoustic measure, which approximates the bandwidth of the auditory filter at each point along the cochlea as the bandwidth of a rectangular filter, having the same peak transmission as the auditory filter and which passes the same total power for a white noise input [151]. The equation describing the value of ERB as a function of centre frequency, F (in hertz), is

$$\text{ERB} = 24.7(0.00437F + 1) \qquad 4.45$$

As per the above equation, the ERB value at a centre frequency of 1 kHz is approximately 132 Hz, which corresponds to one step of ERB number in the ERB scale. The Hz to ERB transformation can be achieved by the formula as in equation 4.46 and is shown in Figure 4.27.

$$ERB_{scale,1step} = 21.4\,log(0.00437F + 1) \qquad 4.46$$

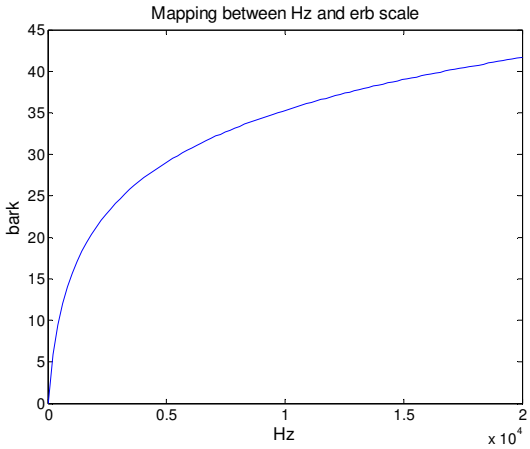

Fig.4.27 The ERB scale

The Gammatone filter bank like the Mel filter bank is a physiologically inspired modelling of the human auditory system in which, the response of the basilar membrane is modelled as a gammatone filter bank with impulse response being the product of a gamma distribution function and a sinusoidal tone centred at frequency f_c as represented in equation 4.47, and is illustrated in Fig.4.28.

$$g(t) = Kt^{(n-1)}e^{-2\pi Bt}\cos(2\pi f_c t + \varphi) \qquad t > 0 \qquad 4.47$$

where *K* is the amplitude factor, *n* is the filter order, *fc* is the centre frequency in Hertz, φ is the phase shift and *B* is the equivalent rectangular bandwidth which represents the duration of the impulse response.

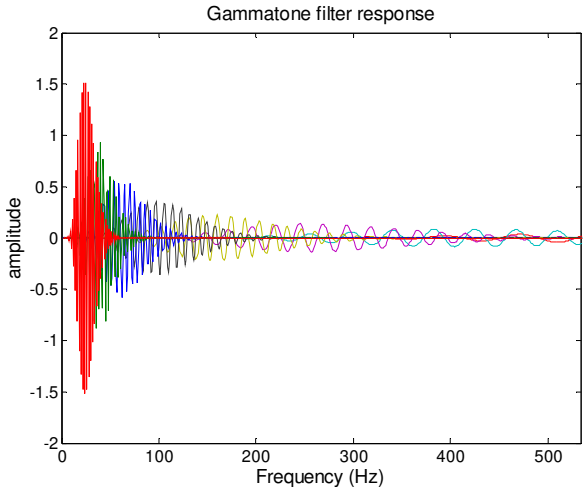

Fig.4.28 Typical time domain response of Gammatone filter bank

The extraction of GTCC is similar to that of MFCC extraction scheme and is depicted as flowchart in Fig.4.29, except that Gammatone filter bank equally spaced in the ERB scale is used instead of triangular filter bank equally spaced in the Mel scale.

Figures 4.30 and 4.31 shows the plots GTCCs of five records of different target noises, namely humpback whale, ship, boat, sealion and snapping shrimp.

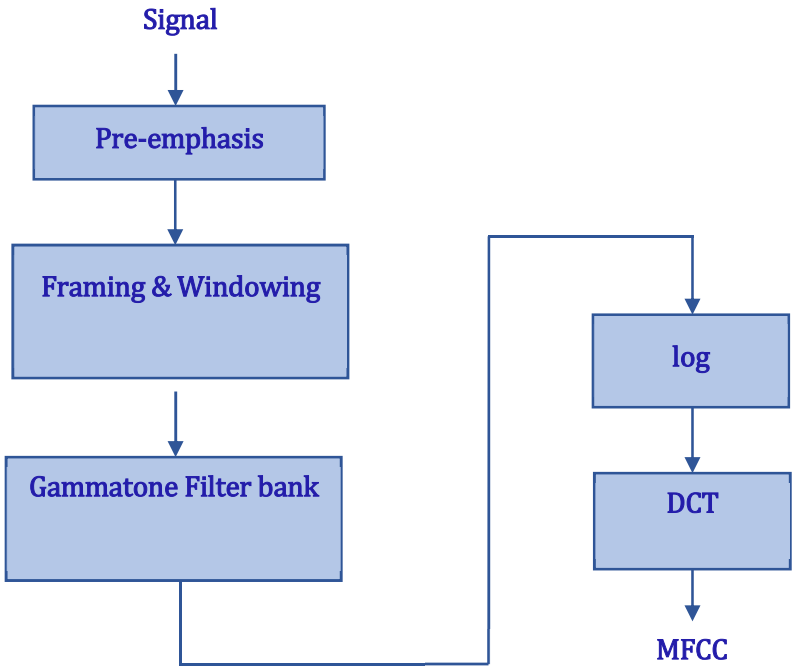

Fig.4.29 Steps in computing GTCCs

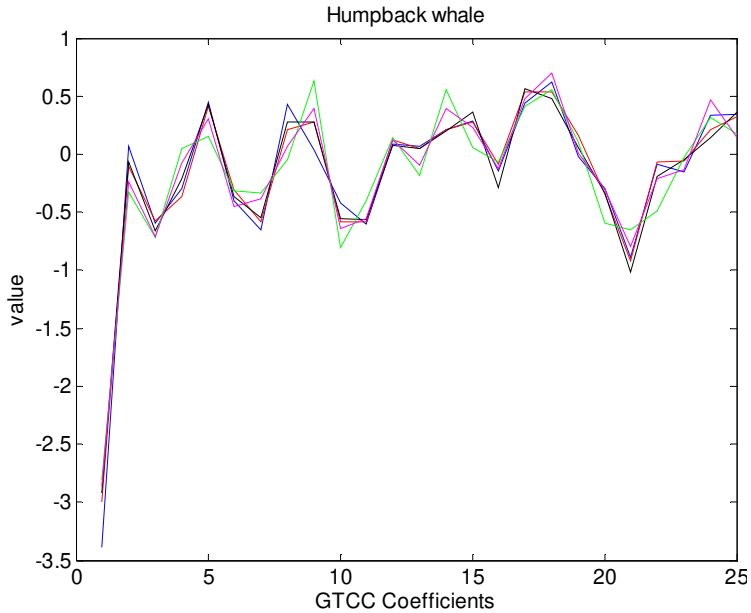

Fig.4.30 Plot of GTCCs of Humpback whale noise

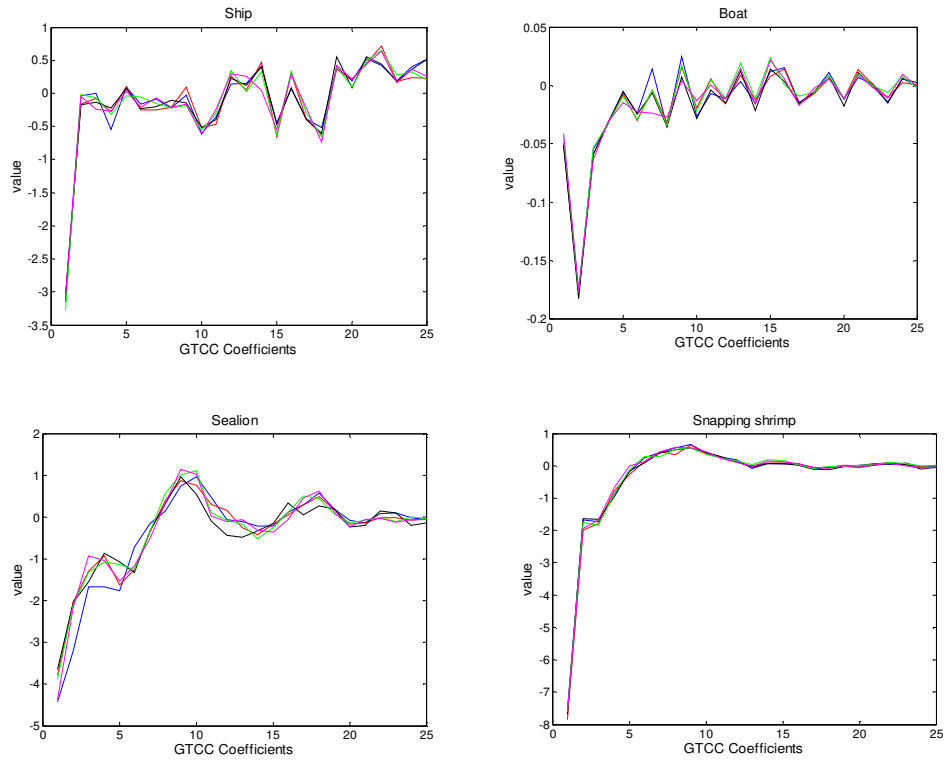

Fig.4.31 Plots of GTCCs of different target types - ship, boat, sealion and snapping shrimp

4.11 *Feature Selection*

The features extracted through the different feature extraction techniques may contain redundant and irrelevant features, which do not, or negatively impact, the performance of the classifier. Such features are identified and removed with the help of feature selection algorithms. Feature selection is a process to identify irrelevant and redundant features that do not contribute to the accuracy of the predictive model, which may be removed to reduce the complexity of the model. The best subset of features contains the least number of dimensions that most contribute to prediction accuracy.

In classifier design, feature selection is an important pre-processing step to avoid noisy, irrelevant and misleading features [152]. Irrelevant,

redundant and noisy features may mislead learning algorithms or cause them to overfit the data. Hence, the obtained classifier in general is less accurate than the one learned from the relevant data. In addition, with the presence of redundant or irrelevant data, it is more likely that the classifier obtained is more complex. A complex classifier tends to be less accurate and un-generalistic compared to a simple classifier. Thus the objective of feature selection is three-fold: to improve the prediction performance of the classifier, to reduce the curse of dimensionality thus reducing the computational burden, and to minimise the chances of overfitting caused by irrelevant data.

Feature selection methods can be broadly categorized into exhaustive search, filter methods, wrapper methods and embedded methods. A brute force feature selection method following an exhaustive search, evaluates all possible combinations of the input features to find the best subset. The computational cost of exhaustive search methods is prohibitively high as its space is $O(2^N)$ and also imposes the danger of overfitting. Hence, exhaustive search techniques are seldom used for feature selection. Filter methods rely on the characteristics of data by applying a statistical measure to assign scoring to each feature. Feature selection is done based on the score obtained by the features. These methods do not consider the effects of the selected feature subset on the performance of the underlying classifier [153]. Wrapper techniques utilize the prediction performance of the underlying classifier to assess the relative usefulness of a feature subset [154]. Embedded methods embed the feature selection with classifier construction, and have the advantages of wrapper models and filter models [153]. They are usually specific to the learning algorithm (classifier).

4.11.1 Filter Methods

Filter methods works on the general characteristics of the data and employs statistical methods such as correlation between the features, and assign scoring to each feature. The features are ranked by their individual scores and are accordingly kept or removed from the feature set. The block diagram for filter method is shown in Figure 4.32.

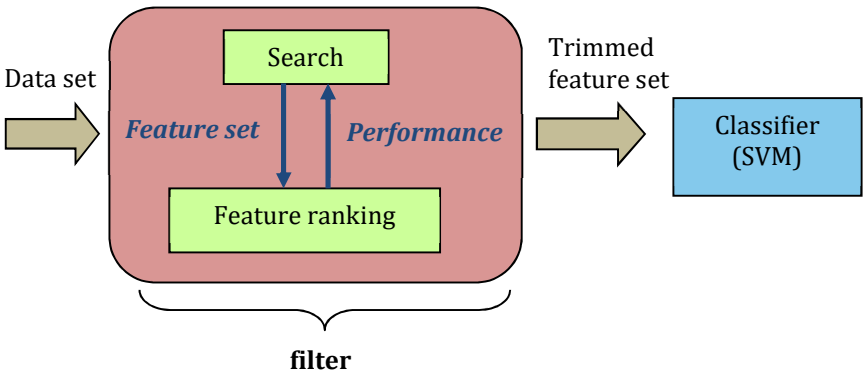

Fig.4.32 General methodology - Filter method

Correlation based feature selection (CFS) is an effective and efficient filter based approach that uses a correlation based heuristic to evaluate the worthiness of features. The heuristic that forms the core of CFS algorithm to evaluate the merit of a subset of features. Good feature subsets contain features highly correlated within the class, yet uncorrelated with other classes. The feature subsets are then ranked depending on their correlation with members of the same class to denote their usefulness in classification. The *m* top ranked features are selected to be retained or removed from the feature set. Eigen vector Centrality based feature Selection (ECS) is a filter method which maps the feature selection problem on an affinity graph where features are the nodes and the importance of nodes are assessed and ranked through Eigen Vector Centrality. The central premise of ECS method is to estimate the importance of a feature as a function of the importance of its

neighbours. Ranking central nodes individuates candidate features, which turn out to be effective from a classification point of view [155]. Filter based approaches are not dependent on classifiers and are usually faster and more scalable than wrapper based methods. In addition, they have low computational complexity since measuring information gains, distance, dependence, or consistency is less complex by time measure, than measuring the performance of a classifier [156]. Also, since filter based approaches rely only on the intrinsic properties of data, the selected features can be used to learn different classifiers.

4.11.2 Wrapper Methods

Wrapper methods are so called because; the feature subset selection algorithm exists as a wrapper around the classification algorithm[157]. For selecting a good feature subset, the wrapper methods use the classification algorithm to evaluate different possible feature subsets. The idea behind the wrapper approach is shown in Figure 4.33.

The classification algorithm is considered as a black box and is run on the dataset partitioned into training and validation sets. The validation set is adjudged through an evaluation metric and the feature set with the highest evaluation metric is chosen.

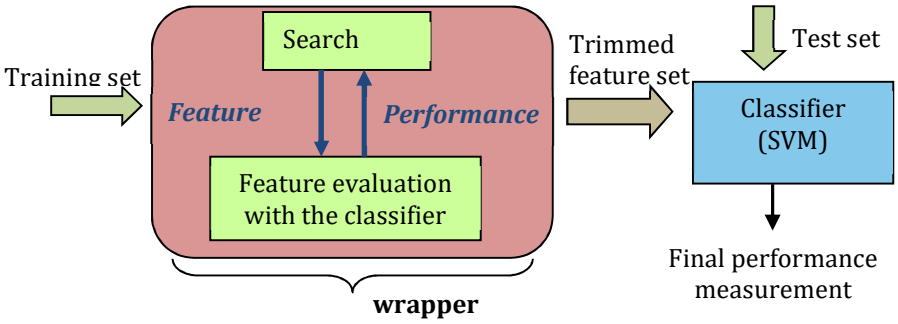

Fig.4.33 General methodology - Wrapper method

The wrapper approaches conducts a search in the space of possible features. The search can be greedy or heuristic in nature. Greedy search techniques include sequential forward selection (SFS), sequential backward selection (SBS), and bi–directional search (BDS). Sequential feature selection methods, learn the usefulness of each feature at each time–step depending on its performance score. Sequential forward selection (SFS) starts with an empty set, incrementally adding features in each step. As each feature is added in, the classifier is evaluated with the feature set and a new feature is retained, only if the performance metric on which the classifier is evaluated has improved [158]. Sequential backward selection (SBS) starts with the full set of features, and removes one of the redundant or irrelevant features at each step. Another method is bi-directional search (BDS), which is a parallel implementation of both SFS and SBS and features are both added and deleted simultaneously until convergence. Sequential techniques have the advantage of picking out features which together work well for classification, as the performance of the classifier is evaluated at each step of the algorithm with different possible combination of features.

Heuristic search based wrapper approaches employ heuristics to search the feature space. These methods do not guarantee the optimal solution but generally arrives at a near optimal solution. They have the advantage of converging faster to near optimal solution but bear the disadvantage of having the tendency of being trapped at a local solution. Meta-heuristic algorithms are a variant of heuristic algorithms, and are strategies which guide the heuristic search towards the optimal solution by not getting trapped at a local solution. Exploitation and exploration – two competing design goals are two key components of a meta-heuristic algorithm and are respectively responsible for local intensification and global diversification. Diversification generates diverse solutions so as to explore

the search space on the global scale, while intensification focuses the search to a local region by exploiting the information that a current good solution is found in this region. The competing design considerations must strike the right balance to derive optimal performance from the algorithm. Different meta-heuristic algorithms employ different degree of exploration and exploitation. A variety of meta-heuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Ant Colony Optimization (ACO) can be used for feature selection. However the most widely employed meta-heuristic algorithm for feature selection is genetic algorithm.

Genetic algorithm (GA) is a meta-heuristic optimization algorithm inspired by the procedures of natural evolution. GA is a population based algorithm, and is capable of effectively exploring large search spaces which is usually required in case of feature selection. Each individual of this population represents a candidate solution to the given problem. Each individual is assigned a fitness score on the basis of a fitness function. GA operates in three stages: selection, crossover and mutation. In the selection step, the best solutions with higher fitness score are selected and are given more chances for reproduction. During the crossover, portions of the parent solutions are exchanged in the hope of generating more adapted solutions. Mutation operates by randomly changing one or more components of a selected individual. Mutation operator introduces diversity in the current solutions, and prevents premature convergence of the algorithm. The population is operated upon by the three GA operators and then re-evaluated until the termination criterion is met. Meta-heuristic search based wrapper methods have the advantage of arriving at near optimal feature subset without an exhaustive search of the feature space thus achieving faster convergence. The trade-off is between optimality and speed and is often

worthwhile because of much gained speed with little loss of optimality in arriving at near optimal solutions.

4.12 *Classification – A Machine Learning Approach*

In machine learning and statistics, classification can be defined as the problem of identifying to which of a set of categories a previously unseen observation belongs to, on the basis of knowledge gained through a training set of data containing observations whose categories are known. Learning is the act of acquiring, modifying or reinforcing existing knowledge or behaviour through experience, study or teaching; which may lead to a potential change in synthesizing information. Learning is built upon and is shaped by previous experience and knowledge. Machine Learning provides computers and related systems the ability to automatically learn and improve from experience without being explicitly programmed.

Designing a machine learning approach involves a number of design choices, including choosing the type of training experience, the target function to be learned, a representation for this target function, and an algorithm for learning the target function from training examples.

The process of machine learning starts with observations or data. Data can be thought of as a collection of instances having an associated label, which is analysed by the learning algorithm to gain knowledge on the process which generated it, and to identify regularities or patterns within. The performance outcome of the learning algorithm on a particular task, measured by a suitable performance metric, generates experience which adds knowledge on the process which created data. Even though the learning algorithm may not completely understand the underlying process, a reasonable approximation of the process could be successfully carried out

which can detect patterns and regularities within data to do a particular task. The niche of machine learning is that even though we may not gain full knowledge about the underlying process or the system, the little know–how we have about the system which is also based on the experience of the learning algorithm at a particular task, helps us to make predictions on the data. According to Mitchell, a learning algorithm learns from expertise concerning some task and performance measure, if its performance at the task improves with expertise [159]. Thus, a well-defined learning necessitates a well-defined task, performance metric, and feedback on training experience.

Machine learning algorithms typically consist of two phases: a learning phase and testing phase. To implement the above two aspects, the algorithm divides the dataset into training data and testing data. In the learning phase depicted in Figure 4.34, the learning algorithm designs a mathematical model of the dependency, which approximates the relationship between the data and outcome, based on the training data given. In the testing phase depicted in Figure 4.35, the models developed by the learning algorithm in the training phase are used to predict the outcome of the data which has not been previously seen by the algorithm.

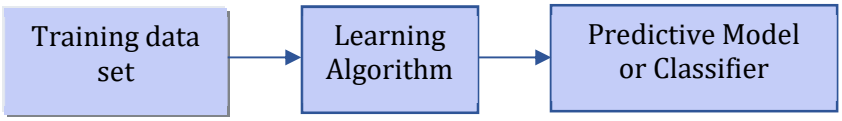

Fig.4.34 Learning phase of machine learning

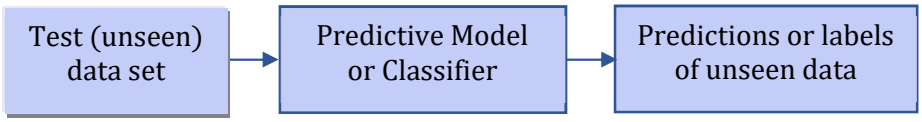

Fig.4.35 Testing phase of machine learning

Machine learning algorithms can be broadly classified into two categories: Unsupervised and Supervised learning algorithms. Unsupervised learning is solely based on the correlations among the input data and is used to find the significant patterns or features in the input data without the help of a teacher. The goal of unsupervised learning is to draw inferences about the underlying structure or distribution in the data. Unsupervised learning does not have labelled data and hence does not involve any target values at the training phase. In case of unsupervised learning, a stopping criterion is needed to terminate the learning phase. Without a stopping criterion, a learning process continues even when a pattern, which does not belong to the training patterns set, is presented to the network. Figure 4.36. depicts the general block diagram of unsupervised learning algorithms. A typical example of unsupervised learning is clustering which is used in exploratory data analysis to find similar hidden patterns or groupings in data.

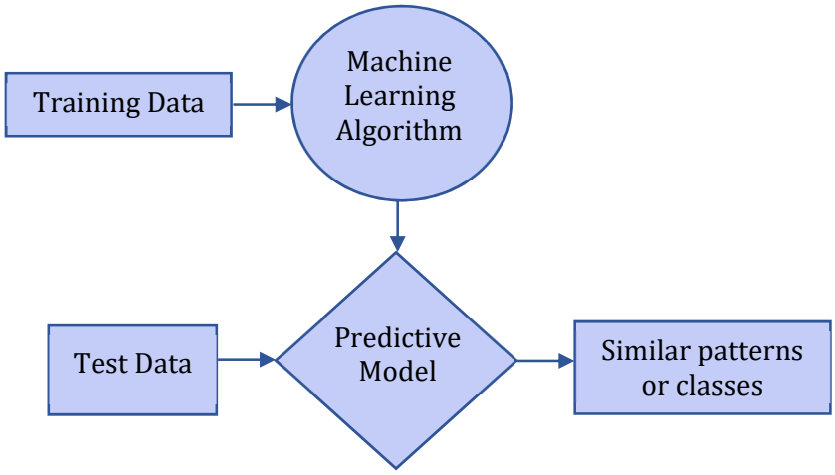

Fig.4.36 General block diagram of unsupervised learning algorithm

Supervised learning algorithms learn a model by inferring a mapping, $y = f(x)$, between labelled training data 'x' with known response y, which makes it capable to make predictions on future unseen data. For a typical classification task, the classifier predicts the output y, $y \epsilon \{1, \ldots\ldots, C\}$, where

C is the number of classes in the classification task. Supervised learning is so called because, the process of the algorithm learning from the training dataset can be thought of as a teacher supervising the learning process.

Figure 4.37 shows the general block diagram of supervised learning algorithms. In the optimal scenario, a large number of labelled training data will allow the algorithm to infer well the mapping between the input and the output, which will enable it to determine the class labels of unseen test data correctly. A lot many learning algorithms such as decision trees, logistic regression, and support vector machines belong to the domain of supervised algorithms.

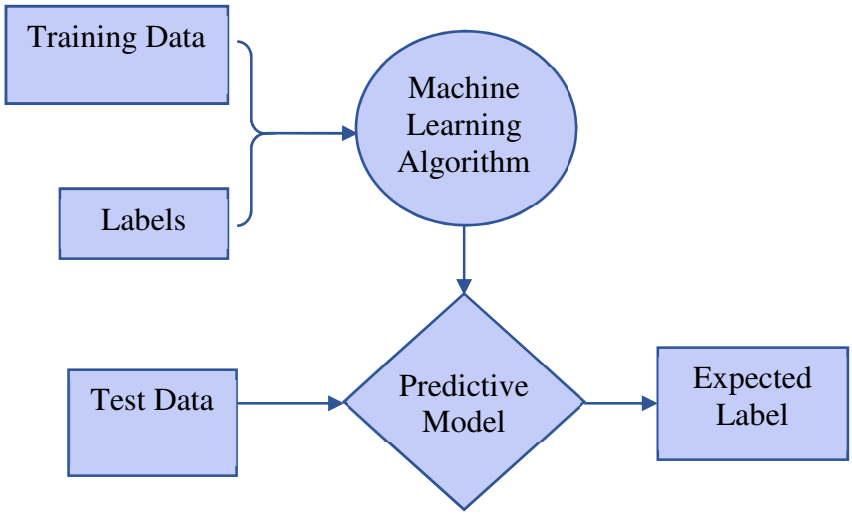

Fig.4.37 General block diagram of supervised learning algorithm

The machine learning field suggests three phases for the design of a supervised learning algorithm: a training phase, validation phase, and testing phase. Hence, it recommends three divisions (or subsets) of the data sets to carry out these tasks. The training dataset is the sample of data used to build the model by inferring the mapping between the input and the output. The validation dataset is used to determine how well the model has been trained,

the estimate of which is used to adjust its parameters. The test dataset is used to assess the performance of a fully-specified classifier. In machine learning, it is also essential to define suitable performance evaluation metrics to train, validate, and test the learning models. The cause of poor performance in machine learning can be attributed to concepts such as under-fitting, and over-fitting. Over-fitting refers to a model that models the training data too well, however, fails to generalise to new data and under-fitting refers to a model that can neither model the training data nor generalise to new data. Over-fitting occurs when the model learns the concepts in the training data along with the noise present in it, to the extent that it negatively impacts the performance of the model on previously unseen data. Underfitting occurs when the model fails to follow the trends in the training data.

