ABSTRACT

Today, the software market forces to develop bigger, better and faster applications in ever quicker times and it is in direct contest with the development teams' need for longer times to develop these same applications. Reducing development and maintenance time is progressively important objective for developers of software. Automatic code generation, Generative Programming and Machine Learning in software engineering are some of the approaches in achieving this goal and the study focuses these methodologies.

Generative Programming is a paradigm for engineering software system families such that, given particular requirements specification, a customized and optimized system can be automatically constructed from elementary, reusable implementation components by means of configuration knowledge. The goal of generative programming in software engineering is to increase the productivity, quality, and time in terms of system families rather than single systems.

Inappropriately, earlier art for code generation has been limited to single system engineering and thus, using code generation to build specific systems to solve specific problems. As a result, previous methods and techniques used in Generative System Development provide limited solutions in terms of software components, which are generated and limited suppleness or variance in those generated components.
In this research, attempts are made to automate software production by means of code generation to create a more efficient product and a study is done on how automated code generation plays a vital role in various fields of computing. The research portrays how code can be generated from design and further explains how code generation plays a role in generic database form generation and Web Search.

The study also brings out the importance of real time code generation and a model for the real time code generation. To facilitate real time code generation and to avoid single system engineering, a relational data set is developed to store the algorithms in an intuitive phase and a search algorithm is proposed that works on this collection to choose the best algorithm and give the code as output. The investigation also focuses on how optimization in software production can be achieved by using Generative programming.

Machine learning deals with issues of how to build programs that improve their performance at some task through experience and it helps greatly in automation. Not amazingly, the field of software engineering turns out to be a productive ground where many software development and maintenance tasks could be devised as learning problems and approached in terms of learning algorithms.

Thus, in this exploration, attempts are made to apply machine learning in software engineering. The research offers guidelines on applying machine learning for work-force modularization, followed by cost prediction,
requirement engineering and for the maintenance phase. Finally to identify the risk and correctness assessment, procedure using supervised machine learning is proposed. The examination is made to produce an efficient agent for code reusability for the software developers using the recommendation systems and algorithms. The investigations are concluded as the future of software engineering and need of software production can be achieved by code generation, generative programming and machine learning.