

Software Fault Prediction Using Feature Selection Algorithms

Prepared by :

Sirisha Medicharla – 2020UGCS110

Shubam Kumar – 2020UGCS014

Praphul – 2020UGCS101

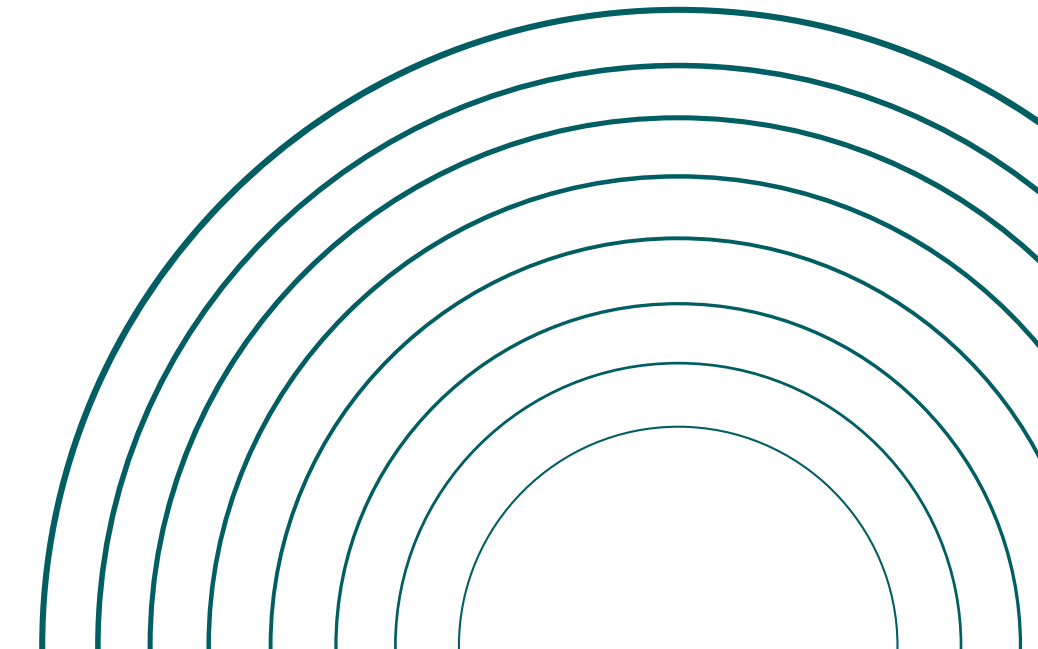

**Under the guidance of
Dr. B Ramachandra Reddy**

ROLE OF GROUP MEMBERS

- **Sirisha Medicharla (2020UGCS110)**
 - Code related work and Deployment
- **Shubham Kumar (2020UGCS014)**
 - Report
- **Praphul (2020UGCS101)**
 - Presentation



CONTENTS

- **Introduction**
 - **Motivation**
 - **Problem statement**
 - **Related work**
 - **Methodology**
 - **Results**
 - **Conclusion**
 - **Future Work**
- 
- 

INTRODUCTION

- Software development follows a set of steps to produce reliable and high-quality software. Faults can occur at any time during the software development process
- Software fault prediction (SFP) helps to predict faults early in the development phase and is helpful in improving the software quality of the final product in a fast and cost effective manner
- Machine learning algorithms, such as Random Forest, SVM, and Neural Networks, have been applied to fault detection in software systems.
- Overall, there are various approaches and techniques that can be used for software fault prediction, and the choice of technique depends on the characteristics of the software system and the specific goals of the fault detection process.



MOTIVATION



- The motivation of our project is to explore the use of machine learning techniques, specifically FeatBoost, PSO, and genetic algorithms, for predicting software faults.
- This research seeks to provide insights into the effectiveness of FeatBoost, PSO, and genetic algorithms for software fault prediction and applying these techniques on a real-world dataset.
- Machine learning has showed promise in this area, as it can help automate the process of finding potential flaws and minimize the time and resources needed for testing and debugging.
- Ultimately, the goal of our project is to contribute to the ongoing effort to improve the reliability and quality of software systems.

