COMPUTATIONAL INTELLIGENCE IN EMPIRICAL DATA MODELING FOR SOFTWARE DEVELOPMENT EFFORT ESTIMATION

Doctoral Thesis Defense

Tirimula Rao Benala

(13022P0508)

Date: 01.07.2020

Supervisor: **Prof. Rajib Mall, IIT KGP**

External Examiner: **Prof. Durga Prasad Mahapatra, NIT Rourkela**

Convener: **Prof. D. Haritha, HOD, CSE**

Department of Computer Science and Engineering,

Jawaharlal Nehru Technological University,

Kakinada – 533 003, Andhra Pradesh, India.

Overview

- Introduction
- Motivation
- Objective (s)
- State-of-Art
- Design and Experimental Details
- Analysis of Results
- Conclusions
- Future Work
- References

INTRODUCTION

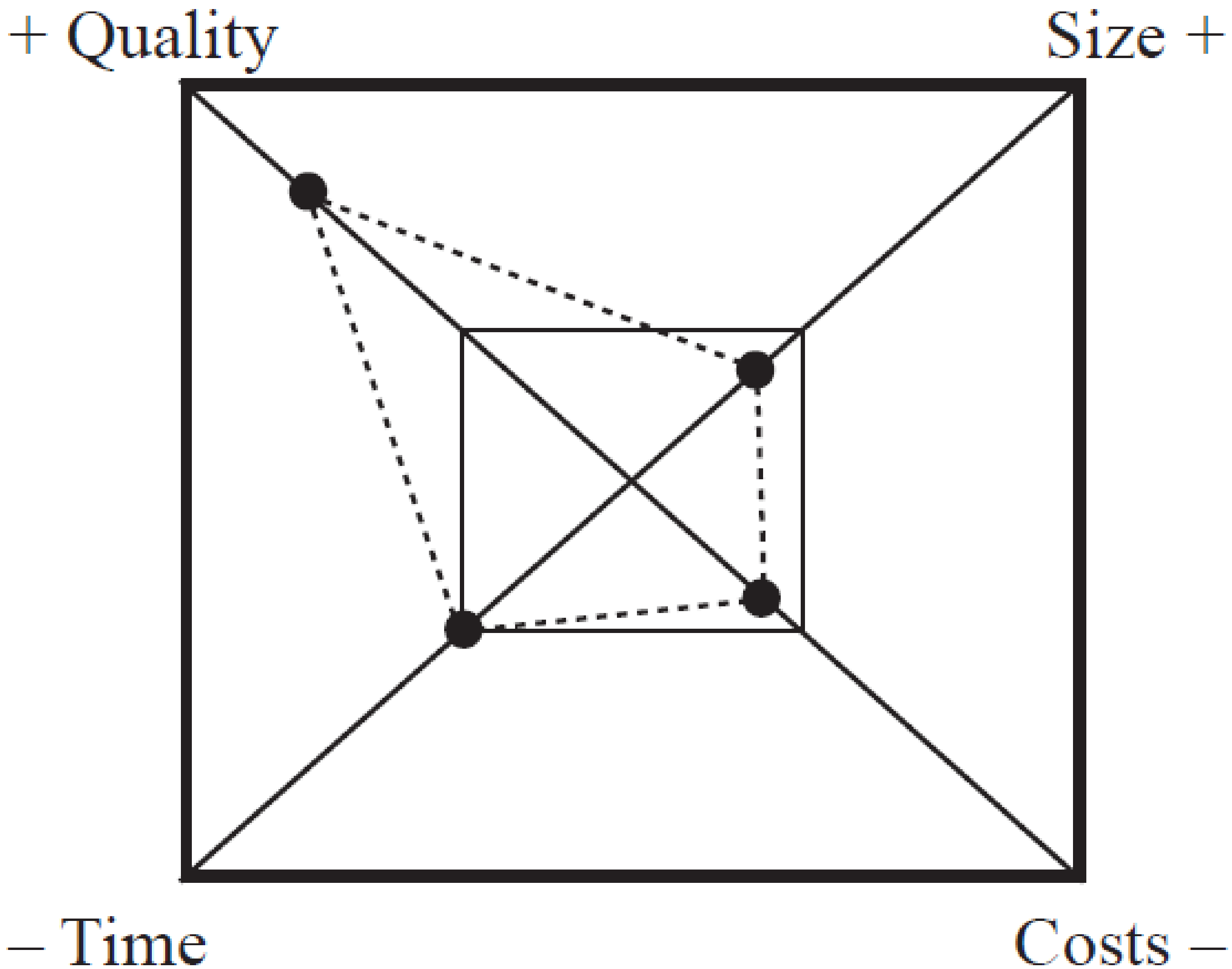
- Software development effort estimation is defined as “the process of predicting the effort required to develop a software system”(Wen et al., 2012).

- Software Cost Estimation has been identified as one of the three great challenges in computer science. (Fred Brooks , 2003).

- Software is Ubiquitous.

- The quality of software generally influences numerous variables and aspects of life with its wide range of diverse applications.

INTRODUCTION



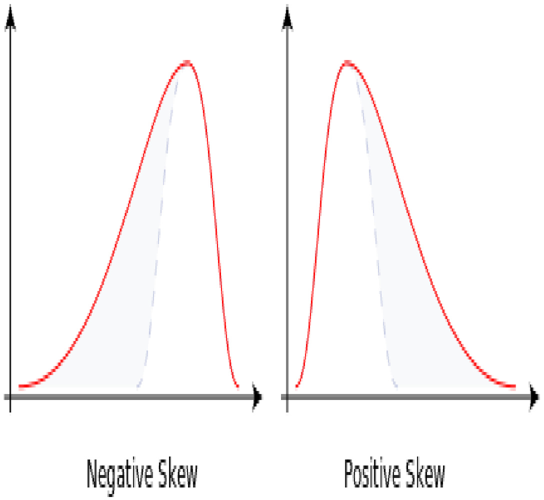
Sneed’s devil square

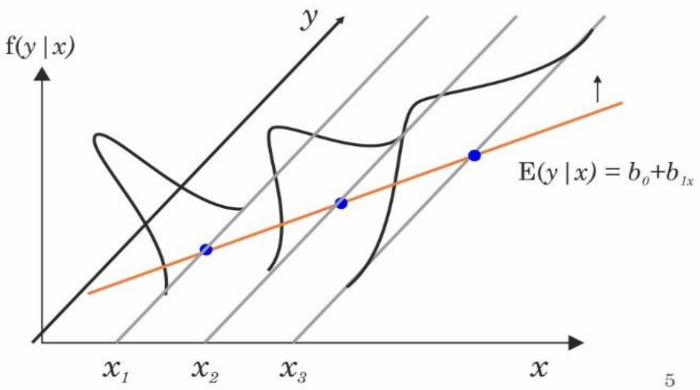
RESEARCH MOTIVATION

- Companies operate in non-stationary environment.
- New Employees could be hired or lost.
- New types of software projects Could be accepted.
- New Programming languages could be introduced.
- Training could be required.

RESEARCH MOTIVATION

- ***Low prediction accuracy***
  - Due to non-normal characteristics of datasets.
  - Skewness,heteroscedasticity(i.e.,each subpopulation in the dataset exhibited non-constant variance), and outliers