An important aspect of machine learning is the bias-variance trade-off. Reducible error incurred by a learning algorithm, is a combination of the error due to squared bias and the error due to variance, and the goal of the designer is to reduce both bias and variance simultaneously. The error due to squared bias is the value by which the expected model prediction differs from the real value or target, over the training data. Bias conveys the ability of the learning model to approximate the data. An overly simplistic model often leads to a solution that is highly biased and does not fit the data. As the complexity of the learning model increases, its ability to approximate and follow the patterns in the data increases, thus keeping the bias low.

Variance refers to the error by which the prediction over one training set, differs from the expected, predicted value, over all the training sets. Variance attributes to the error due to an overly-complex model, that tries to fit the training data as closely as possible, leading to over-fitting. It depends on the extent of training of the classifier, and decreases with rigorously trained classifiers with more training data. However, a model having high

variance when tested on unseen data during the testing phase will not yield satisfactory results, thus exhibiting poor generalisation capabilities. Generalization ability of an algorithm is defined as the ability of the algorithm to predict the outcome of previously unseen data accurately. An algorithm with high variance becomes highly sensitive to high degrees of variation in the training data, since an overfit model closely follows the trends of the training data, including the noise in the training data and hence fails to generalise in the test phase.

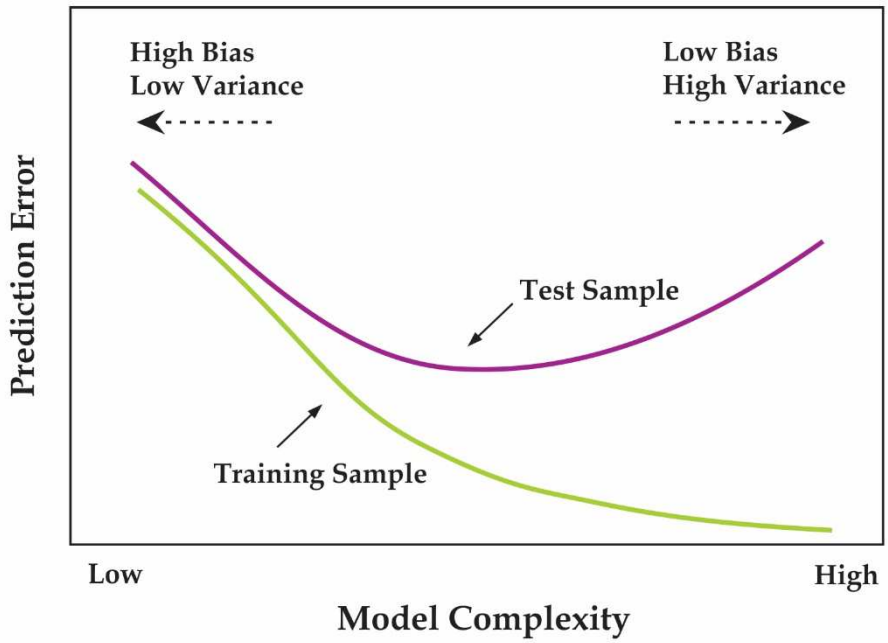

Fig.4.38 Depiction of Bias-variance trade-off

A high variance indicates low training error and high validation error, and high bias implies a high training as well as a high validation error. The designer's goal is to simultaneously reduce bias and variance to obtain the most accurate model feasible. However, a trade-off has to be made when selecting models of differing flexibility or complexity and is referred to as bias-variance trade-off and is depicted in Figure 4.38. Bias is reduced and

variance is increased in relation to model complexity. When the complexity of the model is low, the learning algorithm is too flexible to fit the data well and hence has a high bias. As the complexity of the model increases, with more and more parameters being added, bias steadily falls, and variance becomes the primary concern. The bias-variance trade-off suggests that, a learning algorithm should have the flexibility to fit the data well and at the same time be optimally complex for improved generalisation.

4.13 *Support Vector Machines*

Support Vector Machines (SVM) is a supervised learning algorithm that has stemmed from Statistical Learning Theory (SLT). It is a kernel-based machine learning algorithm which is used for both classification and regression. SVM is based on the Structural Risk Minimization (SRM) induction principle that minimises an upper bound on the expected risk, thus reducing generalisation error. The issues with machine learning algorithms that steered the development of SVMs are the bias-variance trade-off, capacity control and overfitting.

The formulation and constitution of algorithms to overcome the above disadvantages using concepts from SLT has led to the development of SVMs. SVMs have strong generalisation abilities and have advantages in selecting a model with the optimum complexity, and overcoming problems such as overfitting. A detailed description of SLT which has led to the development of SVM based on SRM induction principle has been provided in Appendix. To implement the SRM induction principle in learning algorithms, one has to minimise the value of empirical risk and the capacity factor, to choose a model with appropriate VC dimension. The modelling is achieved in support vector machine by employing a maximal margin

classifier and for non-linearly separable data, the data is mapped to a high dimensional space where they are linearly separable.

4.13.1 Separating Hyperplanes & their Generalization

Consider the problem of minimising empirical risk on the set of linear indicator functions described by equation 4.48.

$$f(x, w) = \theta\{\sum_{i=0}^{n} w_i x^i\}, \quad w \in W \tag{4.48}$$

Let the training set be described as $(x_1, y_1), \ldots\ldots\ldots, (x_l, y_n)$ where each $x_j = \left(x_j^1, \ldots\ldots\ldots, x_j^l\right)$ is a vector belonging to the class $y_j \in \{0,1\}$, $j = 1, \ldots\ldots\ldots\ldots, n$. To minimise the empirical risk, one has to find the parameters $w = (w^1, \ldots\ldots\ldots, w^n)$ (weights) which minimise the empirical risk functional, given by equation 4.49.

$$R_{emp}(w) = \frac{1}{l}\sum_{j=1}^{l}\left(y_j - f\left(x_j, w\right)\right)^2 \tag{4.49}$$

There are several methods for minimising this risk functional. An exact solution can be ensured if the minimum of the empirical risk is zero. A non zero value for the minimum of empirical risk functional will yield only an approximate solution. Therefore, by controlling the weights of the hyperplane, one can control the value of empirical risk. Unfortunately, the set of hyperplanes defined by different weights may not be flexible enough to provide low empirical risk for many real- world problems. However, the flexibility can be increased in one of the following ways [160],

i. selecting a hyperplane with the largest margin
ii. map the input vectors to a higher dimensional feature space which transforms a nonlinear classification problem to a linear one

The above idea has led to the development of Support Vector Machines.

4.13.2 Optimal Separating Hyperplane

Consider a linearly separable training data

$$(x_1, y_1), \ldots\ldots\ldots, (x_n, y_n),\ x \in \mathcal{R}^n,\ y \in \{+1, -1\} \qquad 4.50$$

which can be separated by the set of hyperplanes described by equation 4.51

$$(w^{\mathrm{T}}.x) + b = 0 \qquad 4.51$$

where b is the bias and $w = [w_1, w_2, \ldots\ldots, w_l]$ is the weight vector normal to the hyperplane.

The choice of the hyperplane from the set of hyperplanes described by equation 4.52 should be done in such a way that the resulting classifier can generalise well. The most optimal hyperplane would be the one with the most significant margin and is also known as the maximal margin hyperplane. In other words, maximal margin hyperplane is the hyperplane that has the farthest minimum distance to the closest training vector. A hyperplane with the most substantial margin on the training data can be expected to have a significant margin on the test data, and hence will be able to generalise well on the test data.

4.13.3 Linear SVM

For a linearly separable binary classification problem, the linear separating hyperplane can be described as follows

$$(w^{\mathrm{T}}.x_i) + b > 0 \ \text{if} y_i = 1 \qquad 4.53$$

$$(w^{\mathrm{T}}.x_i) + b < 0 \ \text{if}\, y_i = 0 \qquad 4.54$$

Applying scale transformation on w and b, equations 4.53 and 4.54 are equivalent to

$$(w^{\mathrm{T}}.x_i) + b \geq 1 \ \text{if}\, y_i = 1 \qquad 4.55$$

$$(w^{\mathrm{T}}.x_i) + b \leq -1 \text{ if } y_i = -1 \qquad 4.56$$

which can be alternatively written as

$$y_i[(w^{\mathrm{T}}.x_i) + b] \geq 1, \qquad i = 1, \ldots\ldots n \qquad 4.57$$

The margin *M*, which is the distance from the hyperplane to the closest data point can be derived as

$$M = \frac{2}{\|w\|} \qquad 4.58$$

The support vectors can be defined as those data points the margin pushes against and which satisfies the equality in equation 4.57 [92]. Only, these data points influence in determining the position of the hyperplane. This is illustrated in Figure 4.39.

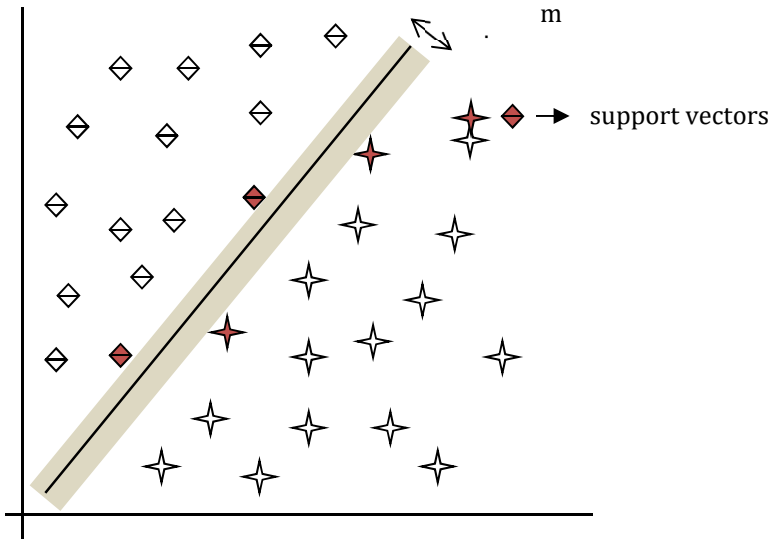

Fig.4.39 Maximal margin linear classifier

For obtaining a maximal margin classifier, the margin *M* described by equation 4.58, has to be maximised which is equivalent to minimising $\frac{\|w\|^2}{2}$, subject to the constraint expressed by equation 4.57. Thus we have a quadratic optimization problem subject to constraints specified by a linear

inequality. The solution to this optimization problem is given by the saddle point of the Lagrange functional (Langrangian). Introducing Lagrange multipliers, $\alpha_i \geq 0, i = 1, \ldots\ldots, n$, the Lagrangian for the above optimization problem can be formulated as

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(w^T x_i + b) - 1\right) \quad 4.59$$

The objective is to find the saddle point of the above Lagrangian which is achieved through its dual formulation. The Karush-Kuhn-Tucker (KKT) conditions specify the requirements to be satisfied by an optimal solution to a general optimisation problem [161]. Given, a primal problem defined by equation 4.59, KKT conditions require, $\mathcal{L}(w, b, \alpha)$ to be minimised with respect to *w*, *b* and maximized over the Lagrangian multipliers, α_i.

$$\frac{\partial}{\partial b} L(w, b, \alpha) = 0 \quad 4.60$$

$$\frac{\partial}{\partial w} L(w, b, \alpha) = 0 \quad 4.61$$

Equations 4.60 and 4.61 gives,

$$\sum_{i=1}^{n} \alpha y_i = 0 \quad 4.62$$

$$w = \sum_{i=1}^{n} \alpha y_i x_i \quad 4.63$$

substituting equations 4.62 and 4.63 into equation 4.59, the dual Lagrangian can be formulated as

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle \quad 4.64$$

The dual optimisation problem is to maximise the dual Lagrangian L_D subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and} \alpha_i \geq 0 \qquad 4.65$$

The dual optimisation problem can be solved using standard programming techniques such as sequential minimal optimisation (SMO) by which we get the coefficients $\alpha_i, i = 1, \ldots\ldots\ldots, n$ from which we can calculate

$$w = \sum_{i=1}^{n} \alpha y_i x_i \qquad 4.66$$

and the classifier can be expressed as

$$f(x_{new}) = \text{sgn}(\sum_{i=1}^{n} \alpha y_i \langle x, x_i \rangle + b) \qquad 4.67$$

4.13.4 Soft Margin Classifier

In most practical applications, training data would be linearly inseparable. Considering the case when the data is almost linearly separable except for some outliers as shown in Figure 4.40.

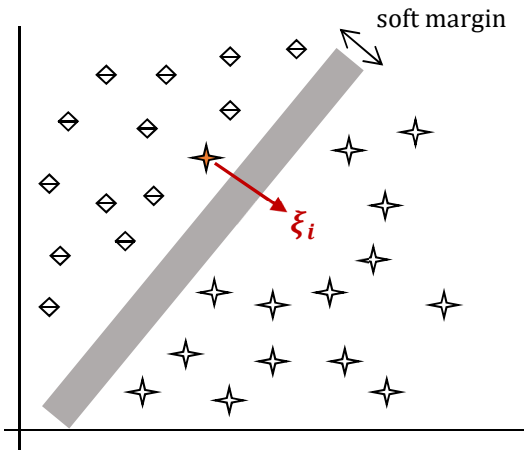

Fig.4.40 Soft Margin Classifier when data is almost linearly separable except for outliers

In such cases, opting a hard-margin classifier as in Figure 4.41 will lead to poor generalisation and hence the hard-margin SVM needs to be modified so that it can cater for the misclassification.

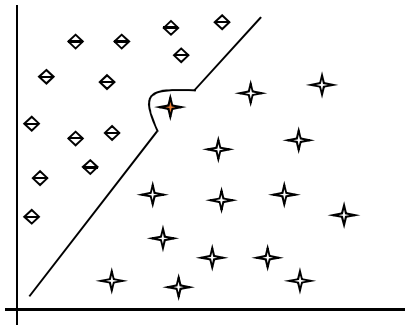

Fig.4.41 Hard Margin Classifier when data is almost linearly separable with outliers

A soft-margin classifier which introduces slack variables in the objective function, that relax the constraint in equation 4.57 is employed. The slack variables define the cost at which each outlier can be moved to its original position. The overall goal of the optimisation is now to find the hyperplane, such that the number of misclassified instances is minimised. The relaxed constraints with slack variables now becomes

$$wx_i + b \geq 1 - \xi_i, \qquad y_i = +1 \tag{4.68}$$

$$wx_i + b \leq -1 - \xi_i, \qquad y_i = -1 \tag{4.69}$$

and the objective function takes the form

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{4.70}$$

such that

$$y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, \ldots\ldots\ldots, n \tag{4.71}$$

A penalty parameter C is included in the objective function described by equation 4.69, which controls the width of the soft margin and also determines the trade-off between the training error and the VC dimension of the model [92]. Introducing Lagrange multipliers, $\alpha_i,\ \beta_i \geq 0, i = 1, \ldots\ldots, n$, the Lagrangian for the above optimization problem can be formulated as

$$L(w,b,\xi,\alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\,(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \quad 4.72$$

The objective is to find the saddle point of the above Lagrangian which is achieved through its dual formulation. For the primal problem specified by equation 4.72, KKT conditions require, $\mathcal{L}(w,b,\alpha)$ to be minimised with respect to *w*, *b*, ξ and maximized over the Lagrangian multipliers, α_i.

$$\frac{\partial}{\partial b}L(w,b,\xi,\alpha) = 0 \quad 4.73$$

$$\frac{\partial}{\partial w}L(w,b,\xi,\alpha) = 0 \quad 4.74$$

$$\frac{\partial}{\partial \xi}L(w,b,\xi,\alpha) = 0 \quad 4.75$$

Equations 4.73, 4.74 and 4.75 gives,

$$\sum_{i=1}^{n}\alpha y_i = 0 \quad 4.76$$

$$w = \sum_{i=1}^{n}\alpha y_i\, x_i \quad 4.77$$

$$\alpha_i = C - \beta_i \quad 4.78$$

substituting equations 4.76, 4.77, and 4.78 into equation 4.72 the dual Lagrangian can be formulated as

$$L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{\substack{i=1,\\ j=1}}^{n}\alpha_i\alpha_j y_i y_j\langle x_i x_j\rangle \quad 4.79$$

The dual optimisation problem is to maximise the dual Lagrangian L_D subject to

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \text{ and } 0 \le \alpha_i \le C \quad 4.80$$

4.13.5 Non-Linear SVM

The soft margin approach works well for data which is close to linearly separable. However, when the data is not linearly separable, and a non-linear decision surface becomes mandatory, SVM's map the input data $x_i \in \Re^m$ into vectors $\phi(x_i) \in \Re^s$ of a higher dimensional space where it can be linearly separated.

$$x \in \Re^m \rightarrow \phi(x) = [\phi_1(x), \phi_2(x), \ldots\ldots\ldots, \phi_s(x)]^T \in \Re^s \qquad 4.81$$

where ϕ represents the mapping: $\Re^m \rightarrow \Re^s$

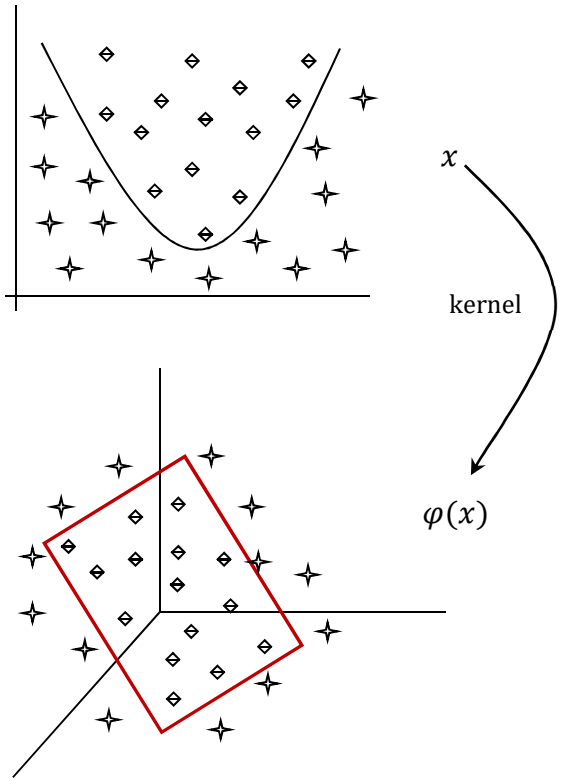

Fig.4.42 Kernel trick by which non-linearly separable data is transformed to linearly separable data in a higher dimensional space

SVM's, employ an easy and efficient way of mapping the data to a higher dimensional space, which is often referred to as the 'kernel trick'. By equation 4.64 the linear SVM classifier depends on the inner product $\langle x_i, x_j \rangle$between the data point vectors. When the data points are transferred to the high dimensional feature space where a linear classifier can be employed, via a mapping function ϕ the inner product becomes$\langle \phi(x_i)^T \phi(x_j) \rangle$. This is as depicted in Figure 4.42.

To define the non-linear SVM classifier in the dual formulation, it is sufficient to know the inner product $\langle \phi(x_i)^T \phi(\mathrm{x_j}) \rangle$ and the explicit mapping function need not be known. The inner product in equation 4.64 has to be replaced by the inner product $\langle \phi(x_i)^T \phi(x_j) \rangle$ to define the non-linear SVM. The inner product $\langle \phi(x_i)^T \phi(x_j) \rangle$is called the kernel function and is denoted as

$$K(x_i, x_j) = \langle \phi(x_i)^T \phi(x_j) \rangle \qquad 4.82$$

The non-linear SVM classifier can be expressed as

$$f(x_{new}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha y_i K(x_i, x_j) + b\right) \qquad 4.83$$

The kernel function can be defined as a function that corresponds to a dot product of two vectors in some expanded feature space. In SVM formulation, using the kernel function, data in the original input space can be easily carried to a higher dimensional feature space. For a given function $K(x_i, x_j)$ to be a kernel (i.e. a dot product in an expanded feature space), the function must satisfy Mercer's conditions [93]. Mercer's theorem states that for K to be a valid kernel function, the kernel matrix K must be symmetric and positive semi-definite. By Mercer's theorem, a symmetric function $K(x_i, x_j)$ can be expressed as a kernel

$$K(x_i, x_j) = \langle \phi(x_i)^T \phi(x_j) \rangle \qquad 4.84$$

for some ϕ if and only if $K(x_i, x_j)$ is positive semidefinite i.e.

$$\int K(x,y)g(x)g(y)dxdy \geq 0 \qquad \forall g \tag{4.85}$$

or, equivalently

$\begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \dots\dots\dots\dots \\ K(x_2,x_1) & \ddots & \vdots \\ \vdots & \dots\dots\dots\dots & \cdot\cdot \end{bmatrix}$ is positive semidefinite for any $\{x_1, \dots\dots\dots\dots, x_n\}$

The kernels often investigated for the pattern recognition problem that are suitable for most common settings are listed in Table 4.1.

The linear kernel results in a classifier which has a linear decision surface and the polynomial kernel results in a polynomial decision surface. The Gaussian kernel results in a Gaussian RBF classifier and the multilayer perceptron kernel results in a sigmoidal network.

Table 4-1 Commonly used Kernels

Type of Kernel	Inner Product
Linear Kernel	$K(x_i, x_j) = x_i^T x_j$
Polynomial Kernel	$K(x_i, x_j) = (1 + x_i^T x_j)^p$
Gaussian (Radial Basis Function (RBF)) Kernel	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
Multi Layer Perceptron (MLP)	$K(x_i, x_j) = \tanh(p_1 . x_i x_j + p_2)$

4.14 *Multiclass SVM*

SVM's are originally defined as binary classifiers. The binary SVM classifiers are extended by different algorithms to solve the multiclass problem. Algorithms for solving multiclass problems are built upon the

binary classifier by reducing the multiclass problem to multiple binary classification problems. Two conventional approaches adopted for multiclass SVM classification are one-against-one (1-a-1) and one-against-all (1-a-a) approach.

Knerr *et al*. proposed the one-against-one classifier [94] as a stepwise building procedure with single layer training, as an alternative to multilayer neural networks. The algorithm was later adopted to solve multiclass SVM problems. The algorithm, also referred to as pairwise classification algorithm, works by creating SVM classifiers for all possible pair of classes as depicted in Figure 4.43. Each unseen example is classified to the class that 'wins' most binary classifications, in the sense that one which gets the highest number of votes. A voting scheme is employed in which each class gets a vote when the classifier assigns a particular instance to the class.

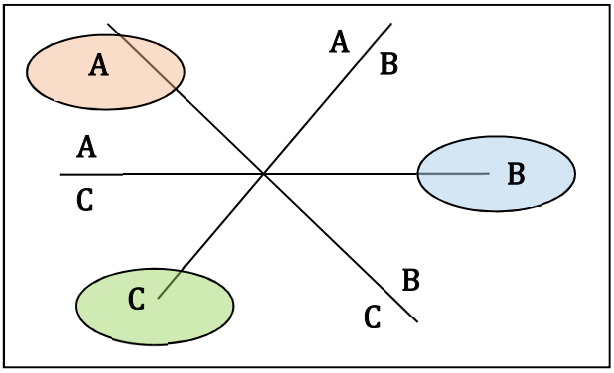

Fig.4.43 One-against-one approach for multiclass problems

For a *K*-class problem, the one-against-one algorithm creates $K(K-1)/2$ SVM classifiers. When an unseen example is classified, all SVM's are evaluated, and the unseen example is classified to the class that wins, i.e. the one with the most votes. The advantage of the one-against-one classifier is that very complex decision boundaries can be realised and an unseen example previously misclassified by one binary SVM, still has a chance of

being correctly classified as there are K-1 binary models per class. The distinct disadvantage with this approach is that when the number of classes, K, is considerable, the number of binary SVMs required, $K(K\text{-}1)/2$, will become exorbitantly large which results in a slower system.

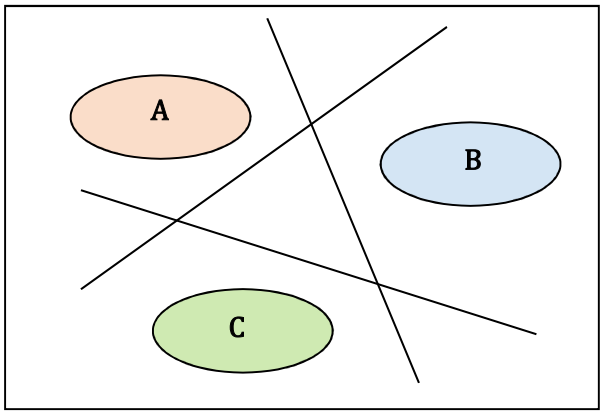

Fig.4.44 One-against-one approach for multiclass problems

The one-against-all classifier proposed by Vapnik [95] constructs K separate binary classifiers for K-class classification as in Figure 4.44. Each classifier attempts to build a decision boundary between itself and the rest. The n-th binary classifier is trained using the data from the n-th class as positive instance and the remaining K-1 classes as negative instances. During the testing phase, a particular example is assigned to a particular class in which the performance metric of the binary classifier gives the maximum output value. The one-against-all approach is faster than the one-against-one approach as only K classifier models need to be built. However, complex decision boundaries cannot be realised with one-against-all as with one-against-one approach.