PROBLEM STATEMENT

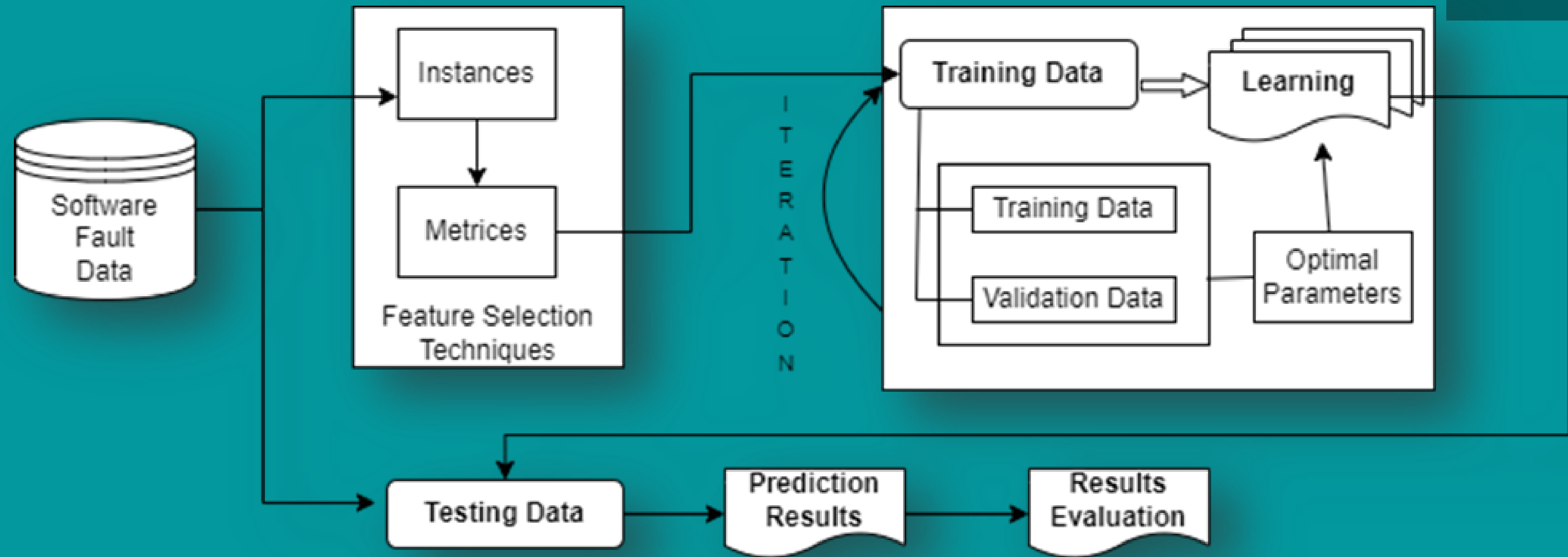
- Develop a machine learning-based software fault prediction model to identify potential faults during software development.
- The model should analyze software metrics and provide early warnings to prevent or mitigate faults, reducing maintenance costs.



RELATED WORK

- In this section, we discuss significant research on software fault prediction using machine learning approaches.
 - We further highlight our reasons to select particular Machine learning techniques to propose our algorithms for Software Fault Prediction(SFP).
 - Singh et al. (2020) proposed a software fault prediction approach based on a hybrid algorithm that combined genetic algorithms and support vector machines
 - They evaluated their approach on a publicly available dataset and found that it achieved higher accuracy than other machine learning algorithms, such as decision trees and random forests.
 - However, they also noted that genetic algorithms can be computationally expensive for large datasets.
- 
- 

PROPOSED METHOD



METHODOLOGY

- We have applied the following steps to perform feature selection with the Feature Boost algorithm and classification with Random Forest and SVM algorithms:
- **Data collection** : Collect data related to software development, such as source code, bug reports, and change history
- **Feature extraction** : Extract features from the collected data, such as software metrics, change frequency, and code complexity.
- **Feature selection** : Select a subset of relevant features that are most predictive of software faults using techniques such as correlation analysis, information gain, and wrapper methods
- **Model training** : Train a machine learning model on the selected features and a labeled dataset of software faults to predict the likelihood of faults in new code.



Cont'd


- **Model evaluation** : Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score.
- Compare the performance of different models and feature selection methods to identify the best approach.
- **Model deployment** : Deploy the trained model in the software development process to assist developers in identifying potential faults and improving software quality.