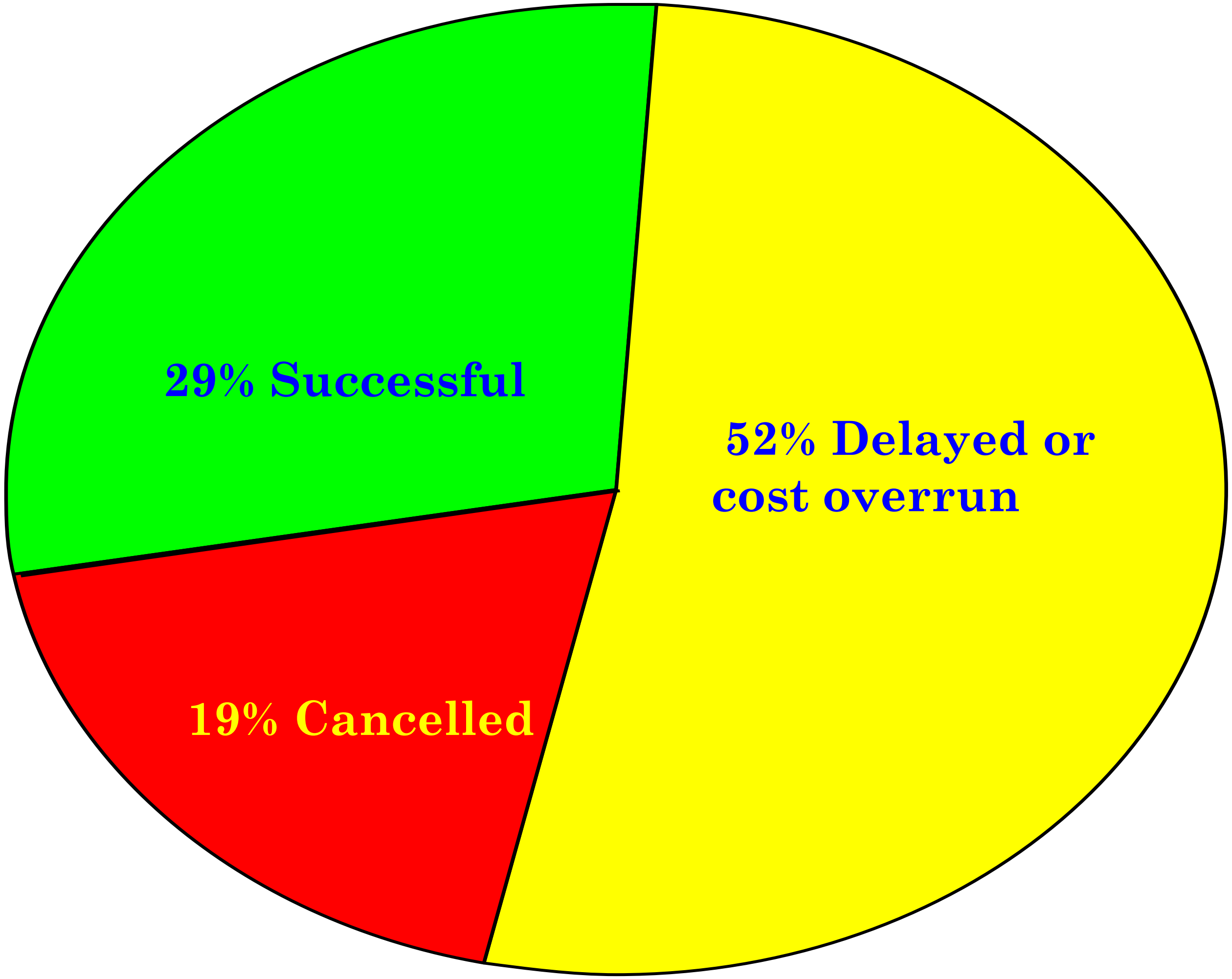
Heteroscedasticity

RESEARCH MOTIVATION

- parametric (algorithmic) models were unsuitable for the effort prediction because the datasets did not exhibit normal distribution.
- The project attributes exhibited multicollinearity, that is, two or more attributes that were highly correlated and influenced the accuracy of the effort prediction.

RESEARCH MOTIVATION

The Standish Group Chaos Report



Research motivation

- The typical characteristics of datasets and limitation of algorithmic models leads to development of non-algorithmic models.
- These are by and large subsets of computational intelligence techniques.
- The computational intelligence techniques are basically black-box optimization techniques.
- They can adapt to the heterogeneous sub-systems of large software system.

RESEARCH MOTIVATION

- Thus, software development effort estimation based on computational intelligence techniques can predict effort more accurately as compare to the conventional models.

- The focus of our research is on application of computational intelligence techniques to enhance the performance accuracy software development effort estimation.

RESEARCH OBJECTIVE

- The overall objective of this research is to investigate the evidence for improvements in SDEEs' by using computational intelligence models.

(1) To enhance the performance of analogy-based software development effort estimation (ABE) using differential evolution by:

i. Reducing the degree of influence of irrelevant attributes,

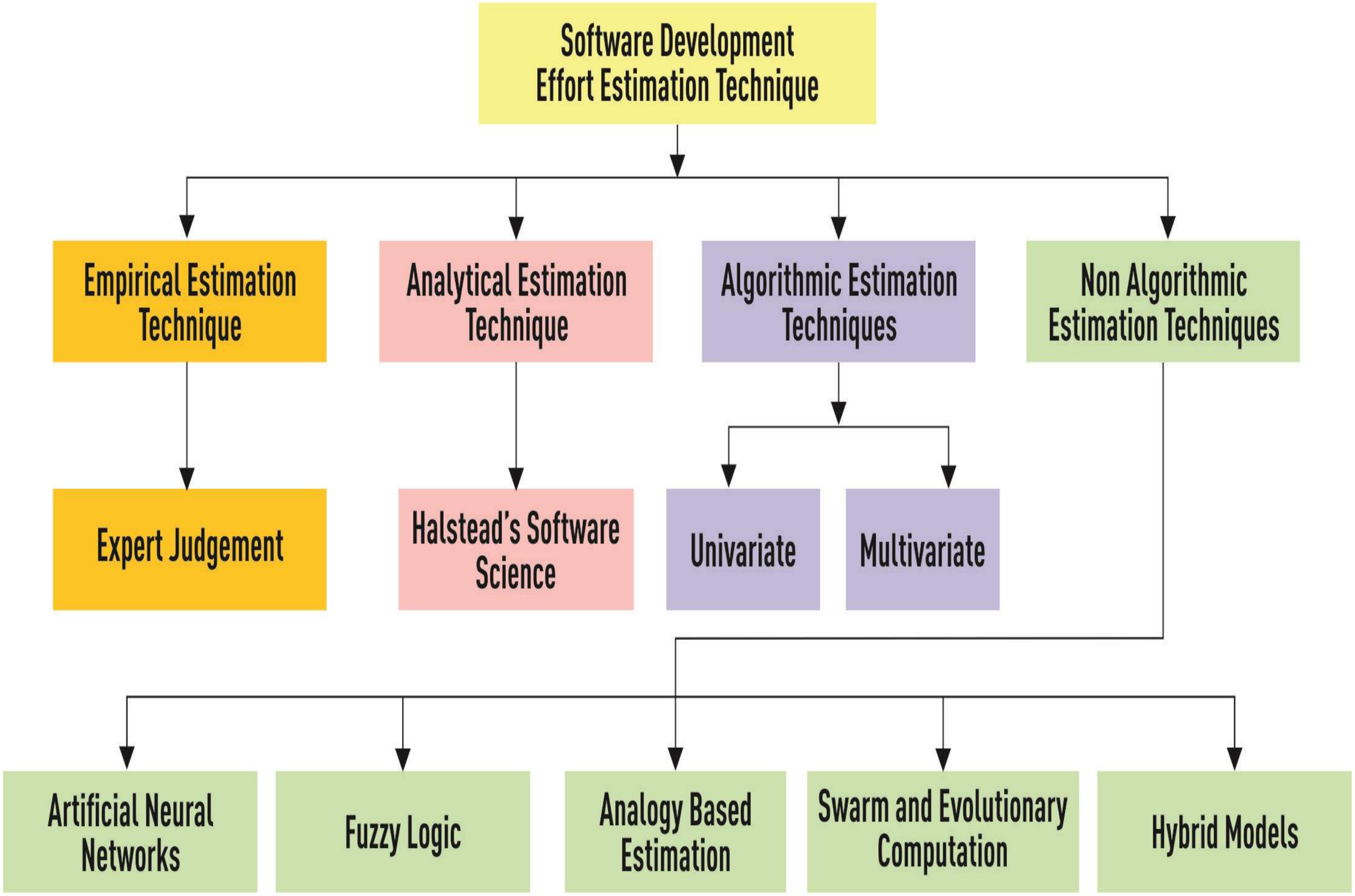
ii. Considering unequal influence of each attribute to determine the estimate.