4.15 *Results and Discussions*

A non-linear multiclass SVM classifier using 1-a-a approach was developed to classify 11 classes of underwater acoustic targets. The targets include the noises of a humpback whale, noises of 4 ships ($ship_1$, $ship_2$, $ship_3$, $ship_4$), sounds of 4 boats ($boat_1$, $boat_2$, $boat_3$, $boat_4$), sound of sea lion and noise of snapping shrimps. To select the best features for classification, the performance of the classifier was tested with different elements and is discussed in the following sections. The range of parameters of different kernels, such as polynomial order, MLP parameter values, RBF sigma value, chosen for the experiments, was by trial and error method, in which the parameter values which exhibited better performance over others are tabulated.

4.15.1 Performance with time domain and frequency domain features

The time domain and frequency domain features were combined into a feature vector with which the classifier was evaluated. The feature vector is formed by combining the average values of energy, energy entropy and spectral entropy as well as ten frame values, each, of the spectral centroid, spectral roll-off and spectral flux leading to a total length of 33. The rationale behind incorporating the average values of energy, energy entropy and spectral entropy is that, they do not change much over time, as can be observed in Figures 4.5, 4.6 and 4.10.

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with the feature vector formed from time and frequency domain features are measured. The different kernels used are polynomial, multilayer perceptron (MLP) and radial basis function (RBF). The average value of F-scores of 11-classes and the overall accuracy,

obtained with time and frequency features for kernel parameters such as, for different values of polynomial order (p) in case of the polynomial kernel, different values of MLP parameters (p_1, p_2) in case of MLP kernel and different values of sigma (σ) in case of RBF kernel are listed in Table 4-2,Table 4-3 and Table 4-4.

Table 4-2 Performance obtained with time domain and frequency domain features with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	**F-score**	**Accuracy (%)**
2	0.57	50.9
3	0.58	52.7
4	**0.62**	**54.9**
5	0.60	53.4
6	0.59	52.3
7	0.57	51.2

Table 4-3 Performance obtained with time domain and frequency domain features with MLP kernel for different values of MLP parameters

MLP Kernel		
[p_1,p_2] values	**F-score**	**Accuracy (%)**
[0.001,-0.001]	0.42	38.0
[0.002,-0.002]	0.47	40.7
[0.003,-0.003]	**0.48**	**42.9**
[0.004,-0.004]	0.46	41.1
[0.005,-0.005]	0.46	40.3

Table 4-4 Performance obtained with time domain and frequency domain features with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	F-score	Accuracy (%)
1	0.58	52.7
2	0.60	54.1
3	**0.62**	**56.3**
4	0.62	55.2
5	0.60	54.2
6	0.58	53.0

4.15.2 Performance with LPCs

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with the feature vector formed from LPCs are evaluated. The appropriate feature vector length was found out by trial and error. The performance results, the average F-score of eleven classes and accuracy, with different feature vector length are tabulated in Table 4-5.

Table 4-5 Classification performance obtained with LPC features of different length

Vector Size	F-score	Accuracy (%)
10	0.48	42.9
20	0.49	49.1
30	0.49	50.9
40	**0.69**	**63.3**
50	0.59	58.9
60	0.61	56.0
Kernel – RBF, RBF Sigma Value - 4		

It can be inferred that, the best feature vector length of LPC for the given classification task is 40. Hence, a feature vector of length 40 was used for evaluating the classification performance. The different kernels used are polynomial, multilayer perceptron (MLP) and radial basis function (RBF).

The average value of F-scores of 11-classes and the overall accuracy, obtained with LPC for kernel parameters such as, different values of polynomial order (p) in case of polynomial kernel Table 4-6, different values of MLP parameters ($p_{1,}\ p_{2,}$) in case of MLP kernel Table 4-7 and different values of sigma (σ) in case of RBF kernel are listed in Table 4-68.

Table 4-6 Performance obtained with LPCs with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	F-score	Accuracy (%)
2	0.62	55.6
3	0.62	56.0
4	0.63	57.1
5	**0.64**	**58.2**
6	0.63	57.4
7	0.61	56.7

Table 4-7 Performance obtained with LPCs with MLP kernel for different values of MLP parameters

MLP Kernel		
[p_1,p_2] values	F-score	Accuracy (%)
[0.001,-0.001]	0.46	41.4
[0.002,-0.002]	0.51	44.7
[0.003,-0.003]	**0.47**	**43.6**

[0.004,-0.004]	0.47	42.2
[0.005,-0.005]	0.44	38.9

Table 4-8 Performance obtained with LPCs with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	F-score	Accuracy (%)
1	0.65	60.5
2	0.67	61.8
3	0.68	62.6
4	**0.69**	**63.3**
5	0.67	62.5
6	0.67	61.0
7	0.66	59.0

4.15.3 Performance with LPCCs

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with feature vector formed from LPCCs, is evaluated. The appropriate feature vector length was found out by trial and error. The performance results, the average F-score of eleven classes and accuracy, with different feature vector length are tabulated in Table 4-9. From Table 4-9, it can be inferred that, the best feature vector length of LPCCs for the given classification task is 40. Hence, a feature vector of length 40 was used for evaluating the performance of the proposed SVM based target classifier. The different kernels used are polynomial, multilayer perceptron (MLP) and radial basis function (RBF).

Table 4-9 Classification performance obtained with LPCCs features of different length

Vector Size	F-score	Accuracy (%)
10	0.52	47.6
20	0.59	52.7
30	0.66	60.7
40	**0.71**	**65.4**
50	0.63	57.4
60	0.58	51.6
Kernel – RBF, RBF Sigma Value - 3		

The average value of F-scores of 11-classes and the overall accuracy, obtained with LPCC for kernel parameters such as, different values of polynomial order (p) in case of polynomial kernel, different values of MLP parameters (p_1, p_2) in case of MLP kernel and different values of sigma (σ) in case of RBF kernel are listed in Table 4-10, Table 4-11 and Table 4-12

Table 4-10 Performance obtained with LPCCs with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	F-score	Accuracy (%)
2	0.64	58.5
3	0.65	59.6
4	**0.66**	**60.3**
5	0.64	59.2
6	0.64	58.2
7	0.64	57.4

Table 4-11 Performance obtained with LPCCs with MLP kernel for different values of MLP parameters

MLP Kernel		
$[p_1,p_2]$ values	F-score	Accuracy (%)
[0.001,-0.001]	0.48	42.5
[0.002,-0.002]	0.49	44.0
[0.003,-0.003]	**0.55**	**49.4**
[0.004,-0.004]	0.51	45.4
[0.005,-0.005]	0.50	44.3

Table 4-12 Performance obtained with LPCC with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	F-score	Accuracy (%)
1	0.68	61.8
2	0.69	63.2
3	**0.71**	**65.4**
4	0.68	62.6
5	0.67	61.2
6	0.66	60.5
7	0.65	59.2

4.15.4 Performance with PLP Cepstral coefficients

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with the feature vector formed from

PLP cepstral coefficients are evaluated. The appropriate feature vector length was found out by trial and error. The performance results, the average F-score of eleven classes and accuracy, with different feature vector length are tabulated in Table 4-13.

Table 4-13 Classification performance obtained with PLP cepstral coefficients of different length

Vector Size	F-score	Accuracy (%)
10	0.66	61.0
15	0.69	63.6
20	**0.71**	**65.8**
25	0.68	62.0
30	0.65	59.0
35	0.62	56.0
Kernel – RBF, RBF Sigma Value - 4		

From Table 4-13, it can be inferred that, the best feature vector length of PLP cepstral coefficients for the given classification task is 20. Hence, a feature vector of length 20 was used for evaluating the performance of the proposed SVM based target classifier. The different kernels used are polynomial, multilayer perceptron (MLP) and radial basis function (RBF). The average value of F-scores of 11-classes and the overall accuracy, obtained with PLP cepstral coefficients for kernel parameters such as, different values of polynomial order (p) in case of polynomial kernel, different values of MLP parameters (p_1, p_2) in case of MLP kernel and different values of sigma (σ) in case of RBF kernel are listed in Table 4-14, Table 4-15 and Table 4-16.

Table 4-14 Performance obtained with PLP cepstral coefficients with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	**F-score**	**Accuracy (%)**
2	0.65	59.2
3	0.67	61.1
4	0.69	62.9
5	**0.67**	**63.2**
6	0.67	62.2
7	0.66	61.4

Table 4-15 Performance obtained with PLP cepstral coefficients with MLP kernel for different values of MLP parameters

MLP Kernel		
$[p_1,p_2]$ values	**F-score**	**Accuracy (%)**
[0.001,-0.001]	0.49	42.9
[0.002,-0.002]	0.50	44.3
[0.003,-0.003]	0.53	48.7
[0.004,-0.004]	**0.58**	**52.7**
[0.005,-0.005]	0.56	50.2

Table 4-16 Performance obtained with PLP cepstral coefficients with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	**F-score**	**Accuracy (%)**
1	0.67	62.5
2	0.69	63.9
3	0.68	64.0

4	**0.71**	**65.8**
5	0.67	62.5
6	0.66	60.9
7	0.65	59.4

4.15.5 Performance with MFCCs

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with the feature vector formed from MFCCs are evaluated. The appropriate feature vector length was found out by trial and error. The performance results, the average F-score of eleven classes and accuracy, with different feature vector length are tabulated in Table 4-17.

Table 4-17 Classification performance obtained with MFCCs of different length

Vector Size	F-score	Accuracy (%)
10	0.65	59.2
15	0.67	62.1
20	0.69	64.0
25	**0.73**	**68.7**
30	0.70	65.0
35	0.68	62.0
Kernel – RBF, RBF Sigma Value - 3		

From Table 4-17, it can be inferred that, the best feature vector length of MFCCs for the given classification task is 25. Hence, a feature vector of length 25 was used for evaluating the performance of the proposed SVM based target classifier. The different kernels used are polynomial, multilayer

perceptron (MLP) and radial basis function (RBF). The average value of F-scores of 11-classes and the overall accuracy, obtained with MFCCs for kernel parameters such as, different values of polynomial order (p) in case of polynomial kernel, different values of MLP parameters (p_1, p_2) in case of MLP kernel and different values of sigma (σ) in case of RBF kernel are listed in Table 4-18, Table 4-19 and Table 4-20.

Table 4-18 Performance obtained with MFCCs with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	**F-score**	**Accuracy (%)**
2	0.68	63.6
3	0.69	64.3
4	0.71	66.2
5	**0.69**	**63.6**
6	0.68	62.5
7	0.67	61.8

Table 4-19 Performance obtained with MFCCs with MLP kernel for different values of MLP parameters

MLP Kernel		
[p_1,p_2] values	**F-score**	**Accuracy (%)**
[0.001,-0.001]	0.61	55.6
[0.002,-0.002]	**0.62**	**56.7**
[0.003,-0.003]	0.59	52.7
[0.004,-0.004]	0.55	49.1
[0.005,-0.005]	0.53	47.2

Table 4-20 Performance obtained with MFCCs with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	F-score	Accuracy (%)
1	0.71	63.2
2	0.72	65.5
3	0.73	66.8
4	**0.74**	**67.8**
5	0.68	64.1
6	0.68	63.2
7	0.67	60.3

4.15.6 Performance with GTCCs

The performance of a non-linear multiclass SVM based classifier to classify eleven classes of acoustic targets with the feature vector formed from GTCCs are evaluated. The appropriate feature vector length was found out by trial and error. The performance results, the average F-score of eleven classes and accuracy, with different feature vector length are tabulated in Table 4-21.

From Table 4-21, it can be inferred that, the best feature vector length of GTCCs for the given classification task is 25. Hence, a feature vector of length 25 was used for evaluating the performance of the proposed SVM based target classifier. The different kernels used are polynomial, multilayer perceptron (MLP) and radial basis function (RBF). The average value of F-scores of 11-classes and the overall accuracy, obtained with GTCCs for

kernel parameters such as, different values of polynomial order (p) in case of polynomial kernel, different values of MLP parameters (p_1, p_2) in case of MLP kernel and different values of sigma (σ) in case of RBF kernel are listed in Table 4-22, Table 4-23 and Table 4-24

Table 4-21 Classification performance obtained with GTCCs of different length

Vector Size	F-score	Accuracy (%)
10	0.64	59.2
15	0.67	62.5
20	0.69	64.0
25	**0.73**	**67.8**
30	0.69	65.1
35	0.68	62.9
Kernel – RBF, RBF Sigma Value - 4		

Table 4-22 Performance obtained with GTCCs with polynomial kernel for different polynomial orders

Polynomial Kernel		
Polynomial order	**F-score**	**Accuracy (%)**
2	0.66	60.3
3	0.68	62.9
4	**0.69**	**64.7**
5	0.68	63.2
6	0.67	62.2
7	0.67	61.4

Table 4-23 Performance obtained with GTCCs with MLP kernel for different values of MLP parameters

MLP Kernel		
$[p_1,p_2]$ values	F-score	Accuracy (%)
[0.001,-0.001]	0.55	49.8
[0.002,-0.002]	0.58	52.0
[0.003,-0.003]	**0.62**	**55.6**
[0.004,-0.004]	0.59	53.1
[0.005,-0.005]	0.55	48.7

Table 4-24 Performance obtained with GTCCs with RBF kernel for different values of RBF sigma

RBF Kernel		
σ value	F-score	Accuracy (%)
1	0.68	63.2
2	0.70	65.4
3	0.71	66.8
4	**0.73**	**67.8**
5	0.68	64.1
6	0.69	63.2
7	0.66	60.3

4.15.7 Analysis of results of different Feature Extraction techniques

Different feature extraction techniques have been evaluated for the proposed classifier with different kernels and varying kernel parameter values. For all the feature extraction methods evaluated, an appropriate feature vector length was found by trial and error with different evaluations. It is inferred from results that the appropriate feature vector length for LPC

& LPCC are 40, PLP is 20, MFCC and GTCC is 25. A lower feature vector length indicates the ability of the feature extraction technique to contain the characteristic information of the data in fewer coefficients, and is therefore advantageous in classifier design as fewer coefficients will result in lower complexity.

The kernels employed for the classifier evaluation are polynomial, MLP and RBF. The experiments were done on trial and error basis. The range of parameter values are heuristically arrived at, in which the values that exhibited better performance over others are tabulated. From the results obtained, RBF kernel has shown to outperform polynomial kernel and MLP kernel for all feature extraction techniques evaluated.

A comparison of the performance results of different feature extraction techniques suggest that cepstral features exhibit better classification results compared to others. This substantiated the ability of cepstral based techniques to yield features that perform well with data which is non-linearly corrupted with noise.

The results and inferences obtained from feature extraction is used for the formation of the final feature vector of the classifier through feature selection technique. The best results, the appropriate feature vector length and the appropriate kernel parameter, obtained from all the described feature extraction techniques is used to form the feature vector for the feature selection stage.

4.15.8 Performance with Feature Selection

Feature selection techniques have been employed in the proposed classifier to select the most relevant features suitable for the classification problem at hand. The feature selection algorithms operate on a feature vector

composed of average value of energy and energy entropy, 10 frame values each of spectral centroid, spectral roll-off, spectral flux and the average value of spectral entropy followed by 40 coefficients of LPC, 40 coefficients of LPCC, 20 coefficients of PLP, 25 coefficients of MFCC and 25 coefficients of GTCC. The total length of the feature vector would be 183 and is depicted in Figure 4.45. The classifier is evaluated by filter based approaches such as CFS and ECS, and Wrapper based approaches such as SFS, SBS and GA.

The experimental results with filter based techniques such as CFS and ECS for varying size of feature subset is tabulated in Table 4-25 and Table 4-26. The feature ranking obtained with CFS and ECS methods are tabulated in Table 4.27.

Table 4-25 Performance results obtained with CFS based feature selection for varying feature subset size

Vector Size	F-score	Accuracy (%)
20	0.61	56.7
30	0.66	61.1
40	0.71	67.6
50	0.70	65.0
60	0.67	61.8
70	0.64	58.9

1	Average value of energy across all frames
2	Average value of energy entropy across all frames
3	Spectral centroid 1st frame value
:	:
:	:
12	Spectral centroid 10th frame value
13	Spectral roll-off 1st frame value
:	:
:	:
22	Spectral roll-off 10th frame value
23	Spectral flux 1st frame value
:	:
:	:
32	Spectral flux 10th frame value
33	Average value of spectral entropy across all frames
34	LPC 1st coefficient
:	:
:	:
73	LPC 40th coefficient
74	LPCC 1st coefficient
:	:
:	:
113	LPCC 40th coefficient
114	PLP 1st coefficient
:	:
:	:
133	PLP 20th coefficient
134	MFCC 1st coefficient
:	:
:	:
158	MFCC 25th coefficient
159	GTCC 1st coefficient
:	:
:	:
183	GTCC 25th coefficient

Fig.4.45 Depiction of feature vector which forms the input to feature selection

Table 4-26 Performance obtained with ECS based feature selection for varying feature subset size

Vector Size	F-score	Accuracy
20	0.63	57.0
30	0.66	60.7
40	0.71	66.9
50	0.69	64.3
60	0.65	60.0
70	0.63	58.1

Table 4-27 Feature rankings obtained with CFS and ECS methods

Rank	Features (CFS)	Features (ECS)
1	MFCC 6th coefficient	MFCC 5th coefficient
2	MFCC 3rdcoefficient	MFCC 11thcoefficient
3	GTCC 5th coefficient	GTCC 5thcoefficient
4	MFCC 11th coefficient	GTCC 4thcoefficient
5	MFCC 2nd coefficient	GTCC 15thcoefficient
6	Average of energy	MFCC 22ndcoefficient
7	LPC 2nd coefficient	MFCC 3rdcoefficient
8	GTCC 6th coefficient	PLPC 7th coefficient
9	PLPC 7thcoefficient	GTCC 3rdcoefficient
10	LPCC 17thcoefficient	Spectral centroid 3rdframe value
11	MFCC 16th coefficient	GTCC 10th coefficient
12	GTCC 15th coefficient	Spectral flux 9thframe value

13	PLPC 2nd coefficient	Spectral roll-off 4th frame value
14	GTCC 10th coefficient	PLPC 2nd coefficient
15	Spectral centroid 3rdframe value	LPCC 23rdcoefficient
16	PLPC 17thcoefficient	LPCC 27thcoefficient
17	MFCC 20th coefficient	LPCC2ndcoefficient
18	LPCC 27th coefficient	LPC 39th coefficient
19	LPC 39thcoefficient	Average of spectral entropy
20	LPC 8thcoefficient	LPC 8th coefficient
21	Spectral roll-off 4thframe value	MFCC 7thcoefficient
22	LPCC3rdcoefficient	MFCC 16thcoefficient
23	LPCC 38thcoefficient	MFCC 20thcoefficient
24	Spectral centroid 1st frame value	MFCC 15thcoefficient
25	Average of spectral entropy	LPC 2ndcoefficient
26	Spectral flux 9thframe value	Average of energy
27	Average of energy entropy	LPCC 38thcoefficient
28	LPC 34thcoefficient	GTCC 25th coefficient
29	Spectral flux 4thframe value	LPCC 17thcoefficient
30	LPC 18thcoefficient	Average of energy entropy
31	PLPC 12thcoefficient	GTCC 23rdcoefficient
32	GTCC 25th coefficient	PLPC 15thcoefficient
33	LPCC 37thcoefficient	MFCC 24thcoefficient
34	GTCC 4th coefficient	PLPC 12thcoefficient

35	LPC 35thcoefficient	Spectral centroid 7thframe value
36	Spectral flux 3rd frame value	GTCC 20th coefficient
37	LPC 5thcoefficient	Spectral flux 3rd frame value
38	MFCC 22ndcoefficient	Spectral flux 4thframe value
39	PLPC 14th coefficient	LPC 18th coefficient
40	MFCC 23rdcoefficient	Spectral centroid 10thframe value
41	PLPC 15thcoefficient	Spectral roll-off 1stframe value
42	Spectral centroid 7thframe value	LPC 27th coefficient
43	Spectral roll-off 9thframe value	LPC 33th coefficient
44	LPCC 33rd coefficient	LPC 34thcoefficient
45	LPC 23rdcoefficient	LPC 23rdcoefficient
46	LPC 40thcoefficient	LPCC 7thcoefficient
47	MFCC 5th coefficient	GTCC 6th coefficient
48	LPC 3rdcoefficient	MFCC 18thcoefficient
49	MFCC 8th coefficient	LPCC 30thcoefficient
50	LPCC 7thcoefficient	MFCC 5thcoefficient
51	PLPC 5thcoefficient	PLPC 5thcoefficient
52	LPCC 11thcoefficient	GTCC 12th coefficient
53	LPCC 4thcoefficient	LPCC 39thcoefficient
54	LPC 31stcoefficient	Spectral roll-off 9thframe value
55	LPC 13thcoefficient	LPC 14th coefficient

56	Spectral flux 6th frame value	LPC 13th coefficient
57	GTCC 12th coefficient	LPC 40th coefficient
58	Spectral centroid 10thframe value	LPCC 12thcoefficient
59	Spectral roll-off 1st frame value	LPCC 37thcoefficient
60	LPC 27thcoefficient	LPCC 21stcoefficient
61	LPCC 35thcoefficient	Spectral centroid 5thframe value
62	LPCC 23rd coefficient	Spectral flux 6thframe value
63	LPCC 39thcoefficient	LPCC 36thcoefficient
64	LPCC 20thcoefficient	LPCC 20thcoefficient
65	MFCC 13th coefficient	PLPC 10thcoefficient
66	LPC 14thcoefficient	GTCC 18th coefficient
67	GTCC 8th coefficient	LPC 5th coefficient
68	MFCC 18th coefficient	LPC 6th coefficient
69	LPC 36thcoefficient	LPCC 11thcoefficient
70	LPCC 21stcoefficient	LPCC 35thcoefficient

The experimental results with wrapper based techniques such as SFS, SBS and GA based feature selection for the varying size of feature subset is tabulated in Table 4-298, Table 4-29 and Table 4-30.

Vector Size	F-score	Accuracy
20	0.63	56.3

30	0.66	60.3
40	0.70	65.4
50	**0.75**	**71.2**
60	0.73	68.3
70	0.68	64.3

Table 4-28 Performance results of SFS based feature selection.

Vector Size	F-score	Accuracy
20	0.63	56.3
30	0.66	60.3
40	0.70	65.4
50	**0.75**	**71.2**
60	0.73	68.3
70	0.68	64.3

Table 4-29 Performance results of SBS based feature selection

Vector Size	F-score	Accuracy
20	0.65	58.5
30	0.66	60.3
40	0.73	68.3
50	**0.70**	**72.3**
60	0.69	70.9
70	0.63	66.9

Table 4-30 Performance results of GA based feature selection

Vector Size	F-score	Accuracy
20	0.55	57.4
30	0.65	60.0
40	0.71	66.1

50	**0.74**	**70.2**
60	0.71	67.2
70	0.70	65.0

From Table 4-28, Table 4-29 and Table 4-30, it can be inferred that the best feature vector size for the proposed classifier is 50. Features selected by SBS method outperforms the features selected by SFS and GA, in terms of performance. However, the run time required for executing sequential feature selection techniques has been found to be approximately 7 times than that of GA, at the compromise of slightly lower performance. For selecting a feature vector of size 50, the runtime required by sequential algorithms 174 minutes as compared to 27 minutes in case of GA, when simulated in MATLAB 2013b platform. Since, time is an important constraint in underwater target classification applications, the faster approach, GA, is selected compromising slightly in performance. The features selected, for feature vector size 50, with SFS, SBS and GA are listed in Table 4-31.