FEATURE SELECTION

- FeatBoost is a boosting algorithm that focuses on learning a set of informative features for gesture recognition
- The key idea behind FeatBoost is to adaptively select a subset of features for each gesture from a large pool of candidate features.
- The algorithm assigns weights to each feature and trains a weak learner on the weighted features.
- It then updates the feature weights based on the classification performance of the weak learner and repeats the process until a strong classifier is obtained
- FeatBoost has been shown to outperform other state-of-the-art gesture recognition algorithms on several benchmark datasets



Particle Swarm Optimization (PSO)

- It is a metaheuristic optimization algorithm that was inspired by the social behavior of bird flocking or fish schooling
 - PSO is a population-based algorithm that employs a swarm of particles moving around in a search space, where each particle represents a potential solution to the problem.
 - In PSO, each particle has a position and a velocity, which are updated in each iteration of the algorithm.
 - The position of each particle represents a candidate solution, and the velocity determines the direction and magnitude of the particle's movement
 - The objective of PSO is to find the optimal solution by iteratively updating the position and velocity of each particle based on its own experience and the experience of the swarm.
 - One of the main advantages of PSO is its simplicity and fast convergence rate
- 

Genetic Algorithms (GA)

- Genetic Algorithms are a type of optimization algorithm inspired by the process of natural selection in biology.
- GAs work by simulating the process of evolution, in which a population of potential solutions undergoes selection, reproduction, and mutation to produce better solutions over time.
- In a typical GA, an initial population of solutions is randomly generated, and each solution is evaluated using a fitness function that measures its quality
- Solutions with better fitness scores are more likely to be selected for reproduction, in which their genetic material is combined to produce new solutions
- They are effective at finding high-quality solutions to complex problems, especially those with a large search space or non-linear relationships between variables.
- However, GAs also has some limitations, including the risk of premature convergence and difficulty in handling constraints

MACHINE LEARNING ALGORITHMS USED FOR CLASSIFICATION

Random Forest :

- Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems.
- It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
- One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification.
- It performs better for classification and regression tasks.

Support Vector Machine (SVM):

- Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification and regression tasks.
- The main idea behind SVM is to find the best boundary that separates the data into different classes.
- The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.
- The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line.