Research objective

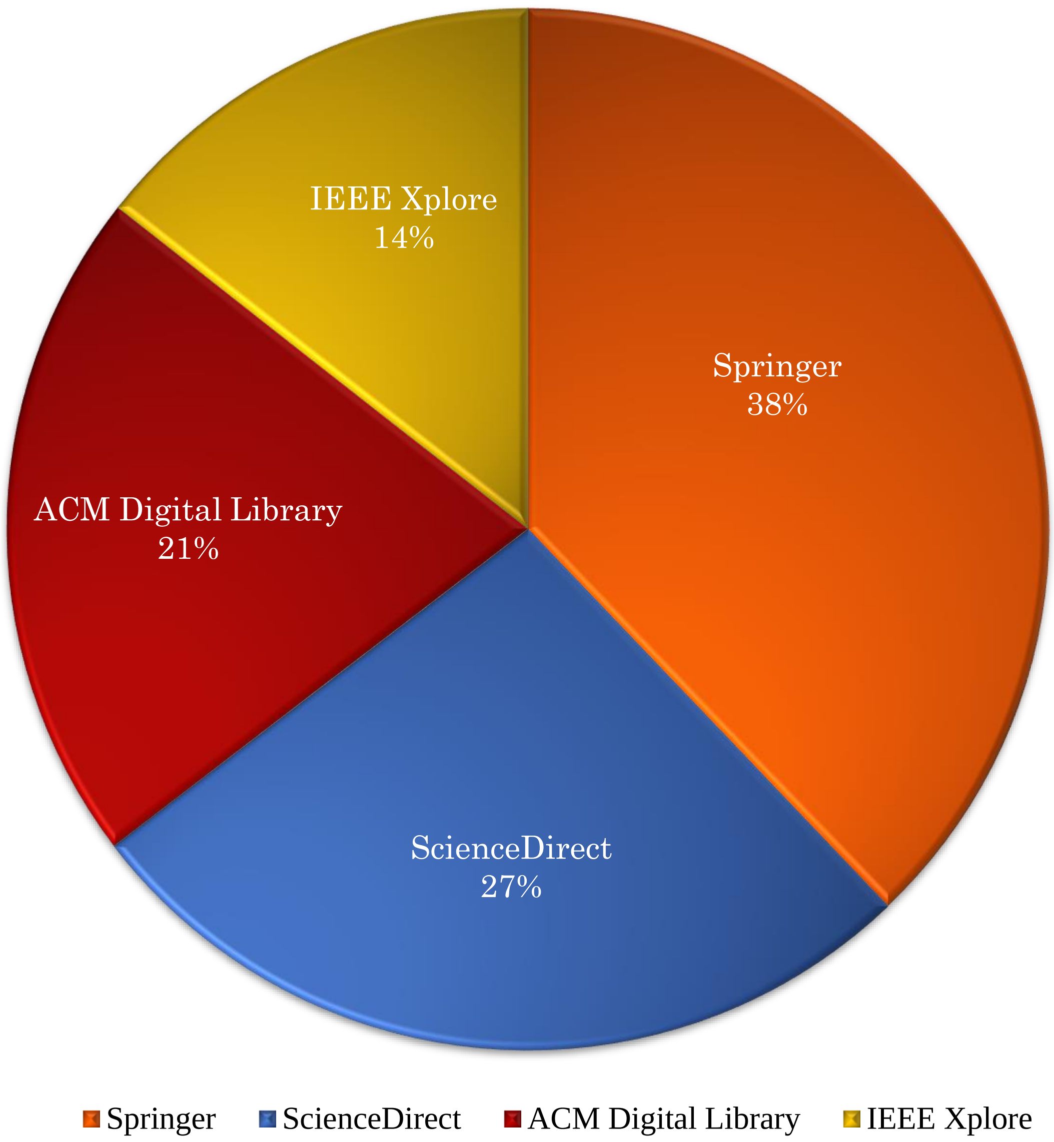
(2) To Design ensemble techniques by combining the strengths of individual machine learning techniques and improve the performance.

(3) To harness the power of random vector functional link neural net (RVFL) to learn nonlinear relationship between input-output of the datasets, and hence improve the performance accuracy of SDEE technique.

STATE-OF-ART

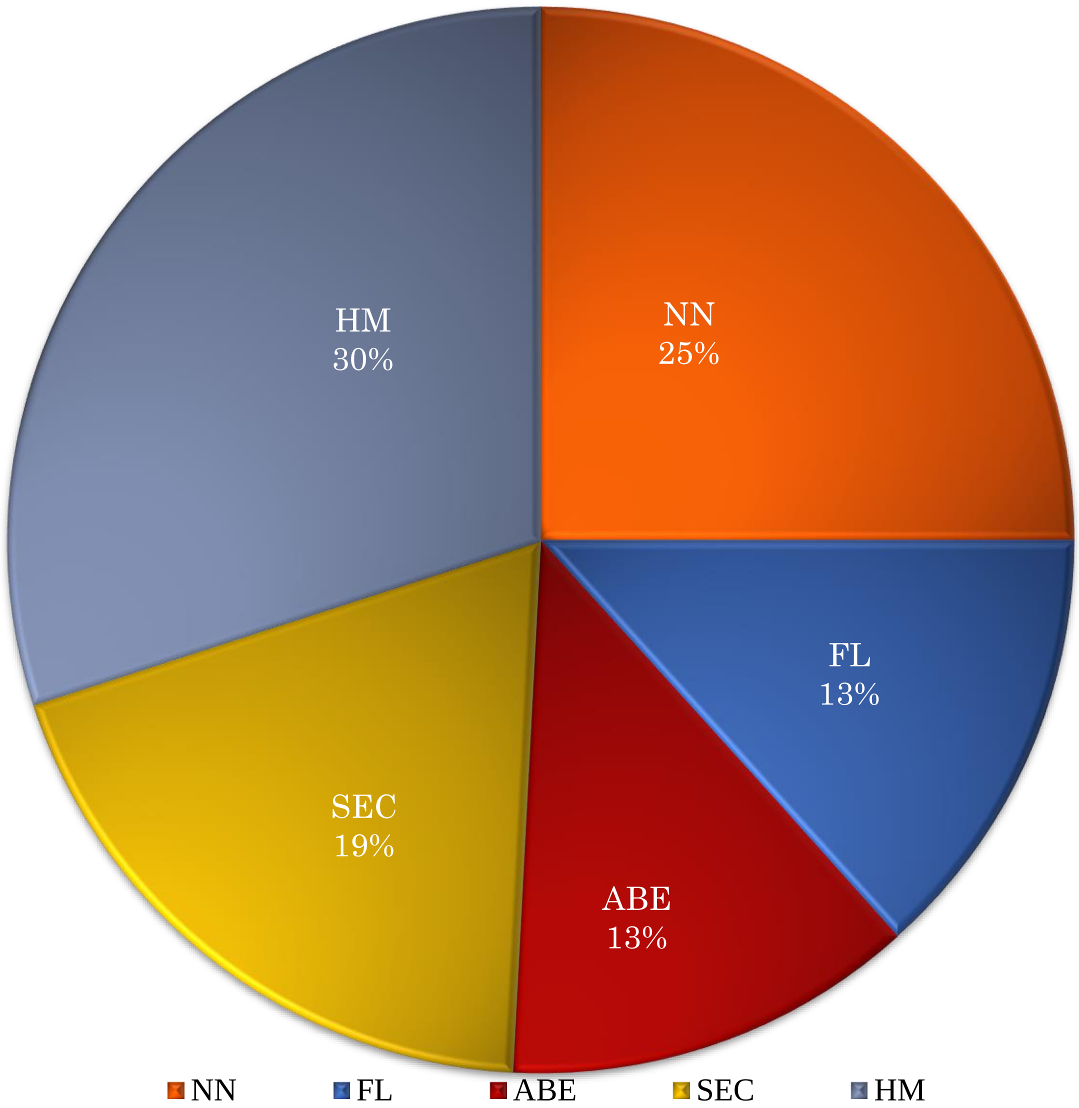

Software Development Effort Estimation Technique
Empirical Estimation Technique
Analytical Estimation Technique
Algorithmic Estimation Techniques
Non Algorithmic Estimation Techniques
Expert Judgement
Halstead's Software Science
Univariate
Multivariate
Artificial Neural Networks
Fuzzy Logic
Analogy Based Estimation
Swarm and Evolutionary Computation
Hybrid Models

STATE-OF-ART



IEEE Xplore
14%
Springer
38%
ACM Digital Library
21%
ScienceDirect
27%
Springer
ScienceDirect
ACM Digital Library
IEEE Xplore

STATE-OF-ART



Descriptive Statistics of Datasets

Dataset	Features	Number of Projects	Effort Data					
			Unit	Min	Max	Mean	Median	Skew
Desharnais	**12**	81	Hours	546	23,940	5046	3647	2.0
Nasa	**17**	93	Months	5	138.3	49.47	26.5	0.57
Cocomo	**17**	63	Months	6	11,400	686	98	4.4
China	**18**	499	Hours	26	54,620	3921	1829	3.92
Maxwell	**27**	62	Hours	583	63,694	8223.2	5189.5	3.26
Albrecht	**7**	24	Months	1	105	22	12	2.2