Table 4-31 Features selected by SFS, SBS and GA with feature vector size 50

Method	Selected features
SFS	average value of energy, average value of energy entropy, spectral roll-off 4th frame value, spectral roll-off 8thframe value, average value of spectral entropy, LPC 3rd coefficient, LPC 8th coefficient, LPC 15th coefficient, LPC 18th coefficient, LPC 31st coefficient, LPC 33rdcoefficient, LPC 36th coefficient, LPCC 2nd coefficient, LPCC 9th coefficient, LPCC 17th coefficient, LPCC 20th coefficient, LPCC 26th coefficient, LPCC 31st coefficient, LPCC 36th coefficient, PLP 2nd coefficient, PLP 5th coefficient, PLP 7th coefficient, PLP13th coefficient, PLP 14th coefficient, PLP 17th coefficient, MFCC 2nd coefficient, MFCC 3rd coefficient, MFCC 5th coefficient, MFCC 6th coefficient, MFCC 11th coefficient, MFCC 12th coefficient, MFCC 15th coefficient, MFCC 20th coefficient,

	MFCC 22nd coefficient, MFCC 23rd coefficient, MFCC 24th coefficient, MFCC 25th coefficient, GTCC 3rdcoefficient, GTCC 5th coefficient, GTCC 6th coefficient, GTCC 7th coefficient, GTCC 15th coefficient, GTCC 16th coefficient, GTCC 18th coefficient, GTCC 19th coefficient, GTCC 20th coefficient, GTCC 22ndcoefficient, GTCC 23rd coefficient, GTCC 25th coefficient
SBS	average value of energy, average value of energy entropy, spectral roll-off 4th frame value, spectral flux 3rd frame value, spectral flux 8th frame value, average value of spectral entropy, LPC 3rd coefficient, LPC 8th coefficient, LPC 15th coefficient, LPC 18th coefficient, LPC 31st coefficient, LPC 33rdcoefficient, LPC 36th coefficient, LPCC 2nd coefficient, LPCC 9th coefficient, LPCC 17th coefficient, LPCC 20th coefficient, LPCC 26th coefficient, LPCC 31st coefficient, LPCC 33rd coefficient, LPCC 36th coefficient, PLP 2nd coefficient, PLP 5th coefficient, PLP 7th coefficient, PLP 11thcoefficient, PLP 14th coefficient, PLP 17th coefficient, MFCC 2nd coefficient, MFCC 3rd coefficient, MFCC 5th coefficient, MFCC 6th coefficient, MFCC 11th coefficient, MFCC 12th coefficient, MFCC 15th coefficient, MFCC 20th coefficient, MFCC 22nd coefficient, MFCC 23rd coefficient, MFCC 24th coefficient, MFCC 25th coefficient, GTCC 3rd coefficient, GTCC 4th coefficient, GTCC 6th coefficient, GTCC 7th coefficient, GTCC 15th coefficient, GTCC 16th coefficient, GTCC 18th coefficient, GTCC 19th coefficient, GTCC 20th coefficient, GTCC 22nd coefficient, GTCC 23rd coefficient
GA	average value of energy, average value of energy entropy, spectral roll-off 4th frame value, spectral flux 3rd frame value, spectral flux 8th frame value, spectral flux 9th frame value, average value of spectral entropy, LPC 3rd coefficient, LPC 8th coefficient, LPC 15th coefficient, LPC 18th coefficient, LPC 31st coefficient, LPC 34th coefficient, LPC 35th coefficient, LPCC 2nd coefficient, LPCC 3rd coefficient, LPCC 17th coefficient, LPCC 20th coefficient, LPCC 26th coefficient, LPCC 31st coefficient, LPCC 33rd coefficient, LPCC 36th coefficient, PLP 2nd coefficient, PLP 5th coefficient, PLP 7th coefficient, PLP13th coefficient, PLP 14th coefficient, PLP 17th coefficient, MFCC 2nd coefficient, MFCC 3rd coefficient, MFCC 5th coefficient, MFCC 6th coefficient, MFCC 11th

	coefficient, MFCC 15th coefficient, MFCC 20th coefficient, MFCC 22nd coefficient, MFCC 23rd coefficient, MFCC 24th coefficient, MFCC 25th coefficient, GTCC 3rd coefficient, GTCC 5th coefficient, GTCC 6th coefficient, GTCC 7th coefficient, GTCC 15th coefficient, GTCC 16th coefficient, GTCC 18th coefficient, GTCC 19th coefficient, GTCC 20th coefficient, GTCC 23rd coefficient, GTCC 25th coefficient

4.16 *Summary*

Features are signature patterns that best represent the signal and are given as classification cues to the classifier. The feature vector given to the classifier is of profound importance in determining the complexity and performance of the classifier. The chapter describes the various acoustic feature extraction techniques as well as feature selection techniques. A properly chosen feature vector enables the classifier to build better generalizable models. The choice of the specific feature vector for a particular classification task is derived from conclusions and inferences drawn from extensive experimentation.

The chapter also gives a brief introduction to machine learning and also the various concepts associated with it such as overfitting, underfitting and the bias-variance trade-off which led to the formulation of SVM. The chapter describes SVM's and the formulation of multiclass SVM's.

CHAPTER 5

CLASSIFIER OPTIMISATION

The parameters of classification algorithms have a profound influence on their performance in terms of generalization ability and robustness to noise. For an SVM-based classifier, the kernel function plays a vital role in dealing with non-linear and arbitrarily structured data, whose parameters also have an impact on the classification performance.

Any attempt to improve the performance of an SVM based classifier must invariably include parameter optimisation. Also, underwater target classification with dynamically changing constraints, necessitates dynamic selection of the optimal algorithmic parameters, kernel function and kernel parameters, which can be achieved through various optimisation techniques. The idea is to find out the parameters that maximize the performance of the proposed classifier. This chapter throws light upon the different optimisation strategies that can be adopted, associated concepts, and a brief about their merits and demerits. The chapter mainly focuses on meta-heuristic optimization algorithms, and also the different meta-heuristic optimisation algorithms adopted in this work.

5.1 *Background - The Optimisation problem*

Solutions to problems may not always have a binary nature, but are often rated in terms of quality concerning a performance metric. Learning algorithms, usually depend on parameters which control the size of the search

space or the way the search is conducted in the search space. Optimisation is performed to select the one which is of the best quality from a set of many candidate solutions that differ in quality, under the given circumstances. Optimisation reflects from the analytical quality of all intelligent beings to pursue the best. The mathematical formulation of the concept of optimisation can be as follows. Consider a set of candidate solutions, X, subjected to the optimisation problem. Typically, X is n-dimensional over certain domain, referred to as the search space. The optimisation problem is defined by an objective function (also referred to as cost function or fitness function), to estimate the performance of the candidate in X on the given problem.

$$f: X \to \mathbb{R} \tag{5.1}$$

The objective function, typically expressed as a function of design variable, defines a criteria to compare possible solutions. The optimisation problem can be formulated with an objective to find the best candidate solution that minimises or maximizes the fitness function f, as follows

$$\text{find} x \in X, \text{ so that} \forall y \in X : f(x) \leq f(y) \tag{5.2}$$

In practical applications, solutions to this problem may be a subset of all possible combinations or permutations of the elements of the vector X. Such problems are characterised by a finite set of solutions and are referred to as combinatorial optimisation problems.

5.2 Classification of Optimisation Problems

Optimisation problem can be classified in several ways as follows

- Constrained and Unconstrained: Constrained optimisation problems are subjected to one or more constraints and hence maintain the

search effort within a feasible region, whereas unconstrained optimisation problems have no constraints at all.

- Single objective and multi-objective: Optimisation problems may be based on a single objective function or multiple objective functions which involves decision making with multiple criteria.
- Nature of objective function and constraints: Optimisation problems can be classified as linear, quadratic, polynomial and non-linear depending upon the nature of the objective functions and the constraints. Determination of the type of solution often depends on the nature of involved functions.
- Deterministic and Stochastic: In deterministic optimisation problem, all the design variables are deterministic. In stochastic optimisation problem some or all the parameters are stochastic (non deterministic or probabilistic).

5.3 *Curse of Dimensionality*

The only way to ensure global optimality in solutions of an optimization problem is to evaluate all the candidate solutions, which is often computationally intractable. The adversity exponentially grows worse with increasing dimensionality, which is termed as the curse of dimensionality. Hence, it is essential to adopt optimisation methods that give acceptable performance in fewer dimensions and avoid exponentially increasing number of fitness evaluations with increasing dimensions. The optimisation methods should preferably have a linear relationship between the dimensionality of the problem and the number of candidate solutions i.e. it should have linear time complexity $O(n)$ in the dimensionality n of the problem to be optimised.

5.4 *NP Theory*

NP theory encompasses concepts that explain the solvability of an algorithm. An optimisation algorithm can be considered as efficient if it has polynomial run time [162]. The problems solved in polynomial time, by a deterministic machine is referred to as P problem while by non-deterministic machines is referred as NP (non-deterministic polynomial). The solutions to these problems are also verifiable in polynomial time.

NP-hardness (non-deterministic polynomial-time hardness), in computational complexity theory, is the defining property of a class of problems that are, informally, "at least as hard as the hardest problems in NP". More precisely, a problem H is NP-hard when every problem L in NP can be reduced in polynomial time to H; that is, assuming a solution for H takes 1 unit time, we can use H's solution to solve L in polynomial time. As a consequence, finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.

5.5 *No Free Lunch Theorem*

The No Free Lunch (NFL) theorem proposed by Wolpert and Macready [163] states that 'if an algorithm performs well in a certain class of problems, then it necessarily pays for that with degraded performance on the set of all remaining problems'. Consider $f: S \rightarrow$ Y which is the pair of all possible mappings within the range of values Y_s in the solution space S.

Then T_m,is defined as

$$T_m = \{(s_1, y_1), \ldots \ldots \ldots \ldots (s_m, y_m)\} \text{ where} y_j = f(s_j) \qquad 5.3$$

NFL theorem states that for any pair of algorithms a and b,

$$\sum_{f \in F} P\left(T_m^y | f, m, a\right) = \sum_{f \in F} P\left(T_m^y | f, m, b\right) \qquad 5.4$$

where $P\left(T_m^y | f, m, a\right)$ is the probability of obtaining a certain sequence of values y_1, y_2, y_3 .., y_m given that the search is for the function f with algorithm a.

The NFL theorem suggests that any two algorithms are equivalent when their performance is averaged across all problems, and hence, any one optimisation method will be as likely as any other to find a satisfactory solution. Thus for any optimisation algorithm, high performance over one class of problems is offset by performance over another class and a universally best method does not exist as illustrated in Figure 5.1. There may exist an optimisation method specialised for a particular problem. There are also algorithms which deliver good results for many different problem classes, but may be outperformed by a highly specialised method in each of them.

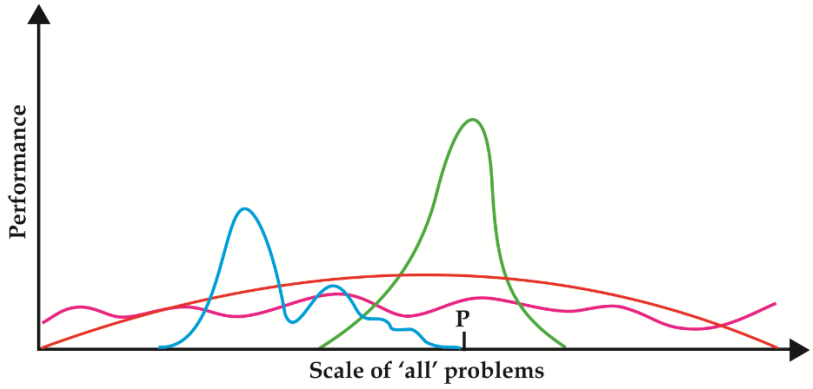

Fig.5.1 Visualization of No Free Lunch Theorem

Another manifestation of NFL theorem is related to the objective of an optimisation problem. The objective function of an optimisation problem is typically formulated based on the expected physical outcome as well as economic considerations. Following the NFL theorem, an algorithm which

performs well for a particular objective, say speed, may perform poorly for another metric say, accuracy. Thus, there always exists a fundamental trade-off between performance vs cost or performance vs reliability aspects of an optimisation algorithm and no choice is universally better than any other [115].

5.6 *Optimisation Search Strategies*

A large number of optimisation algorithms have been developed based on different search strategies. The naïve strategy towards finding the optimal parameters is to conduct an exhaustive search in the parameter space, which guarantees to find an optimal solution at a problem-specific computational cost. Since many practical optimisation problems are NP-hard, no polynomial-time algorithm is known, and the computational cost for exhaustive search is prohibitively high since it would correspond to running the algorithm for every possible value of parameters and hence the ideal linear time complexity $O(n)$ described in section 5.4 cannot be attained.

The traditional way of parameter optimisation is to conduct a grid search, a brute force method which exhaustively searches through the subset of parameters of the learning algorithm, guided by some performance metric. Grid search involves dividing the parameter space into grids of uniform size. The model is trained for the parameter values at each grid locations and the response is evaluated through a suitable performance metric, to obtain the optimal parameter set.

Another method of parameter optimisation is to employ greedy search techniques, which makes a greedy choice at each step for a solution that yields maximum performance, with the hope of finding a global optimum, without assessing its consequences. These techniques have high chances of

getting trapped in local optima. Greedy search techniques work well for monotonous objective function with smooth solution spaces [164]. However, for multimodal objectives, it is likely to get stuck in local optima. This approach is also called hill-climbing approach, referring to a mountaineer who chooses every next step in a manner that yields maximal improvement. Gradient search algorithms are examples of greedy search techniques.

Stochastic, heuristic and meta-heuristic optimisation techniques are resorted to, in applications where exhaustive search techniques seem to be impractical. Stochastic optimisation techniques are applied to solve highly non-linear, high-dimensional data difficult to be solved by classical deterministic methods. Stochastic optimisation techniques generate and use random variables for the formulation of the optimisation problem which involve random objective functions or random constraints. These techniques are faster than exhaustive search techniques but cannot guarantee best solutions or global solutions. Heuristic optimisation techniques are experience based techniques and are employed to solve complex logistics problems of higher dimension [165]. Heuristic derives from the Greek verb *heuriskein* means 'to find' by trial and error, and heuristic algorithms start searching the solution space with an initial guess and try to improve the quality of solutions over the course of iterations. Heuristic algorithms are capable of finding satisfactory solutions at a faster rate, but do not always guarantee to find optimal solutions.

Meta-heuristic algorithms are strategies that guide the search process. 'Meta' means 'beyond' or 'higher level', and the meta-heuristic algorithm guides a subordinate heuristic search through a combination of intelligent randomisation and local search, or in other words, exploration and exploitation. "A meta-heuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently

different concepts for exploring and exploiting the search space; learning strategies are used to structure information to efficiently find near-optimal solutions [166].” Meta-heuristics is an algorithmic architecture that customises the algorithmic procedures of a more problem-specific subordinate heuristics to obtain high-quality solutions. The subordinate heuristics may be a high-level or low-level procedure ranging from a simple local search to a complex search procedure.

Exploration (randomisation/diversification) and exploitation (intensification) are the two competing driving forces of a meta-heuristic algorithm, which allows it to effectively and efficiently explore the search space. Exploration and exploitation of the search space, are two competing design goals, in which the algorithm must be ‘clever’ to intensively exploit areas of the search space with high-quality solutions, as well as move to unexplored search spaces when required. Exploration increases the diversity of solutions and prevents them from being trapped at local optima, while exploitation in promising areas of the search space, based on accumulated search experience will increase the chances of the algorithm to find optimal solutions. A good combination of exploration and exploitation usually ensure global optimality [166].

Nature has evolved over millions of years and has found optimal solutions to a variety of natural phenomena. Nature-inspired meta-heuristic algorithms mimic different strategies of nature in finding optimal solutions. They can be grouped in four main categories: evolution- based, physics-based, swarm-based and human-based algorithms as depicted in Fig. 5.2 [167].

Evolution-based algorithms, inspired by the laws of natural evolution, arrive at an optimal solution by optimising a randomly generated

population of possible solutions, over the course of iterations. The most popular evolution-inspired technique is Genetic Algorithms (GA) that simulates the evolutionary laws of nature. Physics-based meta-heuristic algorithms are formulated by imitating the physical laws in the universe. The most popular algorithms in this category are Simulated Annealing (SA) and Gravitational Local Search (GLS).

Swarm- based techniques are formulated by abstracting the social behaviour of groups of animals or birds. The most popular algorithm is Particle Swarm Optimisation, which inspired by the social behaviour of bird flocking or fish schooling. Human based algorithms are formulated by modelling the human behaviour. An example of human based algorithm is the brain storm (BS) optimisation algorithm which is inspired by the human brainstorming process.

5.7 *Parameter Optimisation*

Machine learning algorithms have behavioural parameters which have profound effects on their performance. These parameters values are heavily problem dependent in the sense that, a set of values might work best on a certain problem, well on certain other problem instances and bad on another class of instances. Hence, setting the right algorithmic parameters is an influential design goal in machine learning.

Attempts to improve the performance of a classifier should invariably consider estimating suitable parameters for the problem at hand. However, finding the best parameters is a consequential task and it is very ambitious to understand the effect of each parameter. Some parameters may even have effects on other parameters which make the problem all the more complicated. Therefore, in order to determine parameters that are adaptable to the classification problem, parameter optimisation algorithms are resorted.

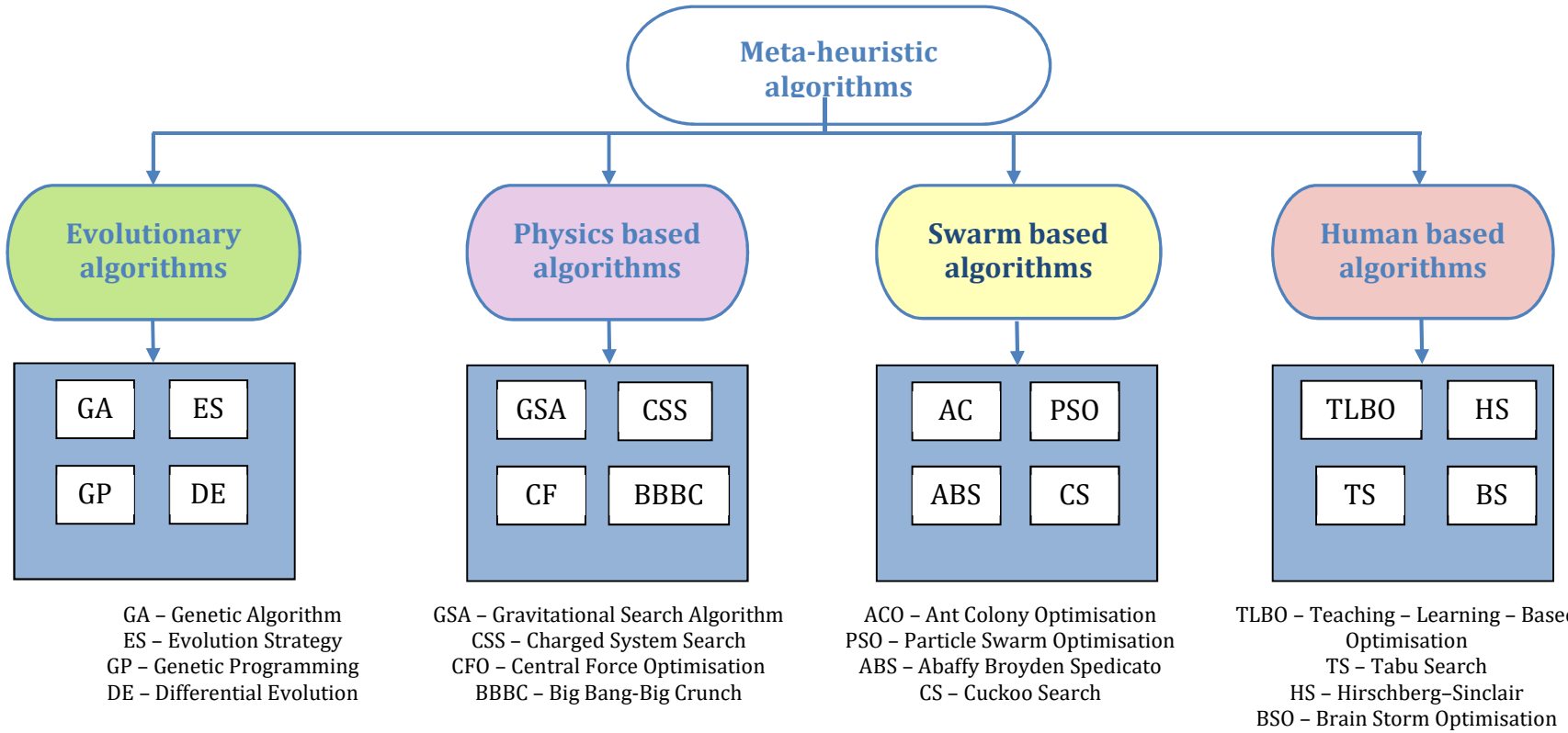

Fig.5.2 Classification of meta-heuristic algorithms

The search space for most parameter optimisation tasks is large, and hence exhaustive search techniques are normally not feasible in terms of runtime. Furthermore, the interdependency between the parameters and their effect on the algorithm's performance is largely unknown most of the times. Meta-heuristic algorithms, which strikes a balance between exhaustive search as well as heuristics are adopted in this work for optimising the parameter values of the underlying SVM based target classifier. Five nature inspired meta-heuristics algorithms namely, genetic algorithm, bat algorithm, whale optimisation algorithm, stochastic fractal search and symbiotic organisms search algorithm have been adopted for improving the classifier performance by tuning the algorithmic parameters. We have also proposed a modified symbiotic organisms search algorithm which is found to have better performance in classifying the underwater targets of interest.

5.8 *Genetic Algorithm (GA)*

Genetic algorithm (GA), is a meta-heuristic search and optimisation algorithm formulated by the abstraction of the 'survival of fittest' among individuals over consecutive generations in natural evolution [168]. GA starts with a population of candidate solutions (represented by chromosomes) for the problem at hand, characterising the natural ecosystem and makes it evolve by iteratively applying a set of stochastic operators like selection, crossover and mutation.

The chromosomes contain genes which encode a particular trait of the individual which may take different settings called alleles. An individual is also referred to as genome and the set of genes in a genome are referred to as genotype, which is the base for an organisms phenotype or the traits and characteristics it exhibits. They use metaphors which follow the theory of

natural evolution such as selection, recombination and mutation to evolve the solution to a problem.

The population consists of a number of co-existing organisms that compete for the same resources. The distinguishing traits of organisms that are most fit and capable of gathering resources will be carried over to the next generation. The entire population of the ecosystem is said to evolve over time, to accommodate organisms that are fitter than the previous generation. Thus, the characteristics that promote survival are preserved across generations.

At the beginning of the run of a genetic algorithm a population of chromosomes is created randomly, representing different solutions to the problem at hand. A fitness measure (objective function) is defined for evaluating the chromosomes and assigns a fitness score to each chromosome to assess its ability to solve the problem at hand. The algorithm then selects two members from the current population as parents to create offspring in the next generation. There are different methods for the selection process, however the chances of being selected as parents for the next generation is proportional to their fitness scores. Offspring for the next generation are created from parents by exchanging the genes in a process called cross-over. The algorithm then mutates the bits in a chromosome depending on a predefined mutation rate. After cross-over and mutation, the off-springs are evaluated just like their parents to measure their fitness of survival in the population. Fig.5.3 depicts the flowchart of GA.

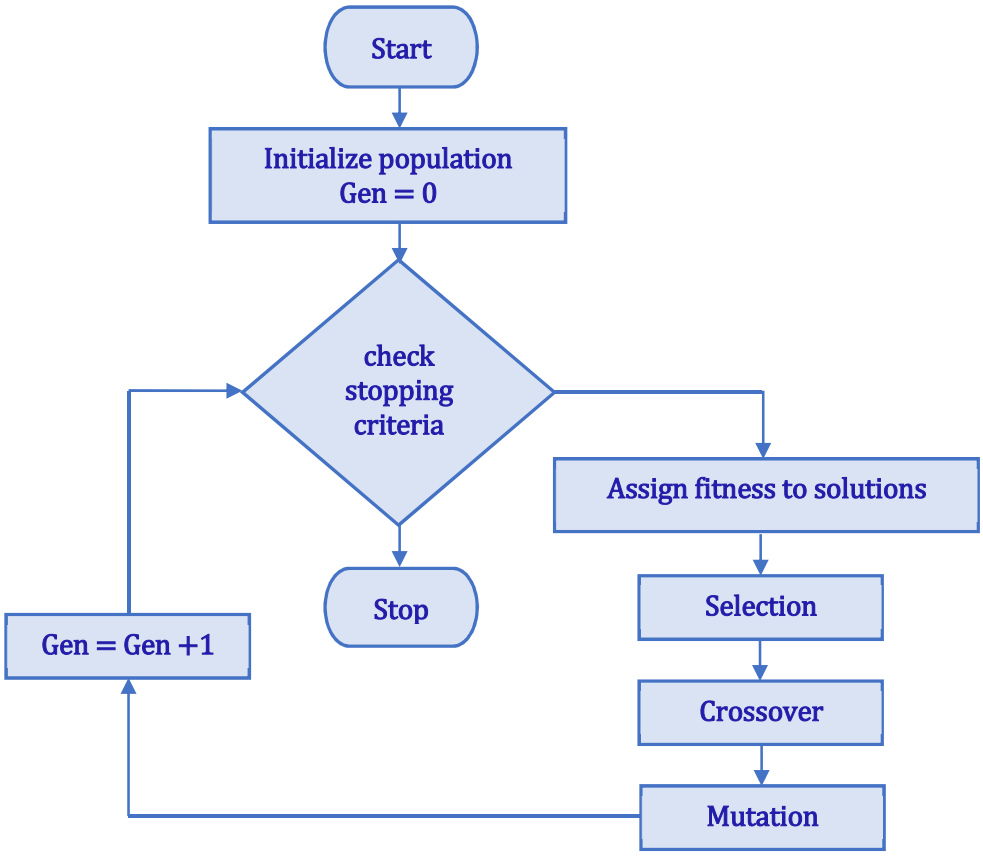

Fig.5.3 Flowchart of Genetic algorithm

5.8.1 GA operators

5.8.1.1 Selection

Selection is the process of choosing successful solutions from the current population as parents, which mate and recombine to create off-springs for the next generation. Selection is very crucial for the convergence rate of the GA, as good parents generate better and fitter solutions. Different techniques for selection in GA include,

- Tournament selection: Several tournaments are played among individuals chosen at random from the population. The winner of the tournament is selected for next generation to become a parent. With

sufficiently large tournament size, weak individuals have a lower probability to be selected, and thus selection pressure can be adjusted by altering the tournament size.

- Roulette wheel selection: In this method of selection, segments are created for each individual according to their fitness. A random number is generated and the individual whose segment spans the random number is selected. This technique is called so as it is homologous to a roulette wheel in which all chromosomes in the population has a slice proportional to its fitness.

- Rank selection: Rank selection is mostly employed when the individuals in the population have very close fitness values and selection by roulette wheel will have very low selection pressure towards fitter individuals resulting in a bad selection of parents. In rank selection, the population is ranked according to their fitness and the parents are selected depending on their ranking, with more preference given to higher ranked individuals.

- Steady-state selection: GA in steady state means that there are no generations, and instead of replacing children of selected parents in the next generation, two best individuals out of the two parents alongwith their children are added back into the population so that the population size remains constant.