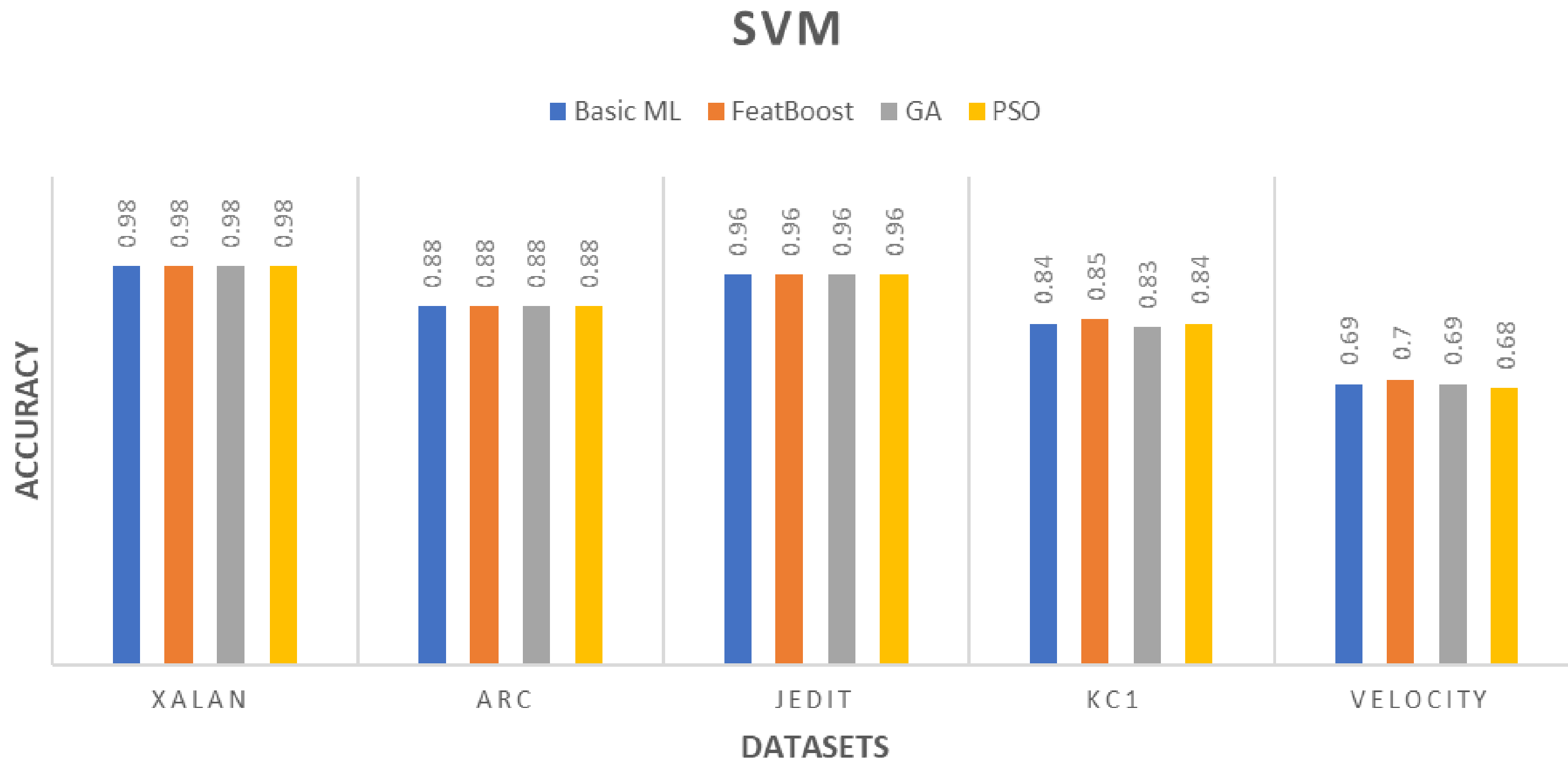
DATA SET

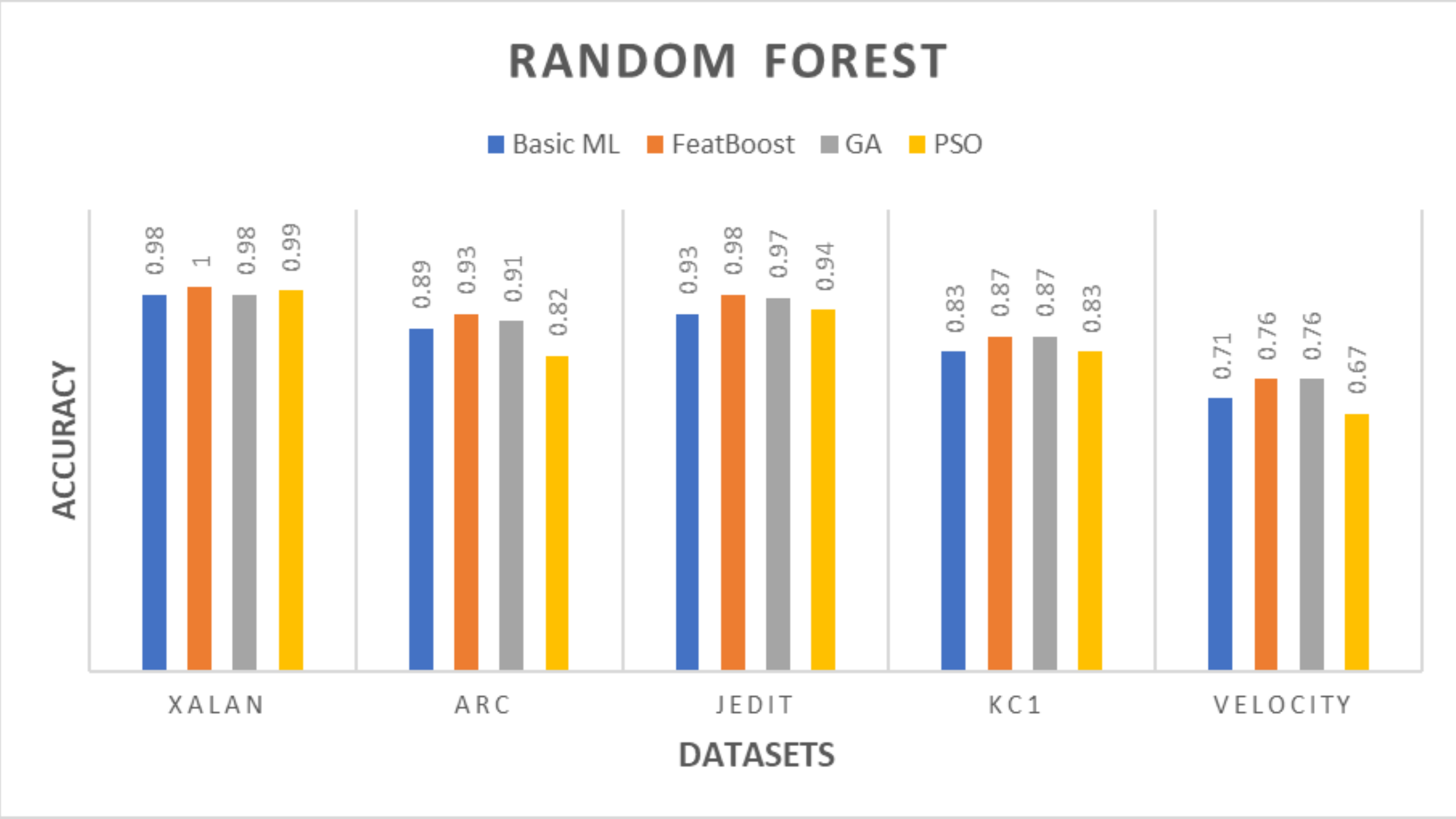
LINK TO DATASET:

<https://ieee-dataport.org/documents/software-defect>

- Dataset nearly contains 19110 number of observations .
- There are a total of 21 columns in our dataset.
- The all 21 columns are :
weight methods per class (wmc), response for a class(rfc),Data abstraction coupling(Dac), Lack of cohension(Loc),Attribute hiding class(Ahc), No of children(Noc),Depth of inheritance(Dot) and etc..


RESULTS





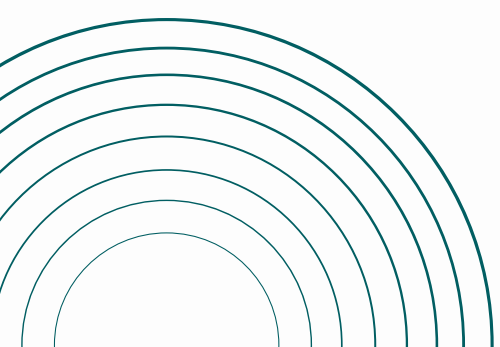
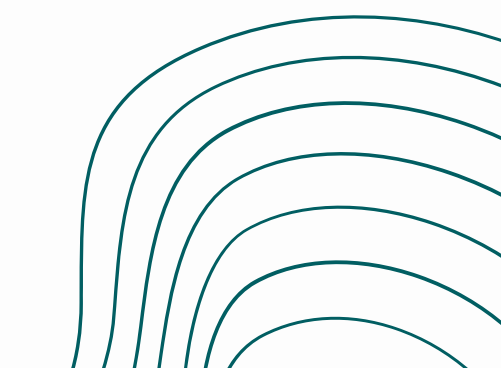


CONCLUSION

- In this study, we conducted a comparative analysis of software fault prediction techniques using FeatBoost, PSO, and genetic algorithms.
 - We utilized 5 dataset of software metrics to evaluate the performance of the three techniques and found that FeatBoost outperformed PSO and genetic algorithms in terms of accuracy.
 - Furthermore, we conducted a literature review of previous studies in software fault prediction and found that many researchers have utilized similar techniques to our study, but few have compared them directly.
 - Our study fills this gap in the research and provides valuable insights into the relative performance of these techniques
- 



Future Work

- First, we focused only on three techniques: FeatBoost, PSO, and genetic algorithms.
 - There are numerous other techniques for software fault prediction, such as decision trees, and neural networks, that could be compared in future studies.
 - Second, we evaluated the techniques based solely on their predictive performance, without considering the interpretability or complexity of the resulting models.
 - Future studies should build on this work by utilizing multiple datasets, comparing a wider range of techniques, and exploring the trade-offs between predictive performance and model interpretability or complexity
- 
- 

REFERENCES...

- Rathore, S.S., Kumar, S. A study on software fault prediction techniques. Artif Intell Rev 51, 255–327 (2019). <https://doi.org/10.1007/s10462-017-9563-5>
- Alsahaf, A., Azzopardi, G., Ducro, B., Veerkamp, R. F., & Petkov, N. (2018). Predicting slaughter weight in pigs with regression tree ensembles. In APPIS (pp. 1–9).
- https://www.researchgate.net/publication/338434732_Optimization_Algorithms_to_Solve_Feature_Selection_Problem_A_Review
- Venkatesh, B. and Anuradha, J.. "A Review of Feature Selection and Its Methods" Cybernetics and Information Technologies, vol.19, no.1, 2019, pp.3-26. <https://doi.org/10.2478/cait-2019-0001>
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785–794)

THANK YOU!