Function point analysis Category	**Datasets**	**Cost metrics**	
	Albrecht	Input, Output, Inquiry, File, FPAdj, RawFP, AdjFP, Effort	
	China	AdjFP, Input, Output, Enquiry, File, Interface, Added, Changed, Deleted, PDR_AFP, PDR_UFP, NPDR_AFP, NPDU_UFP, Resource, Dev.Type, Duration, N_effort, Effort	
	Maxwell	Number of different development languages used (Nlan), Customer participation (T01), Development environment Adequacy (T02), Staff availability (T03), Standards use (T04), Methods use(T05), Tools use (T06), Software's logical complexity (T07), Requirements volatility (T09), Quality requirements (T10), Efficiency requirements (T11), Installation requirements (T12), Staff analysis skills (T12), Staff application knowledge (T13), Staff tool skills (T14), Staff team skills (T15), Duration, time, Application size(number of function points).	
	Desharnais	TeamExp, ManagerExp, YearEnd, Length, Transactions, Entities, RawFP, FPAdj, AdjFP, Language, Effort	
COCOMO II category	Datasets	Cost metrics	
	Nasa93 Coc81	Product aspect	Required software reliability (RELY), Data base size (DATA), Product complexity (CPLX), Require reusability (RUSE), Documentation match to life-cycle needs (DOCU)
		Platform aspect	Execution time constraint (Time), Platform volatility (PVOL)
		Personnel aspect	Analyst capability (ACAP), Programmer capability (PCAP), Application experience (AEXP), Platform Experience (PEXP), Language and tool experience (LTEX), Personnel continuity (PCON)
		Project aspect	Use of software tools (TOOL), Multisite development (SITE), Required development schedule (SCED)

DABE: DIFFERENTIAL EVOLUTION IN ANALOGY-BASED SOFTWARE DEVELOPMENT EFFORT ESTIMATION

Benala, T. R., & Mall, R. (2018). DABE: Differential evolution in analogy-based software development effort estimation. Swarm and Evolutionary Computation, 38, 158-172.

INTRODUCTION TO ANALOGY BASED SOFTWARE EFFORT ESTIMATION

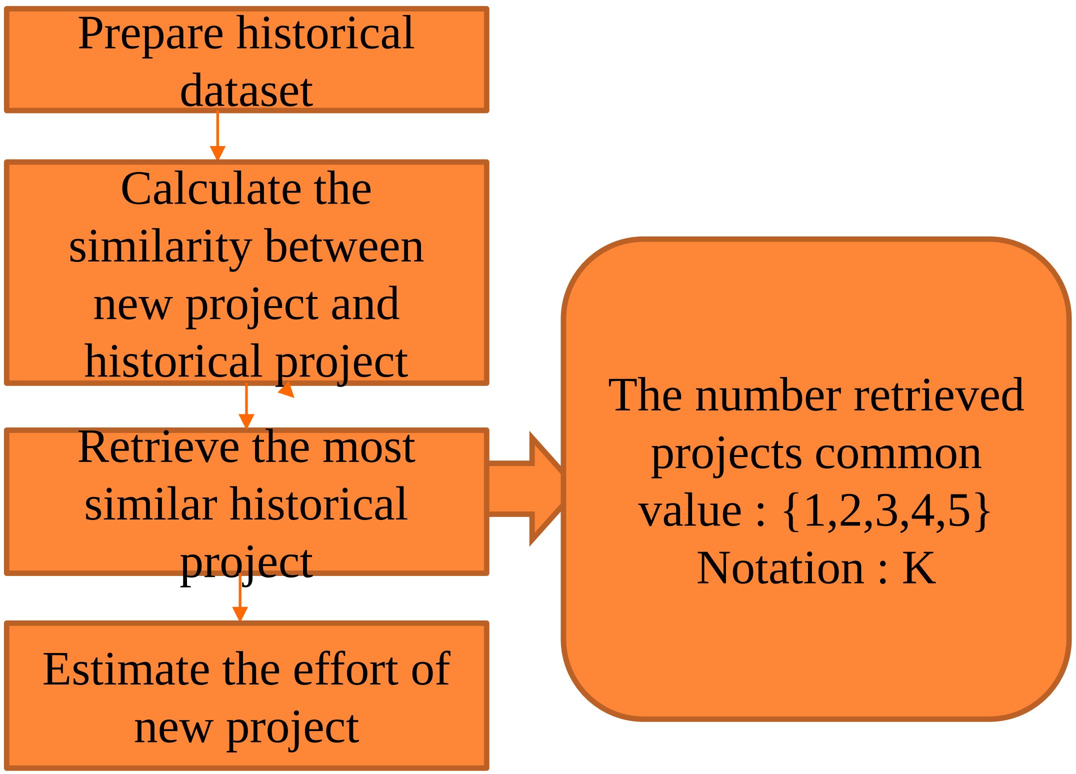
- Principle
  - Estimate effort based on the effort of similar projects from dataset.
  - Process

1. Prepare historical dataset
2. Calculate the similarity between new project and historical project
3. Retrieve the most similar historical project
4. Estimate the effort of new project

- Weighted Euclidean Similarity
- $\mathrm{Sim}(p, p') =$
  $$1/\left[\sqrt{\sum_{i=1}^{n} w_i \mathrm{Dis}\left(f_i, f_i'\right) + \delta}\right]$$
- $\delta = 0.0001$
- $\mathrm{Dis}\left(f_i, f_i'\right) =$
  $$\begin{cases} (f_i - f_i')^2 \text{ if } f_i \text{ and } f_i' \text{ are numerical or ordinal} \\ 1 \text{ if } f_i \text{ and } f_i' \text{ are nominal and } f_i = f_i' \\ 0 \text{ if } f_i \text{ and } f_i' \text{ are nominal and } f_i \neq f_i' \end{cases}$$

INTRODUCTION TO ANALOGY BASED SOFTWARE EFFORT ESTIMATION

- Principle
  - Estimate effort based on the effort of similar projects from dataset.
  - Process



INTRODUCTION TO ANALOGY BASED SOFTWARE EFFORT ESTIMATION

- Principle
  - Estimate effort based on the effort of similar projects from dataset
  - Process

Prepare historical dataset

Calculate the similarity between new project and historical project

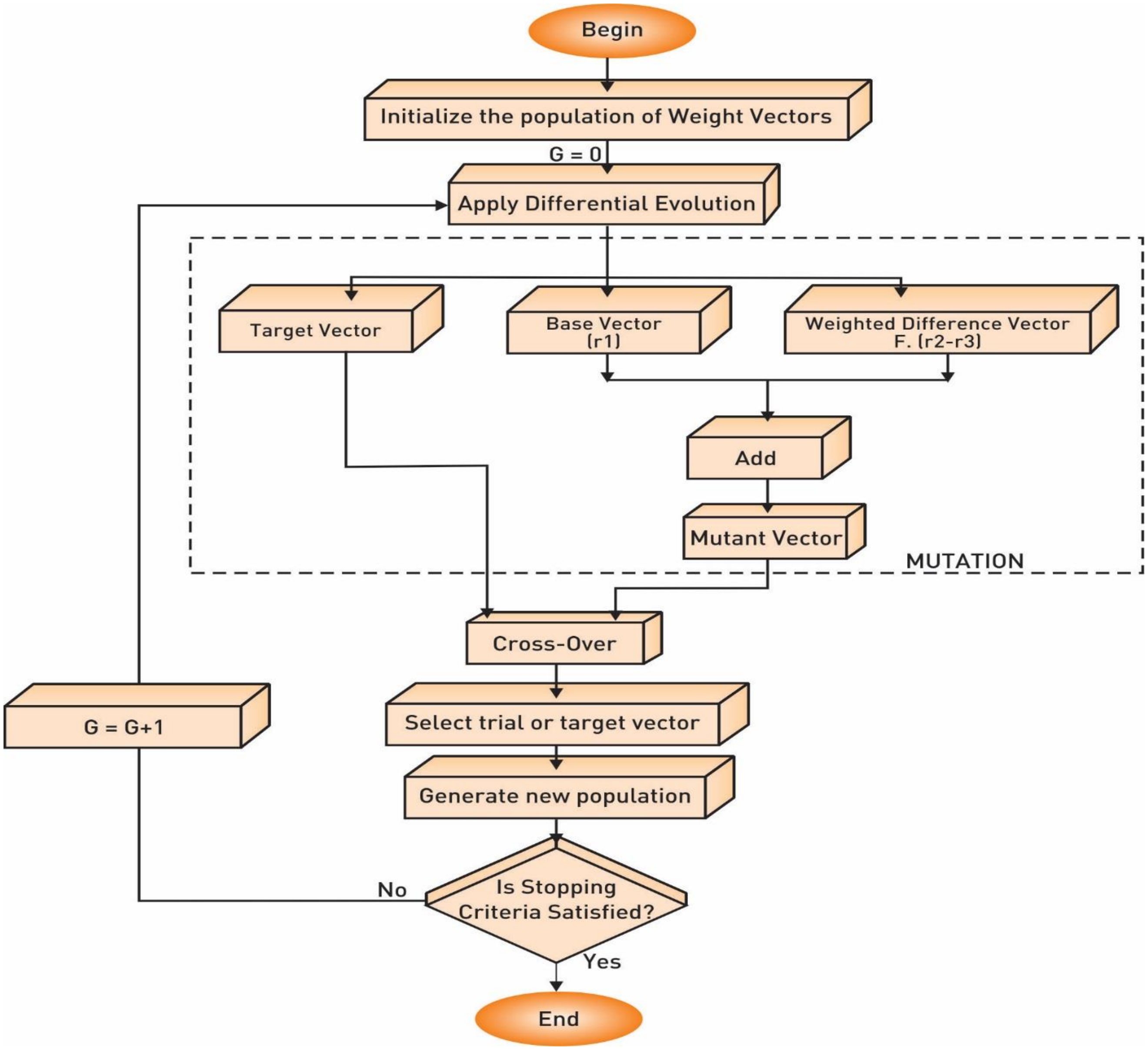
Retrieve the most similar historical project