5.8.1.2 Crossover

Crossover is the process of combining parents to produce offspring and is analogous to reproduction and biological crossover. Crossover operation has the primary responsibility of converging the search algorithm to an optimal solution. There are different techniques of crossover, some of which are detailed as in the following paragraphs.

- Single point crossover: A random crossover point is selected and the bits next to the cross-sites (tails of the parents) are swapped to create new off-springs as depicted in Fig.5.4.

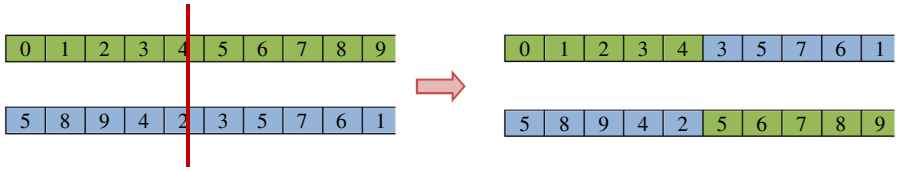

Fig.5.4 Depiction of single point crossover

- N - point crossover: Also called multi-point crossover, N cross over points are selected at random and the off springs are created by combining the parents at the crossover point as depicted in Fig.5.5.

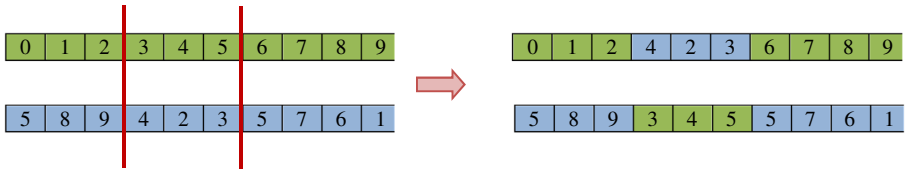

Fig.5.5 Depiction of N - point crossover

- Uniform crossover: In uniform crossover a crossover mask with binary values is created, and off springs are created by copying genes from the parents according to the values in this mask. At positions where there is a '1' in the mask, genes are carried from one parent, and at positions where there is a '0' in the mask, genes are carried from the other parent.

5.8.1.3 Mutation

Mutation operator, analogous to the biological mutation alters the value of one or more genes, to maintain genetic diversity across generations. Mutation allows exploration of the search space and is essential to the convergence of GA. Commonly used mutation schemes are random, swap, scramble, inversion , uniform as well as gaussian mutation.

- Random Mutation: Bits to be mutated are selected randomly and flipped.
- Swap Mutation: Values of individual genes are swapped between two positions on the chromosome.
- Scramble Mutation: Values of a selected subset of genes are randomly scrambled or shuffled.
- Inversion Mutation: Values of a selected subset of genes are inverted.
- Uniform Mutation: The chosen gene to be mutated is replaced by a random value between a user specified upper and lower bounds.
- Gaussian Mutation: A Gaussian distributed random values is added to the chosen gene, and if it falls outside user specified bounds, the gene value is clipped

The proposed SVM based underwater classifier was optimised with GA. The fitness measure employed is F-score. The results of the experiments for different population sizes are tabulated in Table 5-1. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

Table 5-1 Performance results with GA based optimisation for varying population size

Population Size	**10**	**15**	**20**	**25**
F-score (*class* 1)	0.60	0.68	**0.71**	0.71
F-score (*class* 2)	0.74	0.75	**0.76**	0.75
F-score (*class* 3)	0.73	0.74	**0.75**	0.74
F-score (*class* 4)	0.84	0.85	**0.86**	0.86

F-score (*class* 5)	0.79	0.79	**0.80**	0.79
F-score (*class* 6)	0.83	0.89	**0.92**	0.90
F-score (*class* 7)	0.75	0.75	**0.77**	0.75
F-score (*class* 8)	0.82	0.86	**0.87**	0.87
F-score (*class* 9)	0.79	0.79	**0.80**	0.79
F-score (*class* 10)	0.60	0.61	**0.66**	0.63
F-score (*class* 11)	0.59	0.65	**0.7**	0.69
F-score (Average)	0.73	0.76	**0.78**	0.77
Overall Accuracy (%)	69.1	71.3	**75.6**	73.8

5.9 *BAT Algorithm*

BAT algorithm was formulated by Xin-She Yang [169] based on the echolocation behaviour of bats. Bats use sonar to detect prey and avoid obstacles by emitting ultrasonic bursts and use the time delay information between the emission and detection of the echo, variation of loudness in the echo and the time difference between their ears to visualize their surroundings. Bats emit about 10 to 20 ultrasonic sound bursts per second typically of 5 to 20 ms duration. They tend to decrease loudness and increase the rate of emission to about 200 pulses per second when they approach a prey. They are able to detect the distance and orientation of the target, distinguish between different types of prey and can even estimate the moving speed of the prey such as small insects.

The characteristics of the echolocation pulse of bats vary with species and the following approximations are adopted to idealize the echo location characteristics of bats in order to formulate the BAT algorithm.

i. All bats use echolocation and they know the difference between food/prey and background barriers in some magical way.

ii. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min} in, varying wavelength λ and loudness A_0 to search for the prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r in the range [0, 1], depending on the proximity of their target.

iii. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

With the above approximations, the algorithm is formulated as follows. The algorithm commences by initializing the position and velocity of a population of bats. During the course of iterations, the position, velocity and frequency of the bats are updated as,

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad 5.5$$

$$v_i(t+1) = v_i(t) + (x_i(t) - G_{best})Q_i \quad 5.6$$

where $G_{best}$is the best solution obtained so far and Q_i indicates the frequency of the i^{th} bat which is updated in each course of iteration as follows,

$$Q_i = Q_{min} + (Q_{max} - Q_{min})\beta \quad 5.7$$

where, β is a random number drawn from a uniform distribution in [0,1]. To introduce exploitation of search space into the algorithm, a local random walk is performed around randomly picked solutions and is described in equation 5.8.

$$x_{new} = x_{old} + \varepsilon A^t \quad 5.8$$

where $\varepsilon \in$ [-1,1] is a random number, and $A_t =< A_i^t >$ is the average loudness of all the bats at the time step *t*. The loudness of the bats decreases and pulse emission rate increases when the bats approach the prey which is simulated in the algorithm to ensure convergence. During the course of iterations, when the solutions are improved, the loudness and emission rates are updated by the following equations to converge to the optimal solution.

$$A_i^{t+1} = \alpha A_i^t \tag{5.9}$$

$$r_i^{t+1} = r_i^0 \, [1 - \exp(-\gamma t)] \tag{5.10}$$

where $0 < \alpha < 1, \gamma > 0$ and as simulations proceeds and the algorithm converges , $A_i^t \to 0, r_i^t = r_i^0$ as $t \to \infty$. The pseudo-code of BAT algorithm is shown in Fig.5.6.

BAT algorithm is in a way improvement over the particle swarm optimisation algorithm. The update of velocities and positions of the bats are similar to updating the pace and range of the swarming particle in PSO. However, an intense local search controlled by loudness and pulse rate makes BAT algorithm superior to PSO due to its effectiveness in balancing exploration and exploitation.

The proposed SVM based underwater classifier was optimised with BAT algorithm. The results of the experiments for different population sizes are tabulated in Table 5-2. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

```
        Define pulse frequency
        Initialize pulse rate, loudness
        Initialize microbats (position, velocity and frequency)
Calculate the fitness and find the initial best solution
        while (max no. of iteration not reached)
        {
        Generate new solutions by eqns 5.5,5.6&5.7
        if  (rand > pulse rate)
                    Select a solution among the best solutions
                and generate a local solution around the best
                solution, eqn 5.8
        endif
        Evaluate the new solutions by their fitness values
                if (fitness has improved & solution is not too
                loud)
                        decrease loudness and increase pulse
                emission rate, eqns 5.9&5.10
                accept the new solutions
        endif
        find the current best solution
        }
```

Fig.5.6 BAT algorithm pseudo-code

Table 5-2 Performance results with BAT algorithm based optimisation for varying population size

Population Size	10	15	20	25
F-score (*class* 1)	0.60	0.65	0.63	0.63
F-score (*class* 2)	0.73	0.75	0.74	0.73
F-score (*class* 3)	0.72	0.77	0.75	0.72
F-score (*class* 4)	0.86	0.87	0.86	0.86
F-score (*class* 5)	0.79	0.85	0.81	0.81
F-score (*class* 6)	0.85	0.90	0.88	0.87
F-score (*class* 7)	0.75	0.78	0.77	0.75

F-score (*class* 8)	0.82	**0.84**	0.83	0.83
F-score (*class* 9)	0.79	**0.81**	0.80	0.79
F-score (*class* 10)	0.62	**0.67**	0.65	0.65
F-score (*class* 11)	0.61	**0.66**	0.65	0.65
F-score (Average)	0.74	**0.78**	0.76	0.75
Overall Accuracy (%)	70.2	**74.5**	72.4	71.6

5.10 *Whale Optimisation Algorithm*

Whale Optimisation Algorithm proposed by S. Mirjalili and A. Lewis, is a nature inspired meta-heuristic algorithm, formulated by the abstraction of hunting behavior of humpback whales [167].

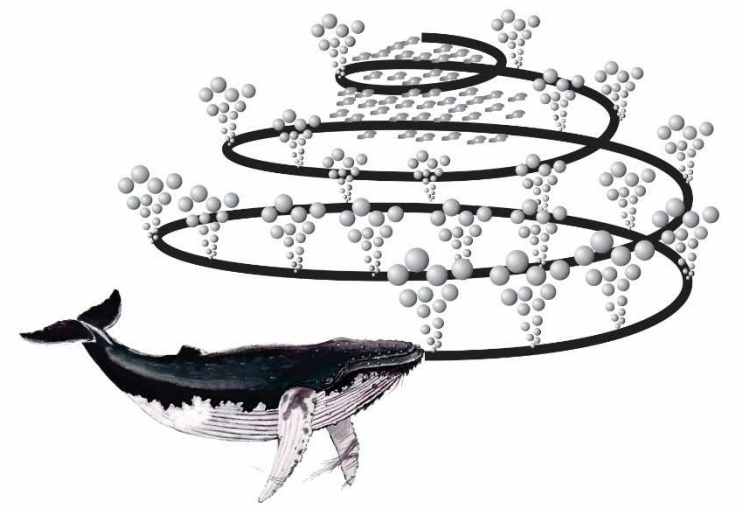

Fig.5.7 Bubble-net feeding behaviour of humpback whales

Humpback whales have a unique co-operative foraging behaviour which is popularly known as bubble-net feeding method shown in Fig.5.7, in which they use co-operative strategies to disorient and corral the fish into a bubble net that they create. They prefer to hunt school of small fish such as salmon, krill or herring. The whales gather together beneath the surface and

exhales out of their blowholes, to blow bubbles along a circle or 9 shaped path while continuing to encircle their prey and corralling the fish into the bubble net. The whales then simultaneously swim upwards with mouths wide open to feed on the trapped fish [170].

The WOA algorithm is modeled in three phases by simulating *i.e.* encircling the prey, the bubble or net attacking (exploitation phase), and the search for prey (exploration phase), behaviour of the humpback whales [167].

5.10.1 Encircling prey

Humpback whale recognizes the location of their prey and encircles them. The WOA starts with an initial population of solutions. The current best candidate solution is assume as the target prey. After the best search agent is defined, the other search agents will hence try to update their positions towards the best search agent which is mathematically formulated in equations 5.11 and 5.12.

$$\vec{D} = \left|\overrightarrow{C X^*} - \overrightarrow{X(t)}\right| \quad 5.11$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}\vec{D} \quad 5.12$$

where t indicates the current iteration, and $\vec{A}$ and $\vec{C}$ coefficient vectors, $\vec{X}$ is the position vector and X^* is the position vector of the best solution obtained so far. The vectors $\vec{A}$and $\vec{C}$ are given by

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad 5.13$$

$$\vec{C} = 2 \cdot \vec{r} \quad 5.14$$

where $\overrightarrow{a}$is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and $\overrightarrow{r}$is a random vector in [0,1].

5.10.2 Bubble-net attacking method

Humpback whales simultaneously swim around their prey within a shrinking circle and also along a spiral-shaped path, which is modelled with a 50% probability for updating the positions of the whales, either between the shrinking encircling mechanism or the spiral model as in equation 5.15.

$$\vec{X}(t+1) = \begin{cases} -\vec{A} \cdot \vec{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \quad 5.15$$

To model the shrinking encircling mechanism the values of $\vec{A}$ is chosen to be a random value in the interval [-a,a], where a is decreased from 2 to 0 over the course of iterations. For modeling the spiral position update, a spiral equation is created between the position of whale and prey to mimic the helix shaped movement. $\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right|$ indicates the distance of the i^{th} whale to the best solution obtained so far, which indicates the position of the prey, b is a constant for defining the shape of the spiral and l is a random number in [-1,1].

5.10.3 Search for prey

Humpback whales search randomly for prey. Hence, in order to model this random behaviour and allow exploration of the search space, the position of a randomly chosen search agent is updated instead of the best agent found so far. The mathematical model is as follows

$$\vec{D} = \left|\overrightarrow{C X_{rand}} - \overrightarrow{X(t)}\right| \quad 5.16$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}\vec{D} \quad 5.17$$

To ensure exploitation, $\vec{A}$ has random values either greater than 1 or less than -1.

The pseudo-code of WOA is shown in Figure 5.8. The WOA algorithm starts with a population of random solutions. At each iteration, the search agents update their position with respect to the position of the randomly chosen search agent in the searching for prey phase, and with respect to the position of the best agent in the encircling phase.

```
    Initialize the whale population, Xi, i = 1,2,…….,n
    Select the best search agent X*
    while (t<maximum no. of iterations)
    {
  update a, A, C, l and p for each search agent
  if (p<0.5)
    if(|A|<1)
    Update the position of the current search agent
    as in encircling phase
    else if(|A|≥1)
    Select a random search agent (Xrand)
    Update the position of the current search agent
    as in exploration phase
    end if
        else if(p≥0.5)
    Update the position of the current search agent as
    in exploitation (bubble net) phase
        endif
        Calculate fitness of each search agent
        Update X* if there is a better solution
        t=t+1;
    }
```

Fig.5.8 WOA pseudo-code

The algorithm also chooses between shrinking encircling mechanism and the spiral model according to a randomly drawn probability measure. The algorithm effectively employs both exploration and exploitation. Adaptive variation of the search vector $\vec{A}$ allows the WOA algorithm to smoothly

transit between exploration and exploitation: by decreasing A, some iterations are devoted to exploration ($|A| \geq 1$) and the rest is dedicated to exploitation ($|A| \leq 1$). The algorithm is terminated when the termination criterion is reached.

The proposed SVM based underwater classifier was optimised with WOA. The results of the experiments for different population sizes are tabulated in Table 5-3. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

Table 5-3 Performance results with WOA based optimisation for varying population size

Population Size	**10**	**15**	**20**	**25**
F-score (*class* 1)	0.62	0.63	**0.70**	0.67
F-score (*class* 2)	0.73	0.73	**0.75**	0.75
F-score (*class* 3)	0.72	0.72	**0.73**	0.73
F-score (*class* 4)	0.80	0.81	**0.82**	0.82
F-score (*class* 5)	0.83	0.85	**0.86**	0.86
F-score (*class* 6)	0.81	0.81	**0.82**	0.82
F-score (*class* 7)	0.73	0.74	**0.75**	0.75
F-score (*class* 8)	0.79	0.83	**0.83**	0.83
F-score (*class* 9)	0.76	0.80	**0.81**	0.81
F-score (*class* 10)	0.63	0.63	**0.65**	0.65
F-score (*class* 11)	0.61	0.62	**0.67**	0.66
F-score (Average)	0.73	0.74	**0.76**	0.76
Overall Accuracy (%)	68.7	70.2	**73.1**	72.0

5.11 *Stochastic Fractal Search (SFS)*

Stochastic Fractal Search (SFS) algorithm is a meta-heuristic algorithm, formulated by Salimi [171] based on the random fractals observed in nature. Fractal refers to objects or quantities that displays self similarity, in somewhat technical sense, in all dimensions [172]. Fractal shapes can be generated by common techniques such as iterated function systems, strange attractors, L-systems, escape time fractals, finite subdivision rules and random fractals. Random fractals can be created by physically motivated models such as diffusion limited aggregation (DLA) model. SFS algorithm is inspired from random fractals grown by DLA model, which can model clusters describing a bacterial colony. In DLA model, virtual particles moving through space following a random walk diffuse and stick together around a seed particle. A cluster is built up over time as more and more particles collide and clump together. While forming the cluster, the probability of the particle which sticks to the farthest end of the cluster is high in comparison to the one that penetrates the interior.

The problem solutions to an optimization problem are considered as individual particles in a population. SFS relies on three simple principles to find a solution

i. Each particle has an electrical potential energy which is updated based on its fitness value

ii. Each particle diffuses, and cause some other random particles to be created, and the energy of the seed particle is divided among generated particles

iii. The best generated particles from the diffusion process are considered, and the rest of the particles are eliminated

Two main processes to perform the SFS, are the diffusion process and the update process. In the diffusion process, the particles diffuse around its current position to form a cluster and in the updating process, the velocity and position of the particles is updated based on its position and the position of other particles.

The algorithm begins by randomly placing the particles at different locations in the search space. Each particle is initialized with equal energy obtained from equation 5.18.

$$E_i = \frac{E}{P} \tag{5.18}$$

where E is the maximum electrical potential energy considered to solve the problem.

After initialization, the fitness function of all particles is calculated and the best point BP is tracked. Each particle is then diffused in each generation, which creates new particles. The diffusion process is modelled through Gaussian walks as in equations 5.19 and 5.20.

$$GW_1 = Gaussian\,(\mu_{BP}, \sigma) = (\varepsilon \cdot BP - \varepsilon' \cdot P_i) \tag{5.19}$$

$$GW_2 = Gaussian\,(\mu_P, \sigma) \tag{5.20}$$

where ε and $\varepsilon'$are uniformly distributed random numbers in the range [0,1]. P_i is the i^{th} point, and BP is the best point in the cluster. μ_{BP}, μ_P and σ are Gaussian parameters where μ_{BP} is exactly equal to BP and $\mu_P$is equal to P. For a particular generation g, the standard deviation is computed as in equation 5.21.

$$\sigma = \left| \frac{log(g)}{g} \cdot (P_i - BP) \right| \tag{5.21}$$

The term $\frac{\log(g)}{g}$ causes the size of Gaussian jumps to decrease as generations proceed, thereby forcing the algorithm to move closer to the solution. The diffusion process introduces the exploitation property of the algorithm and increases the chances of the algorithm in finding the optimal solutions.

After the diffusion process, all points (the total number being N) are ranked based on their individual fitness values, and each of them are assigned the probability of entering the next generation, the value of which follows a uniform distribution as in equation 5.22.

$$Pa_i = \frac{rank(P_i)}{N} \qquad 5.22$$

As per the above equation 5.22, the individual with a higher rank will have a higher probability to be selected into the next generation.

For each point P_i in the cluster, with $Pa_i < \varepsilon$, where ε is a random value in the range [0,1], the j^{th} component of P_i is updated using the equation 5.23,

$$P_i'(j) = P_r(j) - \varepsilon \cdot \left(P_t(j) - P_i(j)\right) \qquad 5.23$$

where P_i' is the updated position of P_i, and P_r and P_t are randomly selected points in the group. All the points are again sorted based on their ranks calculated by equation 5.22. Again the position of points with $Pa_i < \varepsilon$, where ε is a random value in the range [0,1], is updated according to equations 5.24 and 5.25.

$$P_i'' = P_i' - \hat{\varepsilon} \cdot (P_t' - BP) \; for \; \varepsilon' \leq 0.5 \qquad 5.24$$

$$P_i'' = P_i' + \hat{\varepsilon} \cdot (P_t' - BP) \; for \; \varepsilon' > 0.5 \qquad 5.25$$

where, P_r and P_t are randomly selected points and $\hat{\varepsilon}$ are random numbers generated by the Guassian normal distribution. The new point P_i'' is replaced by P_i' if its fitness function value is better than P_i'. The energy is distributed among the new particles according to their fitness value. The distribution energy equation for the selected particles is given by

$$E_i^j = \left[\left(\frac{f_j}{f_i + \sum_{k=1}^{q} f_k}\right)\right] \times E_i \qquad 5.26$$

Where f_j is the energy of the diffused particle and f_i is its fitness value before diffusion and E_i is calculated as per equation 5.26. The updating process contributes to the exploration property of the SFS algorithm.

The proposed SVM based underwater classifier was optimised with SOS algorithm. The results of the experiments for different population sizes are tabulated in Table 5-4. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

Table 5-4 Performance results with SFS algorithm based optimisation for varying population size

Population Size	10	15	20	25
F-score (*class* 1)	0.64	0.67	**0.69**	0.68
F-score (*class* 2)	0.67	0.69	**0.70**	0.69
F-score (*class* 3)	0.76	0.76	**0.76**	0.76
F-score (*class* 4)	0.90	0.91	**0.93**	0.91
F-score (*class* 5)	0.80	0.81	**0.81**	0.80
F-score (*class* 6)	0.81	0.82	**0.82**	0.82
F-score (*class* 7)	0.75	0.77	**0.77**	0.76
F-score (*class* 8)	0.79	0.80	**0.80**	0.79

F-score (*class* 9)	0.77	0.77	**0.77**	0.77
F-score (*class* 10)	0.68	0.70	**0.70**	0.70
F-score (*class* 11)	0.63	0.67	**0.68**	0.66
F-score (Average)	0.75	0.76	**0.77**	0.76
Overall Accuracy (%)	70.5	72.4	**74.5**	73.1

5.12 *Symbiotic Organisms Search*

Symbiosis is derived from two greek words 'sym', which means 'together' and 'bios' which means 'life' and can be stated as close and often long term interaction and reliance between two variant biological species. Symbiotic relationships exist in organisms as a strategy for them to adapt with changes in environment, which may help them increase their fitness and survival advantage in the eco-system over a long term. Symbiotic relationships can be obligate or facultative. Obligate symbiosis is when two organisms are in a symbiotic relationship because they can't survive without each other. Facultative symbiosis is when the species live together by choice. Pollination symbiosis is an obligate symbiosis whereas cleaning symbiosis is a facultative symbiosis. There are three main types of symbiotic relationships: mutualism, commensalism and parasitism depending on the nature and character of interaction between the associated organisms. Cheng and Prayogo formulated Symbiotic organisms Search (SOS) algorithm based on symbiotic interaction strategies observed in nature [173]. Three phases are introduced in the SOS algorithm to resemble the biological interaction occurring in the three types of symbiosis.

SOS algorithm commences with a randomly generated initial population representing the initial ecosystem, in which each organism

corresponds to a candidate solution with an associated fitness value that reflects its survival cost in the ecosystem. In the initial ecosystem, a group of organisms are randomly generated within the search space. Each organism interacts with the other organisms randomly through all phases. This process is repeated for all the organisms in the ecosystems, until termination criteria are met. The pseudo-code of SOS algorithm is shown in Figure 5.9.

```
Initialize Ecosystem: define the no. of organisms
Define termination criteria
Calculate the fitness and find the initial best solution
while (termination criteria not met)
{
for i = 1:ecosize
Mutualism Phase
       Select one organism randomly, Xj where Xj≠Xi
       Determine Mutual Vector  = (Xi + Xj)/2
       Determine Benefit Factors BF1 & BF2 = 1 or 2
       Modify Xi and Xj according to equations 5.27 & 5.28
Calculate fitness values of modified organisms

if(modified organisms are fitter than previous)
       accept modified organism to replace the previous
else
       reject modified organisms and keep the previous

Commensalism Phase
       Select one organism randomly, Xj where Xj≠Xi
       Modify organism Xi according to equation 5.30

Parasitism Phase
       Select one organism randomly, Xj where Xj≠Xi
       Create a Parasite Vector from Organism Xi
       Calculate fitness values of new organisms

if(Parasite Vector fitter than Xj)
       replace organism Xj with Parasite Vector
else
       Keep organism Xj and delete Parasite Vector
end
}
```

Fig.5.9 SOS pseudo-code

5.12.1 Mutualism Phase

Mutualism is a symbiotic relationship in which each individual benefit from the activity of the other, like the relationship between bees and flowers

in which the bee benefits from the nectar it gets from the flower and the flower benefits from the pollination carried out by the bees.

Mutualistic relation can be mathematically modelled as follows. An organism X_i engages in a mutualistic relationship with another organism X_j with the goal of increasing their mutual survival advantage in the ecosystem. Based on the mutualistic symbiosis between organisms X_i and X_j, the new candidate solutions are calculated as follows

$$X_{inew} = X_i + rand(0,1)(X_{best} - Mutual\ Vector \times \mathrm{BF}_1) \quad 5.27$$

$$X_{jnew} = X_j + rand(0,1)(X_{best} - Mutual\ Vector \times \mathrm{BF}_2) \quad 5.28$$

$$Mutual\ Vector = (X_i + X_j)/2 \quad 5.29$$

BF_1 and BF_2 are factors which determine the degree of benefit to each organism.

