Requirements Engineering (RE) is “The area of Software Engineering that focuses on the RE process which involves understanding customer needs and expectations (requirements elicitation), requirements analysis and specification, requirements prioritization, requirements derivation, partitioning and allocation, requirements tracing, requirements management, requirements verification, and requirements validation” (Young 2003). The terms Requirements and Engineering were first tied together by Alford (Alford 1977) in the development of SREM (Software Requirements Engineering Method). RE was first applied to information systems, and therefore was more focused towards organizational and application issues. Since the word Engineering has been attached to requirements, RE research efforts have endeavored to incorporate an Engineering approach to what was traditionally known as Systems Analysis. Originally, RE activities were only related to Requirements Analysis and Specification, the core of the first stage of the Software Development Life Cycle (SDLC). In the 1990s, it was widely accepted that RE was a key process in the software lifecycle. RE is now an established discipline and includes a variety of skills, processes, methods, techniques and tools.

In general, it is assumed that the requirements for a system are clearly specified before start of its design and implementation. Mistakes
during requirement process cause undue cost overruns, delays and many project failures. Requirements can be broadly classified into Functional Requirements (FR) and Non Functional Requirements (NFR). FR deals with requirements that affect the functionality of the system whereas NFR deals with requirements that constrain the system. NFRs are also part of FR and may appear regularly when FR is being elicited. Since NFR are part of FR its discovery is sometimes missed out during the initial stages of the SDLC. The different views of stakeholders on a NFR gives rise to ambiguity on the system wide NFR. Tracing NFR from FR is a time consuming task and requires experienced analyst to identify NFR hidden in business requirements.

Techniques using Data Mining have been proposed for retrieval of NFR from FR with fairly good results. However, effort has not been made in the direction of improving Information Retrieval (IR) algorithms. This specific area motivated this research to propose IR based algorithms based on Soft Computing techniques.

Non Functional Requirements (NFR) of software evaluates the operation of the system and imposes constraints on design and implementation to maintain the overall quality. Incorporating RE to identify NFR in early stages avoids ambiguities, conflicting requirement and other defects. Number of automated tools is available for identifying NFRs. The efficacy of the RE methodology is its ability to capture NFR in an efficient manner. This research can be broadly classified into the following sections.
• Investigation of classification accuracy of Meta learning algorithms including Logitboost and Bagging.

• Investigate effectiveness of Neural Network for identifying NFR features.

• Propose an improved Neural Network architecture Hidden Layer Genetic Optimized Recurrent Neural Network (HLGO - RNN) and Hidden Layer Genetic Optimized Generalized Feed Forward Neural Network (HLGO - GFFNN) for identifying NFR features.

• Propose a feature extraction technique based on NFR Repository.

• Propose a Hybrid Genetic Algorithm Continuous Ant Colony Optimization (HGACACO) which incorporates both Genetic Algorithm and Ant Colony Optimization.

Dataset available in the promise repository was used to evaluate the proposed methodologies in this work. Features were extracted using the term document frequency and 57 words selected based on its importance. Precision, Recall, F measure and Classification accuracy were measured. Using Meta learners classification accuracy in the range of 59.29% to 82.37% were obtained. In the next stage of this research Neural Network techniques were investigated for its efficacy to classify NFR from FR. Investigated
Neural Networks include Multi Layer Perceptron (MLP) and Generalized Feed Forward Neural Network (GFFNN). The classification accuracy obtained was in the range of 73.72% to 85.26%, showing an improvement of 3.5% over Meta learning algorithms that were tested. To further improve the classification accuracy, optimization of the learning rate and momentum was proposed using Genetic Algorithm (GA). Classification accuracy obtained was in the range of 81.25% to 90.23% showing an improvement of 9.54%.

A feature extraction is proposed, a NFR repository is built and features are selected based on the repository to retrieve NFR from documents for classification. To further improve the classification accuracy, a Hybrid Genetic Algorithm Continuous Ant Colony Optimization (HGACACO) which incorporates both Genetic Algorithm and Ant Colony Optimization (ACO) is proposed. The proposed techniques are benchmarked with the dataset from promise repository and evaluated using a new dataset. HGACACO FNN classifier with proposed feature extraction technique achieves a classification accuracy of 93.59% for the promise repository dataset and an accuracy of 91.06% for the new dataset.