Estimate the effort of new project

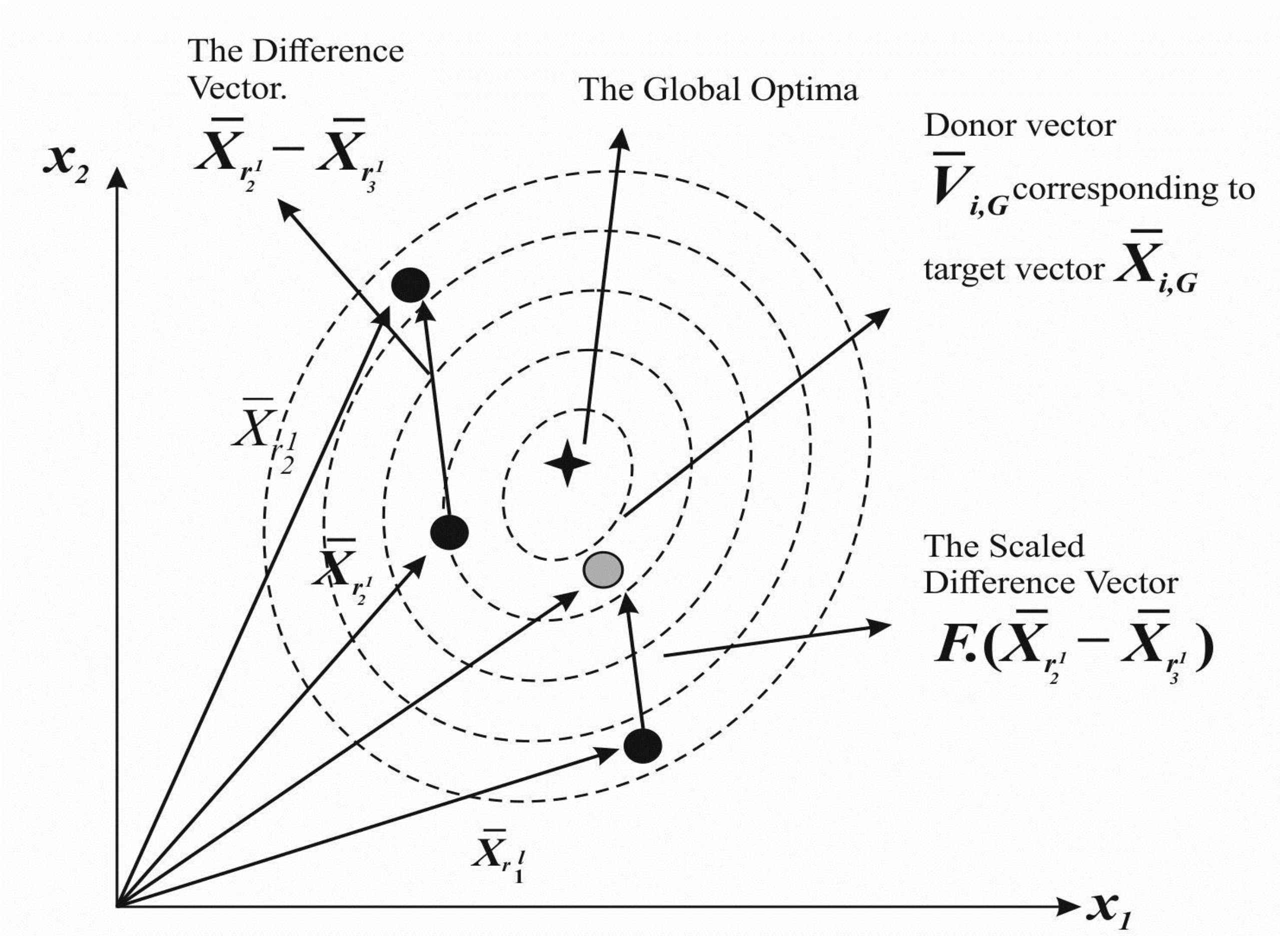
Solution function

- K=1
- Effort of the selected project
- K=2
- Mean of the effort of the selected project
- Median of the effort of the selected project
- K>=3
- Mean of the effort of the selected project
- Median of the effort of the selected project
- Inverse distance weighted mean of the selected project
- $Effort_p = \sum_{i=1}^{K} \frac{\mathrm{Sim}(\mathrm{p},p_i)}{\sum_{\mathrm{j=1}}^{\mathrm{k}} \mathrm{Sim}(\mathrm{p},p_j)} \quad Effort_{p_i}$

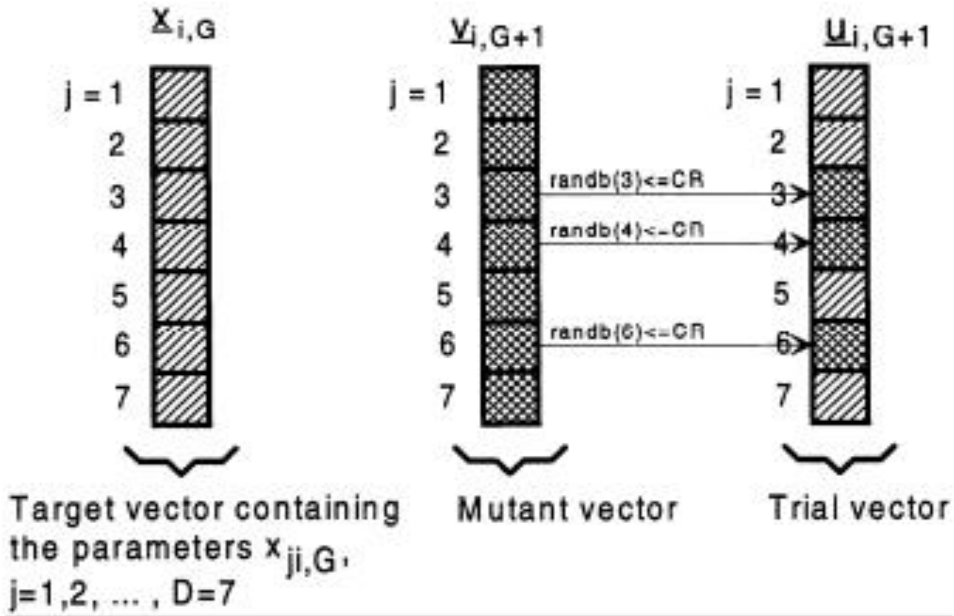
DIFFERENTIAL EVOLUTION



SCEMATIC BLOCK DIAGRAM OF FORMATION DONOR VECTOR



SCHEMATIC BLOCK DIAGRAM OF CROSSOVER OPERATION



$$u_{i,j}(t) = V_{i,j}(t) \quad \text{if } rand_{i,j}(0, 1) <= CR \text{ or } k_i = j.$$

$$= X_{i,j}(t) \quad \text{otherwise}$$

Five most frequently used DE mutation schemes

"DE/rand/1": $\vec{V}_i(t) = \vec{X}_{r_1^i}(t) + F \cdot (\vec{X}_{r_2^i}(t) - \vec{X}_{r_3^i}(t)).$

"DE/best/1": $\vec{V}_i(t) = \vec{X}_{best}(t) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)).$