This phase aids in exploration of new regions, as organisms located far away in the search space are brought to interact by *Mutual Vector*. Further, the two interacting individuals are updated concurrently rather than singly

5.12.2 Commensalism Phase

Commensalism is a symbiotic relationship in which one organism (the commensal) benefits, and the other is apparently unaffected (or receives minimal benefit) like that of algae and barnacles growing on turtles and whales. The new candidate solutions in this phase are found by exploiting promising regions around the best solution. This phase controls the convergence of the algorithm and is mathematically modelled as follows

$$X_{inew} = X_i + rand(-1,1)(X_{best} - X_j) \quad 5.30$$

5.12.3 Parasitism Phase

Parasitism is a symbiotic relationship in which one organism (the parasite) benefits, at the expense of other such that the other organism is adversely affected, like fleas harming the hosts on which they live. To model this phase a Parasite Vector is created in the search space by duplicating organism X_i, and modifying it along a randomly selected dimension. Both the organisms are then evaluated, and the fitter organism is allowed to resume its position in the ecosystem. Parasitism phase allows exploration of the search space and may arrive at unique solutions that may be located in completely different regions of the search space.

The proposed SVM based underwater classifier was optimised with SOS algorithm. The results of the experiments for different population sizes are tabulated in Table 5-5. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

Table 5-5 Performance results with SOS algorithm based optimisation for varying population size

Population Size	**10**	**15**	**20**	**25**
F-score (*class* 1)	0.68	**0.75**	0.71	0.69
F-score (*class* 2)	0.76	**0.78**	0.78	0.76
F-score (*class* 3)	0.74	**0.76**	0.75	0.74
F-score (*class* 4)	0.88	**0.89**	0.89	0.88
F-score (*class* 5)	0.90	**0.92**	0.91	0.90
F-score (*class* 6)	0.78	**0.79**	0.78	0.78
F-score (*class* 7)	0.88	**0.89**	0.88	0.88
F-score (*class* 8)	0.85	**0.86**	0.85	0.85

F-score (*class* 9)	0.85	**0.87**	0.87	0.85
F-score (*class* 10)	0.71	**0.73**	0.73	0.71
F-score (*class* 11)	0.69	**0.73**	0.70	0.69
F-score (Average)	0.79	**0.82**	0.80	0.79
Overall Accuracy (%)	77.5	**80.4**	78.9	76.7

5.13 *Improving the Parameter Optimisation by Modified - Symbiotic Organisms Search*

The right balance between a meta-heuritc algorithm's ability to explore and exploit determines its efficiency to arrive at the optimal solution. Exploitation ability is introduced in meta–heuristic algorithms usually by directing the search towards the best solution. In the proposed modified symbiotic organisms search (*m*-SOS) algorithm, a balanced exploitation is introduced through a cognitive component as well as a social component. The cognitive component is dependent on the best value of a particular organism (X_{best}) and the social component is dependent on the best value among all the organisms (G_{best}). A better coverage of the search space is achieved through the social component and the cognitive component directs the search towards the possibly best solution in the neighbourhood. The social component reduces the algorithms chance to get trapped in a local minima. In the proposed *m*–SOS algorithm another modification proposed is to introduce a weighted mutual vector. By introducing so, the factor *Mutual Vector* × BF in the SOS algorithm controls the degree by which the candidate solution approaches the best solution. The weights are determined by the fitness values obtained in the previous iterations. Hence, it may be considered as the momentum component which determines the impetus with which the candidate solution moves towards the best solution, based on the

previous experience in terms of the fitness value. The right balance between exploration and exploitation determines the efficacy of the algorithm. In *m*–SOS algorithm exploration is also introduced through the GA operators crossover and mutation. Randomly generated indices of the solutions X_{inew} and X_{jnew} are subjected to crossover operators to generate new solutions X'_{inew} and X'_{jnew}. The resulting solutions X'_{inew} and X'_{jnew} if found to have better fitness values are accepted, else X_{inew} and X_{jnew} are retained. In the parasitism phase, the organism is randomly mutated to generate the parasite vector. The algorithm can be described in detail as follows

5.13.1 Mutualism Phase

The new candidate solutions for X_i and X_j are calculated based on the mutualistic symbiosis between organism X_i and X_j.

$$X_{inew} = X_i + rand(0,1)(X_{best} - Mutual\,Vector \times \mathrm{BF}_1)(G_{best} - X_i) \quad 5.31$$

$$X_{jnew} = X_j + rand(0,1)(X_{best} - Mutual\,Vector \times \mathrm{BF}_2)(G_{best} - X_j) \quad 5.32$$

X_{best} is the best value of that particular organism, and G_{best} is the best value among all organisms.

BF_1 and BF_2 are factors which determine the degree of benefit to each organism

$$Mutual\,Vector = (w_1 X_i + w_2 X_j)/2 \quad 5.33$$

where $w_1 = \eta_{i-1}/100$ and $w_2 = \eta_{j-1}/100$

η_{i-1} is the percentage of correctly classified samples in the previous iteration

$$Mutual\ Vector = \left(w_1 X_i + w_2 X_j\right)/2 \tag{5.34}$$

where $w_1 = \eta_{i-1}/100$ and $w_2 = \eta_{j-1}/100$

$$idx = [i_1, i_2, \dots\dots\dots\dots, i_M], M < N \tag{5.35}$$

$i_1, i_2, \dots\dots\dots\dots, i_M$ are random numbers between 1&N

$$X_{CO} \in X_i \qquad \forall x\{x \in X_{CO} \rightarrow x \in X_i\} \tag{5.36}$$

$$X_{CO}(i) = X_i\left(idx(i)\right), i = 1, \dots\dots\dots\dots, M$$

$$Y_{CO} \in X_j \qquad \forall y\{y \in Y_{CO} \rightarrow y \in Y_i\}$$

$$Y_{CO}(i) = X_j\left(idx(i)\right), i = 1, \dots\dots\dots\dots, M$$

$$X' = X_i(idx(i)) = Y_{CO}(i)$$

$$Y' = X_j(idx(i)) = X_{CO}(i)$$

The new solutions can be calculated as

$$X'_{inew} = X' + rand(0,1)(X_{best} - Mutual\ Vector \times \mathrm{BF}_1)(G_{best} - X_i) \tag{5.37}$$

$$X'_{jnew} = Y' + rand(0,1)(X_{best} - Mutual\ Vector \times \mathrm{BF}_2)\left(G_{best} - X_j\right) \tag{5.38}$$

accept the best solutions

$$\text{if } (fitness\ (X'_{inew}) > fitness(X_{inew})) \tag{5.39}$$

accept X^i_{inew} else retain X_{inew}

$$\text{if } (fitness\ (X'_{jnew}) > fitness\left(X_{jnew}\right)) \tag{5.40}$$

accept X^i_{jnew} else retain X_{jnew}

5.13.2 Commensalism Phase

The new candidate solutions in this phase are calculated as

$$X_{inew} = X_i + rand(-1,1)\left(X_{best} - X_j\right)\left(G_{best} - X_j\right) \quad 5.41$$

The proposed SVM based underwater classifier was optimised with *m*-SOS algorithm. The results of the experiments for different population sizes are tabulated in Table 5.6. Each run of the algorithm consists of 500 iterations, except when the algorithm is terminated when the average relative change in fitness value is stalled over 25 generations.

Table 5.6 Performance results with the proposed *m*-SOS algorithm based optimisation for varying population size

Population Size	**10**	**15**	**20**	**25**
F-score (*class* 1)	0.77	0.81	**0.80**	0.78
F-score (*class* 2)	0.81	0.84	**0.84**	0.82
F-score (*class* 3)	0.74	0.79	**0.79**	0.76
F-score (*class* 4)	0.90	0.91	**0.91**	0.90
F-score (*class* 5)	0.92	0.93	**0.93**	0.93
F-score (*class* 6)	0.71	0.76	**0.76**	0.75
F-score (*class* 7)	0.93	0.95	**0.95**	0.94
F-score (*class* 8)	0.85	0.87	**0.86**	0.86
F-score (*class* 9)	0.84	0.88	**0.88**	0.85
F-score (*class* 10)	0.78	0.82	**0.82**	0.80
F-score (*class* 11)	0.79	0.84	**0.82**	0.81
F-score (Average)	0.82	0.85	**0.85**	0.84
Overall Accuracy (%)	82.5	85.1	**88.4**	83.3

5.14 *Summary*

This chapter throws light into optimisation of a classifier and the various concepts associated with it. Different optimisation strategies that can be adopted are briefed. Meta-heuristic optimisation algorithms strike a balance between exploration and exploitation of the search space. Meta-heuristic optimisation is adopted in this work for parameter optimisation. The various meta-heuristic optimisation algorithms adopted are described.

The results of adoption of optimisation algorithms to the proposed classifier, has improved its performance. Thus, it can be concluded that, tuning the algorithmic parameters to the optimum is essential for improving the classifier performance.

CHAPTER 6

CONCLUSIONS

This thesis addresses one of the emerging topics in Sonar Signal Processing, viz. improving the performance of the target classifier for noise sources in the ocean. The underlying classifier implemented is a multi-class SVM based classifier. The main challenges faced by underwater target classification systems are due to diverse noise sources that vary with time and location. The performance of underwater classifiers can be improved by selecting the most representative feature vector that characterizes the signals and also by setting the optimal parameters of the underlying classifier. Different procedures for feature extraction has been studied and implemented for generating the feature vector. Procedures for dynamic feature selection according to changing underwater environment, has also been implemented resulting in an improvement in the classifier performance. The optimal choice of the classifier parameters, kernel function and kernel function parameters has been found by meta-heuristic algorithms. Different meta-heuristic algorithms have been implemented for parameter tuning of the underlying classifier, which has shown to improve the classifier performance. A modified-SOS (m-SOS) algorithm is also proposed which has shown to give higher performance compared to other algorithms implemented. This chapter brings out the salient highlights of the work alongwith enlisting the scope and direction for future research in this area.

6.1 *Highlights of the Thesis*

Underwater target recognition has gained considerable significance due to its strategic as well as commercial importance. The composite and dynamic nature of the propagation medium imposed by the ocean makes underwater target recognition a very challenging task. The ocean, as a propagation medium consists of dynamically varying composite noise sources comprising of man-made noises such as noise due to shipping, natural noises due to environment such as wind, waves, currents and rains, and biological noises emanated by underwater living organisms, that establish a perpetual noise backdrop. Underwater target activity reflected by the acoustic activity of the targets of interest, are captured by hydrophones. However, the hydrophones receive an acoustic mixture of requisite signals embedded with the ocean noise. Individual targets of interest are identified from hydrophone captured acoustic mixture, through their characteristic signatures that are patterned by feature recognition algorithms, which are then provided to the classifier for classification into different classes. In this work, a support vector machine (SVM) based target classifier is used to distinguish between targets of 11 classes. The work reported in the thesis entitled *Underwater Target Classifier with Improved Success Rate using Meta-Optimal Support Vector Machines* addresses one of the emerging topics in Sonar Signal Processing, *viz.* improving the performance of an underwater target classifier for noise sources in the ocean which is achieved through carefully selected feature vector and also through optimising the classifier parameters. The following are the salient highlights of this thesis.

6.1.1 Need and Requirement of optimising the target classifier

The introductory chapter of the thesis throws light on the various noise sources in the ocean as well as the need and requirement of optimising

the target classifier. The chapter also highlights the applications of underwater target classification. The underlying principle of operation of the proposed classifier is also briefly introduced in this chapter.

6.1.2 Preparation of a State-of-the-art Literature

The development of a classifier involves extraction of target specific acoustic signatures using suitable feature extraction and feature selection algorithms and adoption of a suitable classifier for classification. The performance of the classifier is improved through optimisation of the classifier parameters. As prelude to the development of a classifier, a state-of-the-art literature survey has been prepared on various aspects such as the underwater acoustic environment, acoustic feature extraction techniques, and various classifiers such as the statistical classifiers, lazy learning algorithms, decision tree classifiers, neural network classifiers and support vector machines. Literature review has also been prepared on different optimization techniques that can be used for improving the classifier performance.

6.1.3 Feature Vector Based Classifier

The methodology suggested to be adopted for realizing the proposed classifier involves the formulation of the acoustic signature of the targets of interest using suitable feature extraction techniques. Various acoustic feature extraction schemes have been highlighted in the thesis. Cepstral based techniques have been found to give better classification performance. To reduce the dimensionality of the feature vector by removing redundant and irrelevant features, feature selection algorithms are used. A reduced but highly representative feature vector will lower the complexity of the classifier. Various feature selection algorithms have been attempted on the proposed classifier and their performance is analysed. The feature vectors of known targets are labelled according to their classes to create a knowledge

base for the classifier. The classifier uses the labelled signals in the knowledge base to compare against the feature vector of an unknown signal, based on which the system performs the decision-making process.

6.1.4 SVM Based Multi-class Classifier

Once the feature vector has been extracted and the knowledge base is created, a suitable classifier needs to be identified to do the final classification task. In this work an SVM based classifier, which can learn non-linear decision surfaces efficiently, is adopted. SVMs are relatively easy to implement and very robust due to its sound theoretical background. They also have the advantage of creating a model with minimized Vapnik–Chervonenkis (VC) dimension, resulting in a low expected probability of error and thus good generalization performance. SVM's were originally proposed as binary classifiers, but were later extended to solve multi-class problems by decomposing the multi-class problem into simple binary classification problems. Two popular approaches of solving multi-class problems are, one-against-one approach and one-against-all approach. In this work a multi-class SVM based target classifier for classifying 11 classes of acoustic targets is developed using one-against-all approach.

6.1.5 Parameter Optimisation of the classifier

The algorithmic parameters of the classifier impact its performance. Particularly, for an SVM based classifier, which is acclaimed for its high generalization capabilities, setting the right kernel parameters is a very determining factor in the classifier performance. Hence, attempting to set the right classifier parameters results in its performance improvement. Since, the underwater environment is highly dynamic in nature with changing channel properties, dynamic selection of algorithmic parameters, kernel and kernel parameters are required. Optimization algorithms are resorted to, for

scanning the parameter space to determine the most suitable set of parameters. In this work, parameter optimization is attempted using five meta-heuristic algorithms, namely GA, BAT, WOA, SFS and SOS. Though all the optimization algorithms have shown to improve the performance of a multi-class SVM based classifier, SOS algorithm exhibited superior performance improvement over others. A modification to SOS algorithm is also proposed which is called *m*-SOS (modified-SOS) algorithm. The *m*-SOS algorithm has shown to better improve the performance of the classifier in selecting the optimal parameter setting for the underlying classification task.

6.2 *Future Scope for Research*

The work presented in this thesis has a significant role to play in view of its practical applications. This work also has substantial scope for further research for improving the overall system performance. Some of the possible proposals for future work in this area are enlisted below.

6.2.1 Expansion of Knowledge Base

The proposed prototype system for identifying the noise sources in the ocean works on a simulated environment with a limited data set. By expanding the knowledge base, more training data can be obtained. A well trained classifier will certainly yield better performance in actual environment. Attempts were made to obtain more data from the Indian seas, but could not succeed to the expectations, and so could be taken up as a separate major project in collaboration with appropriate funding agencies.

6.2.2 Hardware Implementation

The proposed SVM based classifier works on a simulated environment and the modules have been developed in Matlab. The hardware

version of the system can be developed using high-end Digital Signal Processors, and FPGA systems.

6.2.3 Augmentation of Feature Vector

The performance of the classifier can be improved by augmenting the feature vector used, with higher order features such are bispectrum and trispectrum which are based on third and fourth order statistics respectively. However, as the order increases, the computational complexity and storage requirement also increases, which necessitates the requirement of efficient hardware systems for their implementation.

Most feature extraction techniques based on cepstral analysis are based on auditory models of human ear. Developing a feature extraction technique based on the auditory model of marine mammals may result in better signature features for underwater targets. Incorporation of features motivated by auditory models of marine mammals may also be worked up on.

6.2.4 Incorporating Meta-meta optimal SVMs

Meta-optimization refers to employing an optimization algorithm to optimize the parameters of another algorithm. In this work, we are employing different meta-heuristic algorithms to optimize the parameters of SVM. Meta-meta-optimization can be resorted to, for optimizing the parameters of the meta-heuristic algorithm which is optimizing the parameters of the SVM classifier. Parameter free algorithms such as Teaching Learning Based Optimisation (TLBO) can be attempted as the top level algorithm for meta-meta-optimization, for optimising the parameters of the meta-heuristic algorithm which is optimising the parameters of underlying SVM based classifier.

6.3 *Summary*

An attempt has been made in this chapter to bring out the salient highlights of the work carried out for the implementation of an underwater target classifier with improved success rate using meta-optimal SVMs. A discussion on the scope and directions for future research works in this area has also been presented.

BIBLIOGRAPHY

[1] R. J. Urick, *Principles of underwater sound*, New York: McGraw Hill Book Company, 1975.

[2] L. Mattos and E. Grant, "Passive sonar applications: target tracking and navigation of an autonomous robot," in *IEEE International Conference on Robotics and Automation, 2004, Proceedings* 2004, pp. 4265–4270 Vol.5.

[3] F. H. Fisher and V. P. Simmons, "Sound absorption in sea water," J. Acoust. Soc. Am., vol. 62, no. 3, pp. 558–564, Sep. 1977.

[4] A. D. Waite, *Sonar for Practising Engineers*, Third Edit. England: John Wiley & Sons, Inc., 2002.

[5] "Principles of Underwater Sound.", Division of War Research, Sonar Data Division, University of California California, Technical Report, Div 6, vol. 7.

[6] C. L. Piggott, "Ambient Sea Noise at Low Frequencies in Shallow Water of the Scotian Shelf," J. Acoust. Soc. Am., vol. 36, no. 11, pp. 2152–2163, Nov. 1964.

[7] X. Lurton, *An Introduction to Underwater Acoustics: Principles and Applications*, Springer Verlag, 2002.

[8] S. J. Russell and P. Norvig, *Artificial Intelligence. A Modern Approach.*, Third Edit. Prentice Hall, 2010.

[9] N. Kottege, F. Kroon, R. Jurdak, and D. Jones, "Classification of underwater broadband bio-acoustics using spectro-temporal features," *WUWNet*, pp. 19:1–19:8, 2012.

[10] C. H. Chen, J. D. Lee, and M. C. Lin, "Classification of Underwater Signals Using Neural Networks," *Tamkang J. Sci. Eng.*, vol. 3, no. 1, pp. 31–48, 2000.

[11] Zhang Xinhua, Lu Zhenbo, and Kang Chunyu, "Underwater acoustic targets classification using support vector machine," in *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, 2003, p. 932–935 Vol.2.

[12] T. Meir, M. Tsionskiy, A. Sutin, and H. Salloum, "Decision Learning Algorithm for Acoustic Vessel Classification," *Homel. Secur. Aff.*, vol. 3, pp. 1–7, 2012.

[13] M. Vieira, P. J. Fonseca, M. C. P. Amorim, and C. J. C. Teixeira, "Call

recognition and individual identification of fish vocalizations based on automatic speech recognition: An example with the Lusitanian toadfish," *J. Acoust. Soc. Am.*, vol. 138, no. 6, pp. 3941–3950, 2015.

[14] D. Diep, H. Nonon, I. Marc, J. Delhom, and F. Roure, "Acoustic counting and monitoring of shad fish populations," *Int. AmiBio Work. Recent Prog. Comput. Bioacoustics Assess. Biodivers.*, June, 2013.

[15] J. Noda, C. Travieso, and D. Sánchez-Rodríguez, "Automatic Taxonomic Classification of Fish Based on Their Acoustic Signals," *Appl. Sci.*, vol. 6, no. 12, pp. 443, 2016.

[16] P. H. Dahl, J. H. Miller, D. H. Cato, and R. K. Andrew, "Underwater Ambient Noise," *Acoust. Today*, vol. 3, no. 1, pp. 23, 2007.

[17] D. L. Bradley and R. Stern, "Underwater sound and the marine mammal acoustic environment: A guide to fundamental principles," *Mar. Mammal Comm.*, July 2008.

[18] J. A. Hildebrand, "Anthropogenic and natural sources of ambient noise in the ocean," *Mar. Ecol. Prog. Ser.*, vol. 395. pp. 5–20, 2009.

[19] D. Ross, "Ship sources of ambient noise," *IEEE J. Ocean. Eng.*, vol. 30, no. 2, pp. 257–261, 2005.

[20] M. A. McDonald, J. A. Hildebrand, and S. M. Wiggins, "Increases in deep ocean ambient noise in the Northeast Pacific west of San Nicolas Island, California," *J. Acoust. Soc. Am.*, vol. 120, no. 2, pp. 711–718, 2006.

[21] B. R. Kerman, D. L. Evans, D. R. Watts, and D. Halpern, "Wind dependence of underwater ambient noise," *Boundary-Layer Meteorol.*, vol. 26, no. 2, pp. 105–113, 1983.

[22] R. V Vijayabaskar V, "Analysis and Modeling of Wind Dependence of Ambient Noise in Shallow Water of Arabian Sea," *Eur. J. Sci. Res.*, vol. 50, no. 1, pp. 28–34, 2011.

[23] University of California, Division on War Research, "Underwater Noise Caused By Snapping Shrimp," 1967.

[24] F. A. Everest, "Acoustical Characteristics of Noise Produced by Snapping Shrimp," *J. Acoust. Soc. Am.*, vol. 19, no. 4, pp. 726, 1947.

[25] I. Junaid, S. Rubina, A. Faheem, and H. Muhammad, "Statistical Characterization and Sensitivity Analysis of Underwater Ambient Noise Model," in *International Bhurban Conference on Applied Sciences & Technology*, 2011, pp. 212–217.

[26] M. Bouvet and S. C. Schwartz, "Underwater noises: Statistical

modeling, detection, and normalization," *J. Acoust. Soc. Am.*, vol. 83, no. 3, pp. 1023–1033, 1988.

[27] Y. U. Song, W. Wang, and F. Guo, "Information in News Video," *Learning*, vol. 2, no. July, pp. 12–15, 2009.

[28] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimed.*, vol. 13, no. 2, pp. 303–319, 2011.

[29] S. L. J. Marple, "A tutorial overview of modern spectral estimation," *International Conference on Acoustics Speech and Signal Processing*, vol. 4. pp. 2152–2157, 1989.

[30] "Power spectra Estimation." National Semiconductor Application Note 255, 1980.

[31] M. Luzin, I. Dubinsky, "High Resolution Spectrum Estimating Algorithm," *Proc. IEEE Ocean. Conf.*, pp. 1409–1412, 1998.

[32] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition with timeFrequency audio features," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 6, pp. 1142–1158, 2009.

[33] H. Asefi, B. Ghoraani, A. Ye, and S. Krishnan, "Audio scene analysis using parametric signal features," *Can. Conf. Electr. Comput. Eng.*, pp. 922–925, 2011.

[34] W. Jinfang and W. Jinbao, "Speaker Recognition Using Features Derived from Fractional Fourier Transform," *Fourth IEEE Work. Autom. Identif. Adv. Technol.*, pp. 95–100.

[35] T. Li, X. Li, and Z. Xia, "Classification of underwater mines by means of the FRFT and SVM," *2010 IEEE Int. Conf. Inf. Autom. ICIA 2010*, pp. 1824–1829, 2010.

[36] H. Jleed and M. Bouchard, "Acoustic environment classification using discrete hartley transform features," *Can. Conf. Electr. Comput. Eng.*, pp. 0–3, 2017.

[37] H. Wang, A. Divakaran, A. Vetro, S. F. Chang, and H. Sun, "Survey of compressed-domain features used in audio-visual indexing and analysis," *J. Vis. Commun. Image Represent.*, vol. 14, no. 2, pp. 150–183, 2003.

[38] L. Jian, L. Zhong, and X. Ying, "Underwater target recognition based on WPT and SVM," *CCTAE 2010 Int. Conf. Comput. Commun. Technol. Agric. Eng.*, vol. 1, pp. 275–278, 2010.

[39] G. Shi, J. Hu, M. Han, and Y. Li, "Underwater acoustic target

recognition based on multi-timeslice demodulation line spectrum feature," *Proc. 2008 IEEE Int. Conf. Inf. Autom. ICIA 2008*, pp. 835–839, 2008.

[40] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The Cepstrum: A Guide to Processing," *Proc. IEEE*, vol. 65, no. 10, pp. 1428–1443, 1977.

[41] R. C. Kemerait and D. G. Childers, "Signal Detection and Extraction by Cepstrum Techniques," *IEEE Trans. Inf. Theory*, vol. 18, no. 6, pp. 745–759, 1972.

[42] C. J. van der Merve and J. A. d. Preez, "Calculation of LPC-Based Cepstrum Coefficients using Mel-Scale Frequency Warping," in *South African Symposium on Communications and Signal Processing*, 1991, pp. 17–21.

[43] C. C. Laguno, "Cepstral analysis synthesis on the mel frequency scale, and an adaptative algorithm for it." Graz (Austria), April 2008.

[44] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing Mel-frequency cepstral coefficients on the power spectrum," *2001 IEEE Int. Conf. Acoust. Speech, Signal Process. Proc.,* vol. 1, pp. 1764–1769, 2001.

[45] W. H. W. Han, C. F. Chan, C. S. Choy, and K. P. Pun, "An efficient MFCC extraction method in speech recognition," *2006 IEEE Int. Symp. Circuits Syst.*, pp. 145–148, 2006.

[46] M. Kucukbayrak, O. Gunes, and N. Arica, "Underwater Acoustic Signal Recognition Methods," *J. Nav. Sci. Eng.*, vol. 5, no. 3, pp. 64–78, 2009.

[47] A. B. Nielsen, S. Sigurdsson, L. K. Hansen, and J. Arenas-García, "On the relevance of spectral features for instrument classification," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2, pp. 485–488, 2007.

[48] U. Bhattacharjee, "A Comparative Study of LPCC and MFCC Features For The Recognition Of Assamese Phonemes," *Int. J. Eng. Res. Technol.*, vol. 2, no. 4, pp. 1–6, 2013.