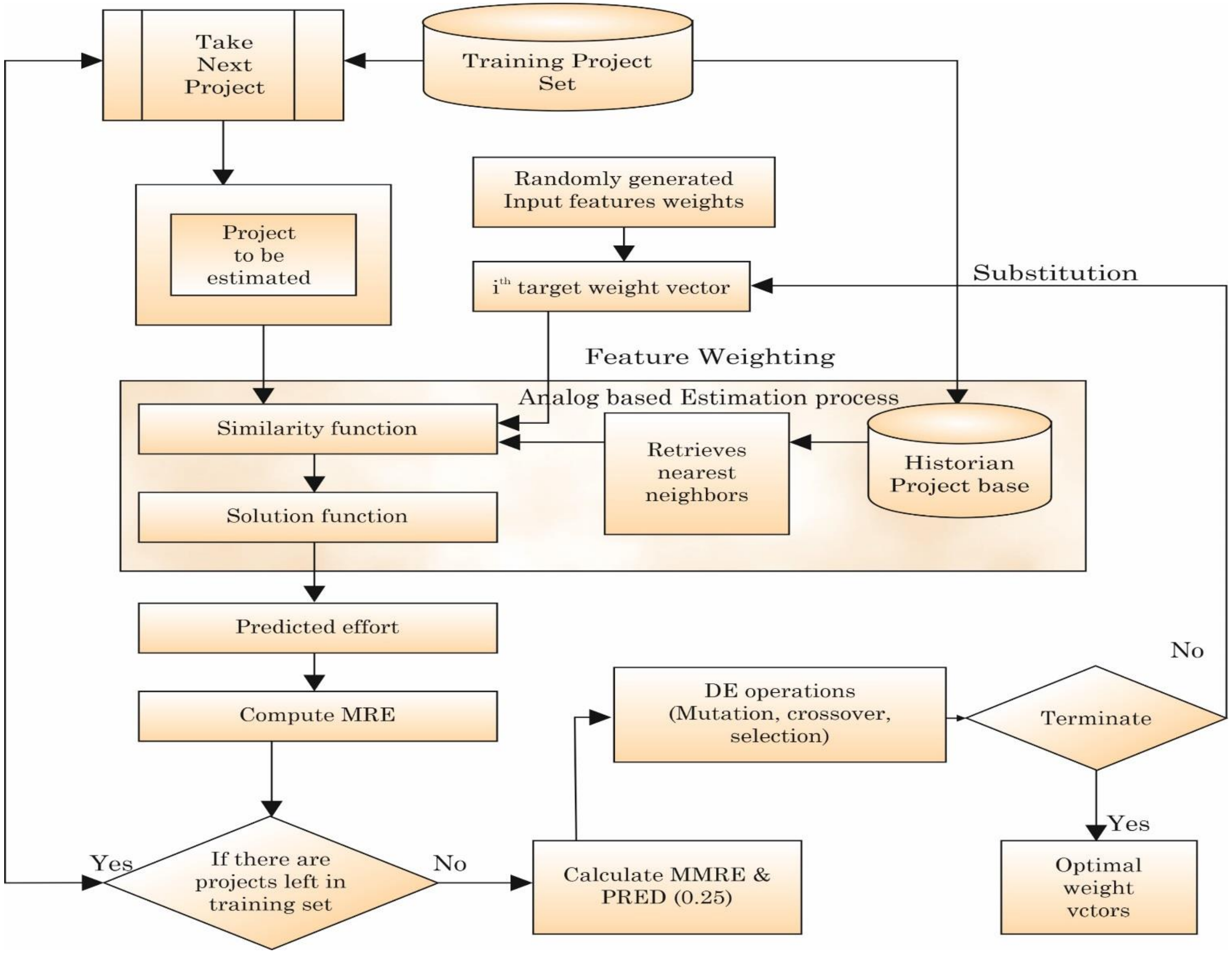
"DE/target-to-best/1": $\vec{V}_i(t) = \vec{X}_i(t) + F.(\vec{X}_{best}(t) - \vec{X}_i(t)) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)),$

"DE/best/2": $\vec{V}_i(t) = \vec{X}_{best}(t) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)) + F.(\vec{X}_{r_3^i}(t) - \vec{X}_{r_4^i}(t)).$

"DE/rand/2": $\vec{V}_i(t) = \vec{X}_{r_1^i}(t) + F_1.(\vec{X}_{r_2^i}(t) - \vec{X}_{r_3^i}(t)) + F_2.(\vec{X}_{r_4^i}(t) - \vec{X}_{r_5^i}(t)).$

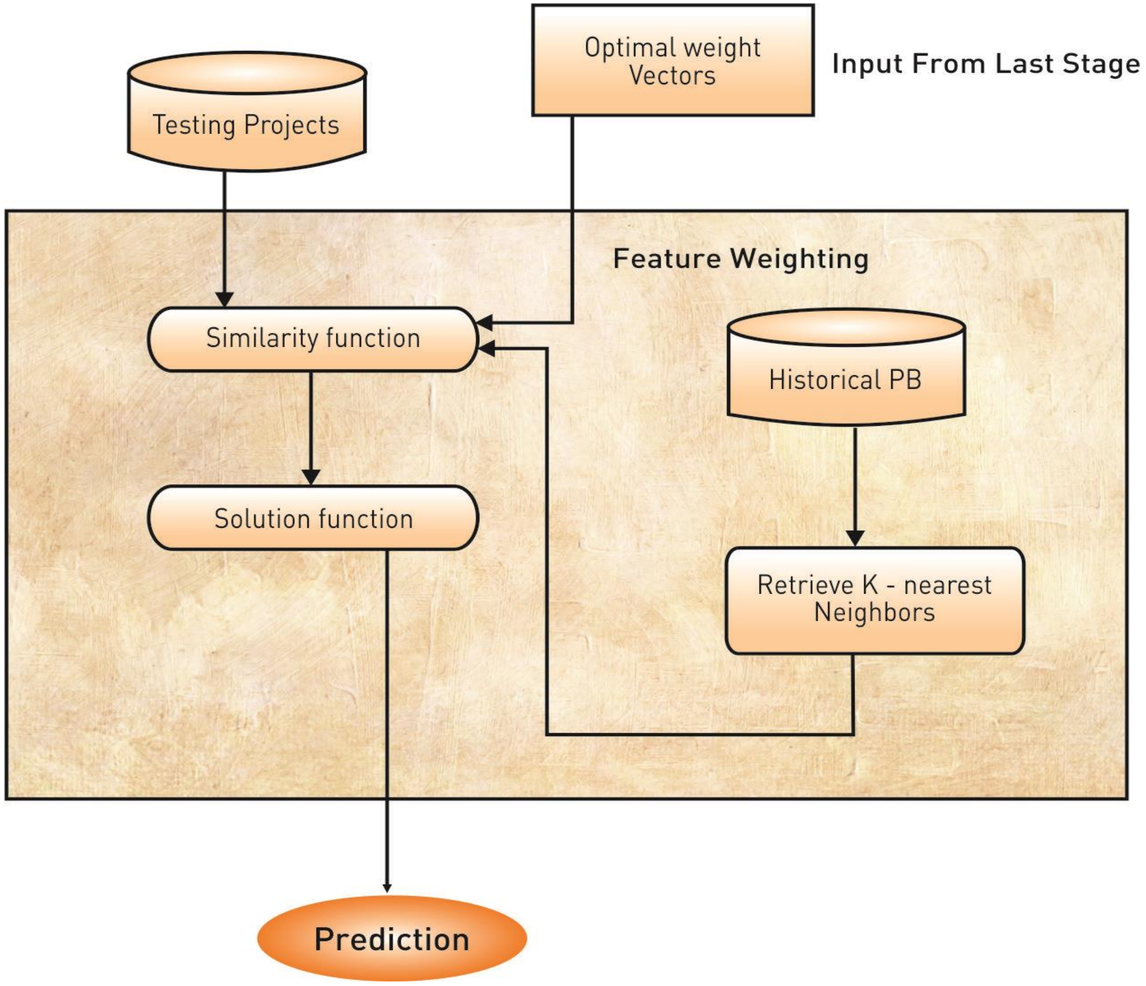
The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exp: exponential; bin: binomial)

DABE



The system architecture of DABE training stage.

DABE



The System architecture of DABE testing stage.

EXPERIMENT SETUP

- **Goal of experiment**
  - compare prediction accuracy of DABE model with PSO-ABE, GA-ABE, SaDE-ABE, and JADE-ABE.
  - **Objective Function**
  - The reciprocal of [MMRE-PRED(0.25)+ϵ] is employed as the objective function
  - **DABE Paramters**
  - Scaling factor F: 0.8
  - Crossover rate CR: 0.5
  - learning strategies “DE/rand/1″, “DE/Best/1”, “DE/target-to-best/1”, “DE/Best/2”, “DE/rand/2”
  - **SaDE ABE Paramters (Qin et al. 2009)**
  - Scaling factor F: N(0.5, 0.3)
  - Crossover rate CR: N (0.5, 0.1)
  - learning strategies “DE/rand/1″ and “DE/
  - current-to-best/1,”

EXPERIMENT SETUP

- **Parameter Setting**
  - JADE-ABE (Zhang et al. 2009)
  - Scaling factor F: Cuachy(0.5,0.1)
  - Crossover rate CR: N(0.5,0.1)
  - learning strategies "DE/current-to-pbest/1" with an optional archive and updated control parameters in an adaptive manner
  - p $\in [5\%, 20\%]$ i.e. life span of F and CR values range from 5 to 10 generations and we consider 5% to 20% high quality solutions in the mutation.

EXPERIMENT SETUP

Technique	Parameters
MLP	Weights: [0,1]/ 4 hidden nodes (trial & Error) Learning rate={0.1,0.2,0.3,0.4,0.5} Momentum={0.1,0.2,0.3,0.4,0.5}
FLANN	Weights: [0,1]/ Chebyshev Polynomial
RBF	#Clusters = {2,3,4,5,6} Variance for the clusters : 0.25-0.75 with increment of 0.05.

PERFORMANCE METRICS

The performance indicator prevalently used for measuring the efficacy of the software prediction models are defined as follows:

$$\mathrm{AE_i} = |\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}|$$

$$\mathrm{MRE_i} = \frac{\mathrm{AE_i}}{\mathrm{y_i}}$$

$$\mathrm{MMRE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{MRE_i}$$

$$\mathrm{PRED(x)} = \frac{100}{\mathrm{N}} \times \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{D_i}$$

Performance Metrics

$$D_i = \begin{cases} 1 \text{ if} MMRE < \frac{x}{100} \\ 0 \text{ otherwise} \end{cases}$$

When x=25, the PRED metric is defined as PRED (0.25)

$$MAR = \frac{\sum_{i=1}^{N} AE_i}{N}$$

$$SA = 1 - \frac{MAR}{\overline{MAR_{P_0}}}$$

$$\Delta = \frac{MAR - \overline{MAR_{P_0}}}{s_{P_0}}$$

MAR: Mean Absolute Residual; SA: Standardized Accuracy; Δ- Cohen's Delta

DABE VARIANTS

- DABE-1 : “DE/rand/1/bin”
- DABE-2 : “DE/best/1/bin”
- DABE-3 : “DE/rand/2/bin”
- DABE-4 : “DE/best/2/bin”
- DABE-5 : DE/target-to-best/1/bin”

Results of effect size for Desharnais dataset

Model	Min. Δ	Max. Δ	Avg. Δ	Std. Δ	# Small	# Medium	#Great/ Immense/High/ Considerable/ Long	# Medium +Great/ Immense/ High/ Considerable/ Long
DABE1	0.075	0.858	0.494	0.229	2	10	12	22
DABE2	0.091	0.99	0.439	0.227	4	12	8	20
DABE3	0.11	1.126	0.529	0.272	4	11	10	21
DABE4	0.221	1.027	0.459	0.197	3	13	8	21
DABE5	0.11	0.919	0.429	0.224	6	9	9	18

DABE-3 outperformed other variants with a mean (standard deviation) of 0.529 (0.272). In 21 cases, Δ were ≥ 0.5.

Results of effect size for Maxwell dataset.

Model	Min. Δ	Max. Δ	Avg. Δ	Std. Δ	# Small	# Medium	#Great/ Immense/High/ Considerable/ Long	# Medium +Great/ Immense/ High/ Considerable/ Long
DABE1	0.133	1.691	0.538	0.351	3	9	12	21
DABE2	0.118	1.435	0.574	0.325	2	7	15	22
DABE3	0.210	2.048	0.614	0.370	2	7	14	21
DABE4	0.151	2.107	0.586	0.693	4	9	12	21
DABE5	0.109	1.758	0.612	0.374	2	9	13	22

DABE-3 outperformed other variants with a **mean (standard deviation**) of **0.614 (0.37).** The Δ was ≥ 0.5 in 21 cases. Thus DABE-3 was selected as the most suitable variant for comparison with other SDEE techniques.

Results of DABE-3 for Maxwell dataset.

Similarity	K	Solution	Training			Testing		
			MMRE	PRED	MdMRE	MMRE	PRED	MdMRE
	K=1	CA	0.003	0.516	0.016	0.010	1.000	0.010
	K=2	Mean	0.072	0.025	0.072	**0.014**	**0.074**	**0.014**
		IWM	0.067	0.049	0.067	0.100	**0.074**	0.100
	K=3	Mean	0.056	0.049	0.024	**0.085**	0.099	**0.028**
		IWM	0.067	0.074	0.067	0.100	**0.111**	0.100
		Median	0.049	0.049	0.050	0.086	0.099	**0.028**
Euclidean	K=4	Mean	0.052	0.074	0.025	0.274	**0.111**	**0.100**
		IWM	0.067	0.099	0.067	0.100	0.148	**0.100**
		Median	0.048	0.074	0.017	**0.208**	**0.111**	0.106
	K=5	Mean	0.051	0.099	0.026	0.313	0.123	0.172
		IWM	0.067	0.123	0.067	0.100	**0.185**	0.100
		Median	0.044	0.099	0.038	**0.203**	0.136	**0.078**
	K=1	CA	0.024	0.554	0.016	0.009	1.000	0.009
	K=2	Mean	0.072	0.025	0.072	**0.014**	**0.074**	**0.014**
		IWM	0.067	0.049	0.067	0.100	**0.074**	0.100
	K=3	Mean	0.056	0.049	0.024	0.288	0.099	**0.028**
		IWM	0.067	0.074	0.067	**0.100**	**0.111**	0.100
		Median	0.049	0.049	0.050	0.307	0.099	**0.028**
Manhattan	K=4	Mean	0.052	0.074	0.025	0.303	0.111	0.079
		IWM	0.067	0.099	0.067	**0.100**	**0.148**	0.100
		Median	0.048	0.074	0.017	0.276	0.123	**0.071**
	K=5	Mean	0.051	0.099	0.026	0.336	0.123	0.131
		IWM	0.067	0.123	0.067	**0.100**	**0.185**	0.100
		Median	0.044	0.099	0.038	0.257	0.148	**0.078**

Results of DABE-3 for Desharnais dataset.

Similarity	K	Solution	Training			Testing		
			MMRE	PRED	MdMRE	MMRE	PRED	MdMRE
	K=1	CA	0.017	0.667	0.010	0.048	0.988	0.031
	K=2	Mean	0.001	0.121	0.001	**0.020**	**0.182**	**0.020**
		IWM	0.067	0.121	0.067	0.100	**0.182**	0.100
	K=3	Mean	0.048	0.121	0.003	**0.023**	**0.273**	**0.021**
		IWM	0.067	0.182	0.067	0.100	**0.273**	0.100
		Median	0.064	0.121	0.003	0.025	**0.273**	0.026
Euclidean	K=4	Mean	0.038	0.182	0.002	0.046	**0.364**	**0.026**
		IWM	0.067	0.242	0.067	0.100	**0.364**	0.100
		Median	0.050	0.182	0.006	**0.037**	**0.364**	0.028
	K=5	Mean	0.060	0.242	0.003	0.093	0.424	**0.025**
		IWM	0.067	0.303	0.067	0.100	**0.455**	0.100
		Median	0.075	0.242	0.009	**0.065**	0.424	0.027
	K=1	CA	0.015	0.667	0.010	0.036	0.988	0.025
	K=2	Mean	0.001	0.121	0.001	**0.033**	**0.182**	0.033
		IWM	0.067	0.121	0.067	0.100	**0.182**	0.100
	K=3	Mean	0.048	0.121	0.003	0.042	**0.273**	**0.015**
		IWM	0.067	0.182	0.067	0.100	**0.273**	0.100
		Median	0.064	0.121	0.003	**0.035**	**0.273**	0.028
Manhattan	K=4	Mean	0.038	0.182	0.002	0.043	**0.364**	0.031
		IWM	0.067	0.242	0.067	0.100	**0.364**	0.100
		Median	0.050	0.182	0.006	**0.035**	**0.364**	**0.022**
	K=5	Mean	0.060	0.242	0.003	0.090	0.424	0.037
		IWM	0.067	0.303	0.067	0.100	**0.455**	0.100
		Median	0.075	0.242	0.009	**0.063**	0.424	**0.025**

- According to the MMRE and MdMRE performance measures, the most suitable ABE configuration for both Maxwell and Desharnais datasets was {ES, k = 3, mean solution function}.