[49] R. S. Zebulum and M. Vellascot, "A Comparison of Different Spectral Analysis Models for Speech Recognition using Neural Networks," in *39th Midwest symposium on Circuits and Systems, IEEE*, 1996, pp. 1428–1431.

[50] D. von Zeddelmann, F. Kurth, and M. M¨uller, "Perceptual Audio Features for Unsupervised Key-Phrase Detection," in *IEEE*

International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 257–260.

[51] G. Muhammad, Y. A. Alotaibi, M. Alsulaiman, and M. N. Huda, "Environment recognition using selected MPEG-7 audio features and mel-frequency cepstral coefficients," *5th Int. Conf. Digit. Telecommun. ICDT 2010*, pp. 11–16, 2010.

[52] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP speech analysis technique," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 1, no. April 1992, pp. 121–124, 1992.

[53] H. Florian, G. Stemmer, C. Hacker, F. Brugnara, I. Centro, and P. Trento, "Revising Perceptual Linear Prediction (PLP)," *Interspeech 2005*, pp. 2997–3000, 2005.

[54] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, 1990.

[55] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Trans. Multimed.*, vol. 14, no. 6, pp. 1684–1689, 2012.

[56] Jun Qi, Dong Wang, Yi Jiang, and Runsheng Liu, "Auditory features based on Gammatone filters for robust speech recognition," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, 2013, no. 1, pp. 305–308.

[57] A. I. Technology and E. B. Tazi, "Robust Features for Noisy Text-Independent Speaker Identification Using Gfcc Algorithm," vol. 36, no. 2, pp. 206–216, 2012.

[58] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan, "An auditory-based feature for robust speech recognition," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, no. 1, pp. 4625–4628, 2009.

[59] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications," *Proc. IEEE*, vol. 79, no. 3, pp. 278–305, 1991.

[60] A. M. Richardson and W. S. Hodgkiss, "Applications of Bispectral Analysis.pdf." University of California, San Diego and Scripps Institute of Oceanography, 1993.

[61] C. L. Nikias and M. R. Raghuveer, "Bispectrum estimation: A digital signal processing framework," *Proc. IEEE*, vol. 75, no. 7, pp. 869–891, 1987.

[62] K. P. Shroff and H. H. Maheta, "A Comparative Study of Various Feature Selection Techniques in High-Dimensional data set to

Improve Classification Accuracy," 2015.

[63] P. Pramokchon and P. Piamsa-Nga, "Effective threshold estimation for filter-based feature selection," *20th Int. Comput. Sci. Eng. Conf. Smart Ubiquitos Comput. Knowledge, ICSEC 2016*, 2017.

[64] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," *Proc. - 2016 Int. Conf. Eng. MIS, ICEMIS 2016*, 2016.

[65] R. Wald, T. M. Khoshgoftaar, and A. Napolitano, "Stability of filter- and wrapper-based feature subset selection," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, pp. 374–380, 2013.

[66] J. Huang, Y. Cai, and X. Xu, "A Wrapper for Feature Selection Based on Mutual Information," *18th Int. Conf. Pattern Recognit.*, vol. 2, pp. 18–21, 2006.

[67] V. Dimitrios and K. Constantine, "Sequential Forward Feature Selection with Low Computational Cost", *Signal Process. Conf. 2005 13th Eur.*, pp. 1–4, 2005.

[68] K. Zhu and J. Yang, "A Cluster-Based Sequential Feature Selection Algorithm," in *Ninth International Conference on Natural Computation (ICNC)*, 2013, pp. 848–852.

[69] H. Laanaya, a. Martin, a. Khenchaf, and D. Aboutajdine, "Feature Selection Using Genetic Algorithm for Sonar Images Classification with Support Vector," *ECPS 2005 Conf.*, March, pp. 15–18, 2005.

[70] P. Y. Xia, X. Q. Ding, and B. N. Jiang, "A GA-based feature selection and ensemble learning for high-dimensional datasets," *Proc. 2009 Int. Conf. Mach. Learn. Cybern.*, vol. 1, July, pp. 7–12, 2009.

[71] H. M. Zawbaa, E. Emary, A. E. Hassanien, and B. Parv, "A wrapper approach for feature selection based on swarm optimization algorithm inspired from the behavior of social-spiders," *Proc. 2015 7th Int. Conf. Soft Comput. Pattern Recognition, SoCPaR 2015*, pp. 25–30, 2016.

[72] B. Shapo and R. Bethel, "A novel passive broadband Bayesian detector/tracker," *Proc. IEEE Sens. Array Multichannel Signal Process. Work.*, pp. 92–96, 2000.

[73] T. BINESH, M. H. SUPRIYA, and P. R. SASEENDRAN PILLAI, "An efficient HMM Underwater Signal Classifier with Enhanced Fading Channel Performance," *J. Circuits, Syst. Comput.*, vol. 23, no. 9, pp. 1-19, 2014.

[74] T. Kim and K. Bae, "HMM-based underwater target classification with synthesized active sonar signals," *Eur. Signal Process. Conf.*,

Eusipco, pp. 1805–1808, 2011.

[75] P. Peso Parada and A. Cardenal-López, "Using Gaussian mixture models to detect and classify dolphin whistles and pulses," *J. Acoust. Soc. Am.*, vol. 135, no. 6, pp. 3371–3380, 2014.

[76] A. M. Patrikar and J. P. Baker, "Improving accuracy of Gaussian mixture model classifiers with additional discriminative training," *Proc. Int. Jt. Conf. Neural Networks,2016*, pp. 1673–1677, 2016.

[77] G. Cipli, F. Sattar, and P. F. Driessen, "A novel approach to low frequency activity detection in highly sampled hydrophone data based on B-spline approximation," *IEEE Pacific RIM Conf. Commun. Comput. Signal Process. - Proc.*, pp. 460–466, 2015.

[78] S. Reese, G. Sukthankar, and R. Sukthankar, "An Efficient Recognition Technique for Mine-like Objects using Nearest-Neighbor Classification," *Process. Underw. Def. Technol. Eur.*, no. 3, pp. 1–7, 2003.

[79] Y. Zhang, J. Yang, H. Hou, and J. Shi, "A new evidence classification algorithm for target recognition in underwater acoustic research," in *INTER-NOISE*, 2016, pp. 5132–5136.

[80] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees Classifiers—A Survey," *IEEE Trans. Syst. Man Cybern.*, vol. 35, no. 4, pp. 476–487, 2005.

[81] R. Rojas, *Neural Networks - A Systematic Introduction*, Springer International Publishing, 1996.

[82] B. Krose and P. van der Smagt, *Introduction To Neural Networks*. 1996.

[83] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Trans. Syst. Man Cybern.*, vol. 30, no. 4, pp. 451–462, 2000.

[84] O. A. Allim and H. F. Hashem, "Automatic recognition of sonar signals using neural network," in *15th National Radio Science Conference, Egypt*, 1998, pp. 1–8.

[85] J. Ribeiro-Fonseca and L. Correia, "Identification of underwater acoustic noise," *Proc. Ocean.*, 1994, vol. 2, pp. 597- 602, 1994.

[86] M. R. Azimi-Sadjadi, D. Yao, Q. Huang, and G. J. Dobeck, "Underwater target classification using wavelet packets and neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 784–94, 2000.

[87] A. Khotanzad, J. Lu, and M. Srinath, "Target detection using a neural

network based passive sonar system," *Neural Networks, 1989*, pp. 335–340, 1989.

[88] M. Farrokhrooz and M. Karimi, "Ship Noise Classification Using Probabilistic Neural Network and AR Model Coefficients," in *Oceans - Europe*, 2005, pp. 1107–1110.

[89] J. J. M. Madrid, J. R. C. Corredera, G. De, and M. Vela, "a Neural Network Approach To Doppler-Based Target Classification," pp. 450–453.

[90] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[91] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer New York, 2000.

[92] T.-M. Huang, V. Kecman, and I. Kopriva, *Kernel Based Algorithms for Mining Huge Data Sets*. 2006.

[93] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[94] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: a stepwise procedure for building and training a neural network," in *Neurocomputing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 41–50.

[95] V. N. Vapnik, *Statistical Learning Theory*, vol. 2. 1998.

[96] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[97] J. Weston and C. Watkins, "Support Vector Machines for Multi-Class Pattern Recognition," *Proc. 7th Eur. Symp. Artif. Neural Networks*, no. April, pp. 219–224, 1999.

[98] K. Crammer and Y. Singer, "On The Algorithmic Implementation of Multiclass Kernel-based Vector Machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2001.

[99] J. Manikandan and B. Venkataramani, "Design of a modified one-against-all SVM classifier," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, no. October, pp. 1869–1874, 2009.

[100] Z. Xinhua, L. Zhenbo, and K. Chunyu, "Underwater Acoustic Targets Classification Using Support Vector Machine," *Neural Networks Signal*, pp. 932–935, 2003.

[101] D. Li, M. R. Azimi-Sadjadi, and M. Robinson, "Comparison of different classification algorithms for underwater target discrimination," *IEEE Trans. Neural Networks*, vol. 15, no. 1, pp. 189–194, 2004.

[102] H. Robotham *et al.*, "Application of classifiers : Support vector machines , artificial neural networks and classification trees to identify acoustic schools," in 4th International Conference on Biomedical Egineering and infromation, pp. 2119–2124, 2011.

[103] T. A. Ferose Babu and R. Pradeepa, "Support vector machine applied to underwater target classification," *Proc. - 2014 4th Int. Conf. Adv. Comput. Commun. ICACC 2014*, pp. 46–49, 2014.

[104] Z. Lian, K. Xu, J. Wan, and G. Li, "Underwater Acoustic Target Classification Based on Modified GFCC Features," no. 1, pp. 258–262, 2017.

[105] H. Li, Y. Cheng, W. Dai, and Z. Li, "A method based on wavelet packets-fractal and SVM for underwater acoustic signals recognition," *Int. Conf. Signal Process. Proceedings, ICSP*, pp. 2169–2173, 2014.

[106] H. Yang, A. Gan, H. Chen, Y. Pan, J. Tang, and J. Li, "Underwater acoustic target recognition using SVM ensemble via weighted sample and feature slection," in *International Bhurban Conference on Applied Sciences & Technology*, 2016, pp. 522–527.

[107] X. Yang, Q. Song, and Y. U. E. Wang, "Data Classification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 21, no. 5, pp. 961–976, 2007.

[108] X. Yang, Q. Song, and A. Cao, "Weighted support vector machine for data classification," *Proceedings. 2005 IEEE Int. Jt. Conf. Neural Networks, 2005.*, vol. 2, pp. 859–864, 2005.

[109] J. A. . Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3. pp. 293–300, 1999.

[110] Y. Lee and S. Huang, "Reduced Support Vector Machines: A Statistical Theory," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 1–13, 2007.

[111] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, 2006.

[112] Y. Tang and H. H. Zhang, "Multiclass Proximal Support Vector Machines," *J. Comput. Graph. Stat.*, vol. 1, no. 2, pp. 339–355, 2006.

[113] D. Zhuang, B. Zhang, Q. Yang, J. Yan, Z. Chen, and Y. Chen, "Efficient text classification by weighted proximal SVM," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 538–545, 2005.

[114] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. 2007.

[115] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley International, 2003.

[116] J. Brownlee, *Clever Algorithms Nature Inspired Programming Recipes*, 2011.

[117] X. S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, First editio, 2014.

[118] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *J. Comput. Appl. Math.*, vol. 196, no. 2, pp. 425–435, 2006.

[119] A. V. Phan, M. Le Nguyen, and L. T. Bui, "Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems," *Appl. Intell.*, vol. 46, no. 2, pp. 455–469, 2017.

[120] J. Zhou, O. O. Maruatona, and W. Wang, "Parameter optimization for Support Vector Machine Classifier with IO-GA," *Proc. - 2011 1st Int. Work. Complex. Data Mining, IWCDM 2011*, pp. 117–120, 2011.

[121] N. E. Ayat, M. Cheriet, and C. Y. Suen, "Optimization of the SVM kernels using an empirical error minimization scheme," *Intl Workshop on Pattern Recognition with Support Vector Machines, 2002,* pp.354-369.

[122] M. Cho and T. T. Hoang, "Feature Selection and Parameters Optimization of SVM Using Particle Swarm Optimization for Fault Classification in Power Distribution Systems," *Computational Int and Neuroscience*, 2017.

[123] Y. Yan, Y. Zhang, and F. Gao, "Dynamic artificial bee colony algorithm for multi-parameters optimization of support vector machine-based soft-margin classifier," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 160, 2012.

[124] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, "Robust optimization of SVM hyperparameters in the classification of bioactive compounds," *J. Cheminform.*, vol. 7, no. 1, pp. 1–15, 2015.

[125] X. Zhu and I. Davidson, Eds., *Knowledge Discovery and Data Mining*, IGI Global, 2007.

[126] J. Žižka and F. Dařena, Eds., *Modern Computational Models of Semantic Discovery in Natural Language*, vol. 30. IGI Global, 2015.

[127] H. He, *Self-adaptive systems for machine intelligence*, New Jersey: John Wiley & Sons, Inc., 2011.

[128] R. Kohavi and F. Provost, "On Applied Research in Machine Learning," *Spec. Issue Appl. Mach. Learn. Knowl. Discov. Process*, vol. 30, 1998.

[129] A. Tesei, S. Fioravanti, P. Guerrini, V. Grandi, and A. Maguer, "Acoustic surveillance of small surface vessels in confined areas," *Int. Symp. Ocean Electron.*, 2011.

[130] P. O. Thompson, W. C. Cummings, and S. J. Ha, "Sounds, source levels, and associated behavior of humpback whales, Southeast Alaska," *J. Acoust. Soc. Am.*, vol. 80, no. 3, pp. 735–740, 1986.

[131] W. W. L. Au, A. A. Pack, M. O. Lammers, L. M. Herman, M. H. Deakos, and K. Andrews, "Acoustic properties of humpback whale songs," *J. Acoust. Soc. Am.*, vol. 120, no. 2, pp. 1103–1110, 2006.

[132] Y. Jia, G. Liu, and L. Zhang, "Bionic camouflage underwater acoustic communication based on sea lion sounds," *ICCAIS 2015 - 4th Int. Conf. Control. Autom. Inf. Sci.*, pp. 332–336, 2015.

[133] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[134] D. Nasien, H. Haron, and S. S. Yuhaniz, "Support Vector Machine (SVM) for English Handwritten Character Recognition," in *2010 Second International Conference on Computer Engineering and Applications*, 2010, pp. 249–252.

[135] T. C. Mota and A. C. G. Thome, "One-Against-All-based multiclass SVM strategies applied to vehicle plate character recognition," in *2009 International Joint Conference on Neural Networks*, 2009, pp. 2153–2159.

[136] K. Fukunaga, *Introduction to Statistical pattern Recognition*, 2nd editio. Academic Press, 1990.

[137] A. V. Oppenheim, "Speech Analysis-Synthesis System Based on Homomorphic Filtering," *J. Acoust. Soc. Am.*, vol. 45, no. 2, pp. 458–465, Feb. 1969.

[138] A. V. Oppenheim and R. W. Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 95–100, 2004.

[139] H. Wu, Y. Wang, and J. Huang, "Identification of Electronic Disguised Voices," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3, pp. 489–500, Mar. 2014.

[140] Guang Hua, J. Goh, and V. L. L. Thing, "Cepstral Analysis for the Application of Echo-Based Audio Watermark Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1850–1861, Sep. 2015.

[141] H. Malik and H. Zhao, "Recording environment identification using acoustic reverberation," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 1833–1836.

[142] Y. Wu, Y. Yang, C. Tao, F. Tian, and L. Yang, "Robust underwater target recognition using auditory cepstral coefficients," in *OCEANS 2014 - TAIPEI*, 2014, pp. 1–4.

[143] N. Sharma, "Speech Compression using Linear Predictive Coding (LPC)," *Int. J. Adv. Res. Eng. Appl. Sci.*, vol. 1, no. 5, pp. 16–28, 2012.

[144] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, vol. 50, no. April, pp. 637–655, 1971.

[145] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio Based Event Detection for Multimedia Surveillance," *2006 IEEE Int. Conf. Acoust. Speech Signal Process. Proc.*, vol. 5, pp. 813–816, 2006.

[146] J. B. Allen, "Harvey Fletcher's role in the creation of communication acoustics," *J. Acoust. Soc. Am.*, vol. 99, no. 4, pp. 1825–1839, Apr. 1996.

[147] J. Allen, "Cochlear modeling," *IEEE ASSP Mag.*, vol. 2, no. 1, pp. 3–29, 1985.

[148] T. Ganchev, *Contemporary Methods for Speech Parameterization*, Springer New York, 2011.

[149] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *J. Acoust. Soc. Am.*, vol. 8, no. 3, pp. 185–190, Jan. 1937.

[150] S. Gupta, J. Jaafar, W. F. wan Ahmad, and A. Bansal, "Feature Extraction Using Mfcc," *Signal Image Process. An Int. J.*, vol. 4, no. 4, pp. 101–108, Aug. 2013.

[151] M. Sayles and I. M. Winter, "Equivalent-rectangular bandwidth of single units in the anaesthetized guinea-pig ventral cochlear nucleus," *Hear. Res.*, vol. 262, no. 1–2, pp. 26–33, Apr. 2010.

[152] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," pp.1-25.

[153] C. C. Aggarwal, *Data classification : algorithms and applications.*, CRC Data Mining and Knowledge Discovery Series, 2015.

[154] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[155] G. Roffo and S. Melzi, "Ranking to learn: Feature ranking and selection via eigenvector centrality," *New Frontiers in Mining Complex Patterns*, *Lect. Notes Comput. Sci. Springer,* 2017.

[156] P. Yildirim, "Filter Based Feature Selection Methods for Prediction of Risks in Hepatitis Disease," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 4, pp. 258–263, 2015.

[157] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.

[158] M. Shardlow, "An Analysis of Feature Selection Techniques," *Univ. Manchester*, pp. 1–7, 2016.

[159] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[160] J. A. K. Suykens, I. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, *Advances in Learning Theory: Methods, Models and Applications*. IOS Press, 2003.

[161] R. W. Cottle, "William Karush and the KKT Theorem," *Doc. Math. · Extra Vol. ISMP*, vol. I, pp. 255–269, 2012.

[162] Z. Á. Mann, *Optimization in Computer Engineering: Theory and Applications*, 1st Editio. Scientific Research Publishing, 2011.

[163] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[164] R. Chiong, Ed., *Nature-Inspired Algorithms for Optimisation*, Springer Berlin Heidelberg, 2009.

[165] G. D. Taylor, Ed., *Logistics engineering handbook.*, CRC Press, 2007.

[166] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography," *Ann. Oper. Res.*, vol. 63, no. 5, pp. 511–623, Oct. 1996.

[167] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[168] S. Sivanandam, SN and Deepa, *Introduction to Genetic Algorithms*, Springer Berlin Heidelberg, 2008.

[169] X. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, 2012.

[170] J. H. W. Hain, G. R. Carter, S. D. Krauss, C. A. Mayo, and H. E. Winn, "Feeding behavior of the humpback whale, Megaptera novaeangliae, in the western North Atlantic," *Fish. Bull.*, vol. 80, no. January, pp. 259–268, 1982.

[171] H. Salimi, "Stochastic Fractal Search: A powerful metaheuristic algorithm," Knowledge-Based Syst., vol. 75, pp. 1–18, Feb. 2015.

[172] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer New York, 2007.

[173] M. Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, 2014.

List of Publications

[1]. B. M. Sherin, M. H. Supriya, and P. R. S. Pillai, "Underwater Acoustic Target Classification System Using SVM," Int. J. Electron. Commun. Eng., vol. 2, no. 5, pp. 73–80, 2013.

[2]. B. M. Sherin and Supriya M. H., "Selection and parameter optimization of SVM kernel function for underwater target classification," in 2015 IEEE Underwater Technology (UT), 2015, pp. 1–5.

[3]. B. M. Sherin and M. H. Supriya, "GA based selection and parameter optimization for an SVM based underwater target classifier," in 2015 International Symposium on Ocean Electronics (SYMPOL), 2015, pp. 28-34.

[4]. B. M. Sherin and M. H. Supriya, "SOS based selection and parameter optimization for underwater target classification," in OCEANS 2016 MTS/IEEE Monterey, 2016, pp. 1–4.

[5]. B. M. Sherin and M. H. Supriya, "WOA based Selection and Parameter Optimization of SVM Kernel Function for Underwater Target Classification," Int. J. Adv. Res. Comput. Sci., vol. 8, no. 3, pp. 223–226, 2017.

[6]. B. M. Sherin and M. H. Supriya, "Parameter Selection and Optimization of an SVM based underwater target classifier using Stochastic Fractal Search," in 2017 International Symposium on Ocean Electronics (SYMPOL), 2017, pp. 40–48.

[7]. B. M. Sherin and M. H. Supriya, "Parameter Selection and Optimization of an Underwater Target Classifier using Modified Symbiotic Organisms Search," : Communicated to Signal Processing: An International Journal (SPIJ).

APPENDIX

The Supervised Learning Model

The supervised learning model can be described using three components:

i. a generator of random vectors x, drawn independently from a fixed but unknown distribution $P(x)$;
ii. a supervisor that returns an output y, the class label for each x in the test dataset, according to a conditional distribution function $P(y|x)$, also fixed but unknown;
iii. a learning machine capable of implementing a set of functions $f(x,\alpha)$, $\alpha \in \Lambda$.

The problem of learning is that of choosing from the given set of functions $f(x,\alpha)$, $\alpha \in \Lambda$, the one which predicts the class label of the input in the best possible way. The selection is based on a training set of l random independent identically distributed (i.i.d.) observations $(x_1,y_1),\ldots\ldots\ldots\ldots(x_l,y_l)$, drawn according to $P(x,y) = P(x)\ P(y|x)$. The function $f(x,\alpha)$, is called a predictor, a hypothesis or a classifier. The response of the algorithm, measured by the error of the classifier is the probability that it does not predict the correct label y on a random input x, which cannot be calculated since the underlying distribution is unknown. The best available approximation to the response of the learning algorithm is a loss functional which measures the loss or disagreement $L(y, f(x, \alpha))$ between the response of the learning machine to a given input x, which is a randomly drawn new sample. The expected value of loss is given by the risk functional also called the expected risk, and is defined by equation A.1.

$$R(\alpha) = \int L\big(y, f(x,\alpha)\big) dP(x,y)) \qquad \text{A.1}$$

The goal is to find the function $f(x, \alpha_0)$ which minimises the risk functional $R(\alpha)$ from the class of functions $f(x,\alpha)$, $\alpha \in \Lambda$, in the situation where the joint probability distribution is unknown and the only available information is contained in the training set.

Statistical Learning Theory

According to Vapnik, a pioneer in statistical learning theory and the developer of SVM, there are three main problems in machine learning: pattern recognition, regression estimation and density estimation. In all cases, the goal is to choose a model from the hypothesis space, which is closest (with respect to some error measure) to the underlying function in the target space.

Considering the problem of binary pattern recognition, the principle behind classification. The classifier's output y can take only two values y = {0,1}, and let $f(x,\alpha)$, $\alpha \in \Lambda$, be the set of models or indicator functions. The classification problem can be expressed as

$$L\big(y, f(x,\alpha)\big) = \begin{cases} 0, if\ y = f(x,\alpha) \\ 1, if\ y \neq f(x,\alpha) \end{cases} \qquad \text{A.2}$$

The general setting of the learning problem can be described as follows. Let the underlying probability distribution of data, $P(z)$ be defined on the space Z. Consider the set of functions $Q(z, \alpha)$, $\alpha \in \Lambda$. The goal is to minimise the risk functional

$$R(\alpha) = \int Q(z,\alpha) dP(z), \qquad \alpha \in \Lambda \qquad \text{A.3}$$

where the probability measure $P(z)$ is unknown. However the i.i.d. sample $z_1, \ldots \ldots \ldots \ldots \ldots, z_n$, where z describes a pair (x,y) is available.

The learning problem considered minimises the risk functional given by equation A.3 where $Q(z,\alpha)$ is the specific loss function, on the basis of empirical data.

For the above classification problem, the risk functional given by equation A.3 requires the probability of classification error. Since the underlying probability distribution is unknown, the problem therefore is to find the model which minimises the classification error when only the data is known, and the underlying probability distribution is unknown.

Empirical Risk Minimization Induction Principle

As mentioned earlier, a typical machine learning task requires the data to be divided for training, validation and testing phases. During the training phase, the learning algorithm receives as input, a labeled training set sampled from an unknown distribution. The goal of the learning algorithm is to find the approximation (hypothesis) that minimises the error with respect to an unknown probability distribution and unknown dependency between the input and the output. The error of the classifier is the probability that it does not predict the correct label on a random data point generated by the underlying distribution. Since we do not know the underlying probability distribution and the dependency between the input and output, the expected value of error given by equation A. cannot be calculated. However, we can measure the empirical error or the training error, which is the average error incurred by the learning model due to wrong classification of samples drawn from already trained data. Hence, in order to minimise the risk functional, for an unknown probability measure, the empirical risk minimization (ERM)

induction principle is usually used. The expected risk functional given by equation A. is replaced by the empirical risk functional

$$R_{\text{emp}}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} Q(z, \alpha_l) \qquad \text{A.4}$$

constructed on the basis of i.i.d. training set.

The principle is to approximate the function $Q(z,\alpha_0)$ which minimises risk specified by equation A. by the function $Q(z, \alpha_l)$ which minimises empirical risk dictated by equation A.. This principle is called the Empirical Risk Minimization (ERM) induction principle. The difference between the expected and empirical risk is called generalization error and denotes the difference between error on the training set and error on the underlying joint probability distribution. It is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data.

Consistency of Empirical Risk Minimization Principle

The ERM principle is consistent for a set of functions $Q(z,\alpha)$, $\alpha \in \Lambda$ and for the probability distribution function $F(z)$ if the following two sequences converge in probability to the same limit.

$$R(\alpha_l) \xrightarrow[l\to\infty]{} \inf_{\alpha\in A} R(\alpha) \qquad \text{A.5}$$

$$R_{\text{emp}}(\alpha_l) \xrightarrow[l\to\infty]{} \inf_{\alpha\in\Lambda} R(\alpha) \qquad \text{A.6}$$

In other words, the ERM method is consistent if it provides a sequence of functions $Q(z,\alpha_l)$, $l = 1,2,\ldots\ldots$ for which both expected risk and empirical risk converge to the minimal possible value of risk $\inf_{\alpha\in\Lambda} R(\alpha)$ as depicted in Figure A.1. Equation A. asserts that the values of achieved risks converge to the best possible and equation A. asserts that we can estimate the minimal possible value of risk on the basis of the values of empirical risk.

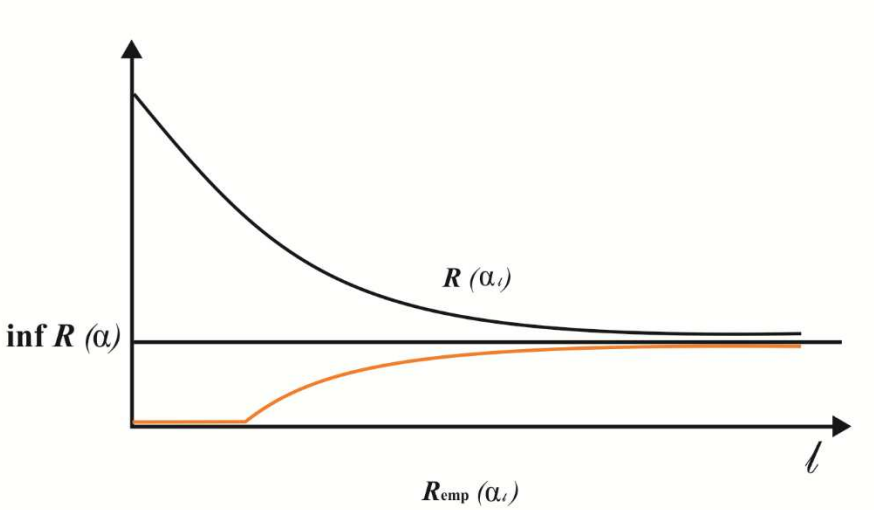

Fig. A.1 Consistency of the learning process

The above definition does not exclude trivial cases of consistency, that depend on whether the given set of functions contains a minorizing function. Therefore, in order to create a theory of consistency of the ERM method that would not depend on the properties of the elements of the set of functions, but would depend only on the general properties of this set of functions, the definition of consistency is to be modified to exclude the trivial consistency cases.

The modified definition for consistency states that the ERM method is nontrivially consistent for the set of functions $Q(z,\alpha)$, $\alpha \in \Lambda$ and the probability function $P(z)$, if for any non empty subset $\Lambda(c)$, $c \in (-\infty,\infty)$ of this set of functions defined as

$$\Lambda(c) = \{\alpha: \int Q(z,\alpha)dF(z) > c, \alpha \in \Lambda\} \qquad \text{A.7}$$

the convergence

$$\inf_{\alpha\epsilon\Lambda(c)} R_{\text{emp}}(\alpha) \underset{l\to\infty}{\longrightarrow} \inf_{\alpha\epsilon\Lambda(c)} R(\alpha) \qquad \text{A.8}$$

is valid. In other words, the ERM is nontrivially consistent if it provides convergence for the subset of functions that remain after the functions with the smallest values of risks are excluded from this set.

The conditions for non trivial consistency are described by the theory of consistency of learning process which describes the necessary and sufficient conditions for convergence of the ERM inductive principle.

The Theory of Consistency of Learning Process

The theory of consistency is an asymptotic theory. It describes the necessary and sufficient conditions for the convergence of solutions obtained using ERM method to the best possible as the number of observations is increased.

The Key Theorem of the Learning Theory

The Key theorem of the learning theory was proposed by Vapnik and Chervonenkis in 1989. It advocates that the conditions required for the consistency of ERM principle are equivalent to the conditions for existence of uniform one sided convergence.

The Key theorem is stated as follows. Let $Q(z,\alpha)$, $\alpha \in \Lambda$ be a set of functions that as a bounded loss for probability measure $P(z)$.

$$A \leq \int Q(z,\alpha)dP(z) \leq B \qquad \forall \alpha \in \Lambda \qquad \text{A.9}$$

Then for the ERM principle to be consistent, it is necessary and sufficient that the empirical risk $R_{\text{emp}}(\alpha)$ converge uniformly to the actual risk $R(\alpha)$ over the set $Q(z,\alpha)$, $\alpha \in \Lambda$ as follows

$$\lim_{l\to\infty} \text{prob}\left\{\sup_{\alpha\in\Lambda}(R(\alpha) - R_{\text{emp}}(\alpha)) > \varepsilon\right\} = 0 \qquad \forall\varepsilon \qquad \text{A.2}$$

This type of convergence is called uniform one sided convergence. The theorem is called the key theorem because it asserts that the conditions for consistency of ERM principle are necessarily (and sufficiently) determined by the worst function over the set of functions given by equation A.3, converges in probability to zero.

$$\Delta(\alpha_{worst}) = sup_{\alpha \epsilon A}\big(R(\alpha) - R_{emp}(\alpha)\big) \qquad \text{A.3}$$

Therefore with the above condition, from this theorem it follows that the analysis of the ERM principle requires an analysis on the properties of uniform convergence of the expectations to their probabilities over a given set of functions.

Conditions for Uniform Convergence

For uniform two sided convergence of the frequencies to their probabilities

$$\lim_{l\to\infty} \text{Prob}\{\sup_{\alpha\epsilon\Lambda}|R(\alpha) - R_{\text{emp}}(\alpha)| > \varepsilon\} = 0 \qquad \text{A.4}$$

where $R_{\text{emp}}(\alpha)$ defines frequency and $R(\alpha)$ defines probability, it is necessary and sufficient that the equality defined in equation A.5 be valid.

$$\lim_{l\to\infty} \frac{H^{\Lambda}(l)}{l} = 0, \quad \forall\varepsilon > 0 \qquad \text{A.5}$$

where $H^{\Lambda}(l)$ is the expectation of random entropy over the joint distribution function $P(z_1,\ldots\ldots,z_l)$

$$H^{\Lambda}(l) = E \ln N^{\Lambda}(z_1, \ldots\ldots\ldots, z_l) \qquad \text{A.6}$$

where $N^{\Lambda}(z_1, \ldots\ldots\ldots, z_l)$ is the number of different vertices of the l - dimensional cube that is obtained on the basis of sample $z_1,\ldots\ldots,z_l$ and the set of functions $Q(z,\alpha)$, $\alpha \in \Lambda$. In other words, $N^{\Lambda}(z_1, \ldots\ldots\ldots, z_l)$ represents the number of different separations of the sample that can be obtained using functions from the given set of indicator functions.

Equation A.5 describes the necessary and sufficient condition for consistency of the ERM principle, which should be satisfied by any learning machine minimizing the empirical risk.

Conditions for Fast Convergence of the ERM Principle

The annealed VC entropy $H^{\Lambda}_{\text{ann}}(l)$ for sets of indicator functions is defined as

$$H^{\Lambda}_{ann}(l) = E\, ln\, N^{\Lambda}(z_1, \ldots\ldots\ldots, z_l) \qquad \text{A.7}$$

and the growth function $G^{\Lambda}(l)$ is defined as

$$G^{\Lambda}(l) = \ln \sup_{z_1,\ldots\ldots,z_l} N^{\Lambda}(z_1, \ldots\ldots\ldots, z_l) \qquad \text{A.8}$$

These functions are determined in such a way that for any l inequalities, $H^{\Lambda}(l) \leq H^{\Lambda}_{\text{ann}}(l) \leq G^{\Lambda}(l)$ is valid. The asymptotic rate of convergence is fast if for any $l > l_0$ dictated by equation A.9 holds true.

$$P\{R(\alpha_l) - R(\alpha_0) > \varepsilon\} \; < \; e^{-c\varepsilon^2 l} \qquad \text{A.9}$$

where c is a constant greater than zero.

The sufficient condition for fast convergence is given by equation A.

$$\lim_{l\to\infty} \frac{H^{\Lambda}_{ann}(l)}{l} = 0 \qquad \text{A.18}$$

The necessary and sufficient condition for fast convergence is given by equation A..

$$\lim_{l\to\infty} \frac{G^{\Lambda}(l)}{l} = 0 \qquad \text{A.19}$$

The above condition specified in equation A. is independent of probability distribution and also describes the necessary and sufficient condition for consistency of ERM for any probability measure.

Equation A. describing the sufficient condition for fast convergence and equation A.5 describing the necessary and sufficient condition for consistency are valid only for a given probability measure $P(z)$. However, equation A. describes the necessary and sufficient conditions for consistency of learning machine implementing ERM as well as the sufficient condition for fast convergence.

VC Dimension

The growth function $G^{\Lambda}(l)$ has a remarkable property that it either satisfies the equality specified by equation A. or is bounded by the inequality specified by equation A..

$$G^{\Lambda}(l) = l \ln 2 \qquad \text{A.20}$$

$$G^{\Lambda}(l) < h\left(\ln \frac{l}{h} + 1\right) \qquad \text{A.21}$$

where h is an integer for which

$$G^{\Lambda}(h) = h \ln 2 \qquad \text{A.102}$$

$$G^{\Lambda}(h+1) \neq (h+1) \ln 2 \qquad \text{A.11}$$

In other words, the growth function will be either a linear function or will be bounded by a logarithmic function. This is depicted in Fig. A..

The VC dimension of the set of indicator functions $Q(z,\alpha)$, $\alpha \in \Lambda$ is infinite if the Growth function for this set of functions is linear. An alternate way of defining VC dimension is as follows. The VC dimension of a set of indicator functions $Q(z,\alpha)$, $\alpha \in \Lambda$ is the maximum number h of vectors z_1, ……., z_h which can be separated in all $2h$ possible ways. If for any n there exists a set of n vectors which can be shattered by the set, then the VC dimension is equal to infinity.

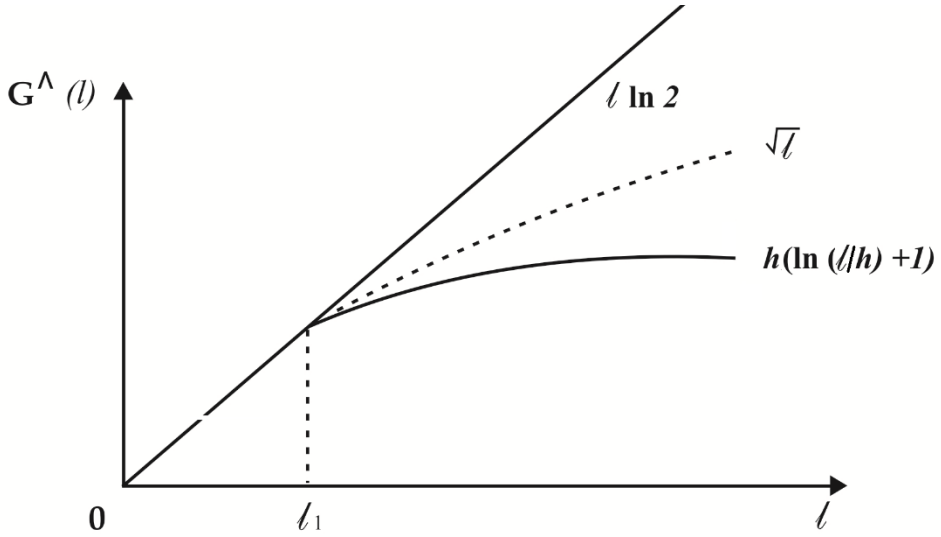

Fig. A.2 The growth function is either linear or bounded by a logarithmic function. It cannot, for example behave like as in dashed line

Distribution Independent Bounds for the Rate of Convergence of Learning Process

Consider a set of totally bounded loss functions $Q(z,\alpha)$, $\alpha \in \Lambda$, which possess a finite VC dimension h described by equation A.24.

$$0 \leq Q(z,\alpha) \leq B, \qquad \alpha \in \Lambda \qquad \text{A.12}$$

By the theory of bounds for sets of totally bounded functions the inequality in equation A.13 holds true with a probability of atleast $1-\eta$ for all functions described by equation A.12.

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{B\varepsilon}{2}\left(1+\sqrt{1+\frac{4R_{emp}(\alpha)}{B\varepsilon}}\right) \qquad \text{A.13}$$

$$\text{where } \varepsilon = 4\frac{h\left(ln\frac{2l}{h}+1\right)-ln\,\eta}{l}$$

The theory of bounds provides bounds for risks of all indicator functions for $Q(z,\alpha)$, $\alpha \in \Lambda$, described by equation A.146.

$$I(z,\alpha,\beta) = \theta\{Q(z,\alpha)-\beta\}, \quad \alpha \in \Lambda \qquad \text{A.14}$$

where $\alpha < \beta < A$ is some constant and $\theta(u)$ is a step function.

The bounds follow from the bound on uniform convergence for sets of totally bounded functions that have finite VC dimension.

Structural Risk Minimization Induction Principle

The ERM principle is intended for dealing with a large sample size. The sample size *l* is considered to be large if the ratio, *l*/*h* (ratio of the number of training patterns to the VC dimension of functions of a learning machine) is large. The ERM principle can be justified by considering the inequalities in equation A.13. When *l*/*h* is large, the second summand on the right side of inequality in equation A.13 becomes small. The actual risk is then close to the value of empirical risk. A small value of empirical risk provides a small value of expected risk.

However, if *l*/*h* is small, a small $R_{emp}(\alpha_l)$ does not guarantee a small value of the actual risk. In this case, to minimise the actual risk $R(\alpha)$, one has to minimise the right hand side of inequality in equation A.13 simultaneously over both terms, one which depends on the value of the empirical risk while the second depends on the VC dimension of the set of functions. However, the ERM principle does not consider the capacity of the learning machine and hence tends to overfit the data. Predictions done by ERM are often unable to generalise well and prone to overfitting. This is due to the fact that

there can be infinitely many functions that have minimal risk, amongst which a single unique function will have the highest generalization ability. Hence, in order to develop a model that has good generalization capabilities and works well on previously unseen data, a principle which takes into account, the capacity of the learning machine should be developed. Thus the minimization of $R(\alpha)$ requires a new principle, based on the simultaneous minimization of two terms in equation A.13. To minimise risk, it is necessary to find a method which, along with minimizing the value of empirical risk, controls the VC dimension of the learning machine. Structural Risk Minimization (SRM) principle which is described in the following, is intended to minimise the risk functional with respect to both empirical risk and VC dimension of the set of functions.

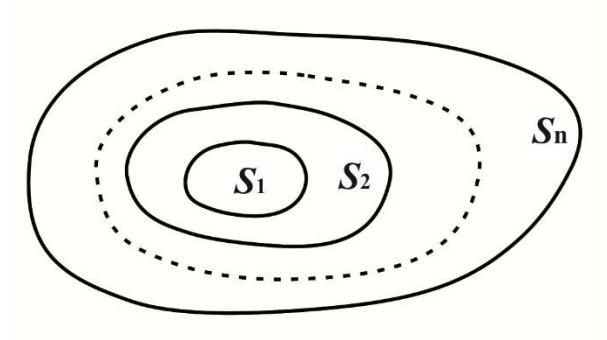

Fig. A.3 A structure on the set of functions is determined by the nested subsets of functions

Let the set S of functions $Q(z, \alpha)$, $\alpha \in \Lambda$, be provided with a structure consisting of nested subsets of functions $S_k = \{ Q(z, \alpha), \alpha \in \Lambda_k \}$ as depicted in Fig. A., such that $S_1 \subset S_2 \subset \ldots\ldots\ldots \subset S_n$ and $S^* = \cup_k S_k$.

An admissible structure is one satisfying the following three properties

i. The set S^* is everywhere dense in S
ii. The VC dimension h_k of each set S_k of functions is finite

iii. Any element S_k of the structure contains totally bounded functions, $0 \leq Q(z, \alpha) \leq B_k, \alpha \in \Lambda_k$

For a given set of observations z_1,……..,z_l the SRM principle chooses the function $Q(z, \alpha_l^k)$ minimizing the empirical risk in the subset S_k for which the guaranteed risk is minimal.

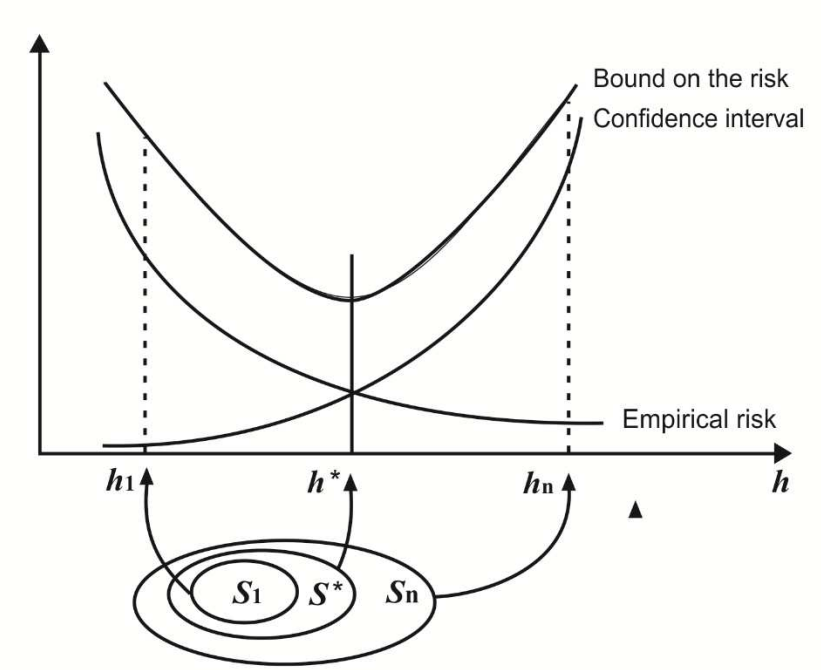

Fig. A.4 Bound on the risk: The bound on the risk is the sum of empirical risk and the confidence interval. The empirical risk decreases with the index of the element of the structure, while the confidence interval increases.

The SRM principle defines a trade off between the quality of the approximation of the given data and the complexity of the approximating function. As the subset index *n* increases indicating an increase in the complexity of the learning machine, the minima of the empirical risks decrease. However, the term responsible for the confidence interval (the second summand in inequality of equation A.13) increases. This is depicted in Fig. A.. The SRM principle takes both factors into account by choosing

the subset S_n for which minimizing the empirical risk yields the best bound on the actual risk. For any distribution function the SRM method provides convergence to the best possible solution with probability one. A learning machine implementing SRM will yield a classifier with minimal model complexity and risk. In other words SRM method is universally strongly consistent.

ADDENDUM

The addendum to the thesis entitled 'Underwater Target Classifier with Improved Success Rate using Meta-Optimal Support Vector Machines' contains relevant information about the thesis, which was inadvertently omitted during the preparation of the thesis.

Sources of Data

The database used in the classifier consists of noises of 11 classes of acoustic targets. Data of different targets have been recorded during scheduled cruises conducted off Cochin and Mangalore. For incorporating the channel effects and the environment, ambient noise has also been recorded. Other recordings have been collected from the open source databases available in the internet.

Man Made Noises

(a) Ships:

For the proposed study, three sound signatures of ships have been collected and labelled as Ship1, Ship2 and Ship3. The noise Ship1 is a recording of a commercial ship cruising at approximately 20 knots and about 3.2 km away from the hydrophone and Ship2 is a recording of a Merchant vessel in the Cochin Shipyard which was recorded as the vessel was approaching from 1.7 km away. The noise Ship3 was collected from an open source database in the internet.

(b) Boats:

Small boats such as Zodiac, have outboard motors whose propellers creates sound and these boats are popular in coastal waters. The propeller

produces a cavitation noise which is at higher frequencies than larger vessels, as they have high rotation rates. The database in the thesis included noises from 4 boats, labelled as Boat1, Boat2, Boat3 and Boat4. The noise Boat1 and Boat2 are recordings of a Zodiac with a 35 HP engine and 50HP engine respectively. The noises Boat3 and Boat4 were collected from open source databases in the internet.

(c) Humpback Whale:

Humpbacks are best known for their vocalizations and is one of rorquals which have two characteristics in common , *viz.* dorsal fins on their back, and ventral pleats running from the tip of the lower jaw back to the belly area. The noise of humpback whale has been collected from an open source database in the internet.

(d) Sealion:

Sea lions are sea mammals of the family Otariidae. Sea lions haul out in large colonies on rocks and sandy shores on the Islands. The noise of sealions has been collected from an open source database in the internet.

(d) Snapping Shrimps:

Snapping shrimp produce sound by snapping of their claws. The sound produced by a shrimp colony can be so loud that the sonars may miss other nearby targets. The noise produced by a shrimp colony has been collected from an open source database in the internet as well as recorded using hydrophones in the costal shores of Vypeen, Kochi.

Classifier training and testing for varying SNR

The classifier has been trained with single signal at a time. For the testing phase, the classifier is fed with a simulated signal which is actually

an additive combination of the target signal and the ambient noise recorded from the sea while the team had gone for sea trials. The test bench created had different inputs like simulated signals, the target signal corrupted by ambient noise collected from Arabian sea as well as pure signals.

For incorporating the channel effects and the environment, this ambient noise has been additively added to the target signal noise and all the algorithms has been tested for the simulated signal which is a combination of original signal and ambient noise. Since it was the real data collected during the sea trial that was added, performance of the classifier under other noise conditions with varying SNR has not been studied.

Performance in Active Scenario

The prototype classifier has been developed for a **passive sonar scenario**. In active sonar scenario, factors like Reflection, Reverberation, Multipath, Scattering, Doppler effects etc. will be influencing the target signal strength. The underwater channel effects will have to be considered and signals will have to be appropriately pre-processed before passing on to the classifier. The basic methodology of the proposed classifier is expected to yield good results in the active scenario also. However, this scenario has not been considered in the thesis.

Population Size

The optimal population size of a meta-heuristic algorithm is dependent on a number of factors including the number of generations and the problem to which the algorithm is applied. Commonly for low dimensional optimization tasks ($d < 100$), a population size $n > d$ is adopted as mentioned in O. Roeva *et al.* in the paper *Influence of the Population Size on the Genetic Algorithm Performance in Case of*

Cultivation Process Modelling. S. Chen *et al.* in the paper *Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution*, had studied the effect of the influence of population size on the GA performance in which he had observed that increasing the optimal population size for a fixed number of iterations does not improve the quality of solutions. The same has been observed in our work. The stopping criteria of all the optimization algorithms viz. GA, BAT, WOA, SFS, SOS and *m*-SOS employed in this thesis, was either 500 iterations or when the average relative change in the fitness value is stalled over 25 iterations. While the prototype was implemented, the optimal population size of GA, WOA, SFS and *m*-SOS was found to be 20 and that for BAT and SOS was found to be 15. The optimal population size can vary for a different stopping criteria and a different task.

Computational Complexity

The simulation has been carried out on a system with Intel Core i3 CPU (M350@2.27GHz) with 4 GB RAM installed with 64-bit Windows 7 Professional Operating System. As we have followed a supervised classification scheme, the complete procedure involves training and testing phase. The training phase involves a considerable computational cost which included the time for selection of features. However, during the testing phase, the system can achieve near real time performance as the SVM classifier is already trained.

Courtesy to figures

The author would also like to give courtesy to the following figures which have been reproduced from textbooks and articles:

Fig. 1.2 A. D. Waite/ Sonar for Practising Engineers, John Wiley & Sons

Fig. 3.6 C. Neumuller et al./ Proceedings of the 13th International Conference on Computer Aided Systems Theory, 1(2011), pp.367-374.

Fig. 4.32 Juan Tapia Farias/ Article entitled 'Feature Selection Methods' available in the internet

Fig. 4.33 R. Kohavi, G. H. John/ Artificial Intelligence, 97(1997), pp.273-324.

Fig. 4.38 T. Hastie *et al.*/ The Elements of Statistical Learning, Springer Series in Statistics

Fig. 4.43 Ben Aisen/ Article entitled 'A Comparison of Multiclass SVM Methods', (2016) available in the internet

Fig 4.44 Ben Aisen/ Article entitled 'A Comparison of Multiclass SVM Methods', (2016) available in the internet

Fig. 5.1 Leon fedden/ Article entitled 'The No Free Lunch Theorem' available in the internet

Fig. 5.2 S. Mirjalili, A. Lewis/ Advances in Engineering Software, 95(2016), pp.51-67.

Fig. 5.7 S. Mirjalili, A. Lewis/ Advances in Engineering Software, 95(2016), pp.51-67.

Fig. A.1 V. N. Vapnik/ The Nature of Statistical Learning Theory, Springer

Fig. A.2 V. N. Vapnik/ The Nature of Statistical Learning Theory, Springer

Fig. A.3 V. N. Vapnik/ The Nature of Statistical Learning Theory, Springer

Fig. A.4 V. N. Vapnik/ The Nature of Statistical Learning Theory, Springer