- However, the PRED performance indicator test performance indicated that the most favorable ABE configuration for both data sets was {ES/MS measure, k = 5, IWM solution function}.

Results of SA for Desharnais dataset

Similarity	K	Training				Testing			
		Min. SA	Max. SA	Avg. SA	Std. SA	Min. SA	Max. SA	Avg. SA	Std. SA
Euclidean	3	10.602	90.126	36.584	26.206	28.165	**89.886**	53.74	19.18
	4	10.03	88.438	38.077	22.325	25.39	83.669	53.99	17.887
	5	11.102	30.681	21.014	5.895	22.859	88.623	42.171	15.934
Manhattan	3	11.539	63.363	31.873	16.996	22.398	79.268	51.367	17.755
	4	11.803	68.415	30.112	17.092	16.102	80.788	38.788	11.165
	5	10.928	31.294	20.94	6.826	15.164	65.388	37.503	11.165

Results of SA for Maxwell dataset

Similarity	K	Training				Testing			
		Min. SA	Max. SA	Avg. SA	Std. SA	Min. SA	Max. SA	Avg. SA	Std. SA
Euclidean	3	12.126	89.657	50.484	30.418	27.38	**96.183**	59.954	14.693
	4	13.301	95.683	35.395	27.094	23.07	53.403	39.464	8.356
	5	15.175	27.556	22.055	3.826	32.329	85.308	50.43	12.522
Manhattan	3	11.77	68.118	39.293	14.523	21.219	88.736	57.88	18.927
	4	13.709	79.116	33.149	20.539	23.108	71.861	45.239	14.652
	5	13.284	63.447	26.114	10.045	26.837	75.489	50.177	12.898

Results of solution function for Maxwell Dataset For K=3

Solution function	Training				Testing			
	Min. SA	Max. SA	Avg. SA	Std. SA	Min. SA	Max. SA	Avg. SA	Std. SA
Mean	26.348	83.84	42.43	21.68	44.36	**96.18**	58.26	18.77
IWM	12.13	84.92	33.89	33.74	27.38	84.18	58.59	18.12
Median	38.52	89.66	75.13	19.24	47.57	66.11	57.01	6.01

Results of solution function for Desharnais dataset for k=3

Solution function	Training				Testing			
	Min. SA	Max. SA	Avg. SA	Std. SA	Min. SA	Max. SA	Avg. SA	Std. SA
Mean	27.5	39.9	30.54	4.46	30.96	**83.19**	**51.35**	22.58
IWM	10.03	88.44	33.79	30.05	25.39	83.66	50.6	19.02
Median	19.2	84.51	49.9	25.3	44.14	74.24	60.04	11.44

The optimal ABE Configuration is {ES, k=3, and Mean Solution Function}

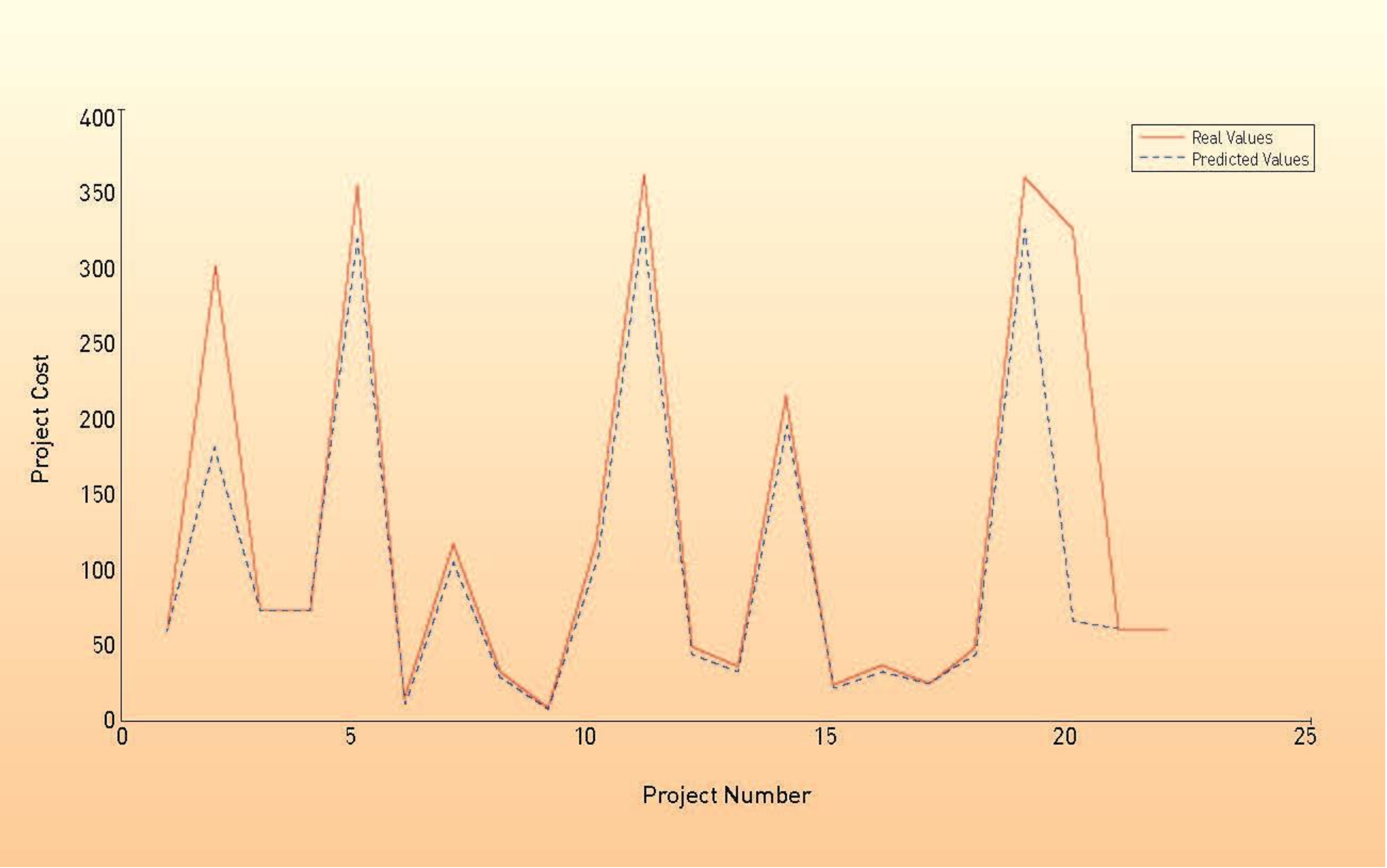
Results and comparisons for Albrecht dataset

Models	MMRE		PRED		MdMRE		SA		Δ	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
ABE	0.050	0.221	0.367	0.875	0.037	0.091	16.584	27.811	0.633	3.471
GA-ABE	0.095	0.018	0.167	0.250	0.001	0.018	89.452	83.233	0.291	1.734
DABE-3	0.095	0.020	0.167	0.375	0.001	0.019	95.523	**93.636**	0.266	**1.698**
PSO-ABE	0.081	0.018	0.333	0.250	0.039	0.018	76.824	83.180	0.286	1.714
SADE-ABE	0.015	0.027	0.376	0.291	0.018	0.031	92.451	90.453	1.481	1.251
JADE-ABE	0.023	0.016	0.421	0.262	0.291	0.018	91.953	88.753	0.451	0.985

Results and comparisons for China dataset

Models	MMRE		PRED		MdMRE		SA		Δ	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
ABE	0.131	0.116	0.259	0.868	0.116	0.041	16.035	12.493	1.313	0.042
GA-ABE	0.076	0.100	0.374	0.167	0.027	0.100	82.344	86.196	0.327	3.318
DABE-3	0.075	0.068	0.621	0.573	0.020	0.039	95.847	**96.509**	0.259	**1.818**
PSO-ABE	0.067	0.010	0.111	0.167	0.067	0.100	69.451	92.880	0.350	5.843
SADE-ABE	0.079	0.084	0.561	0.483	0.058	0.046	95.189	94.203	0.784	1.231
JADE-ABE	0.092	0.076	0.347	0.212	0.075	0.081	90.651	88.265	1.143	0.891

Prediction results for Nasa93 data set (test set) using DABE-3 algorithm: actual values (solid lines) and predicted values (dashed lines)



CONCLUSION

- We have proposed a feature weight optimization technique called DABE to improve the performance of ABE.

  - The mutation strategy “DE/rand/2/bin” was a version efficient for SDEE.

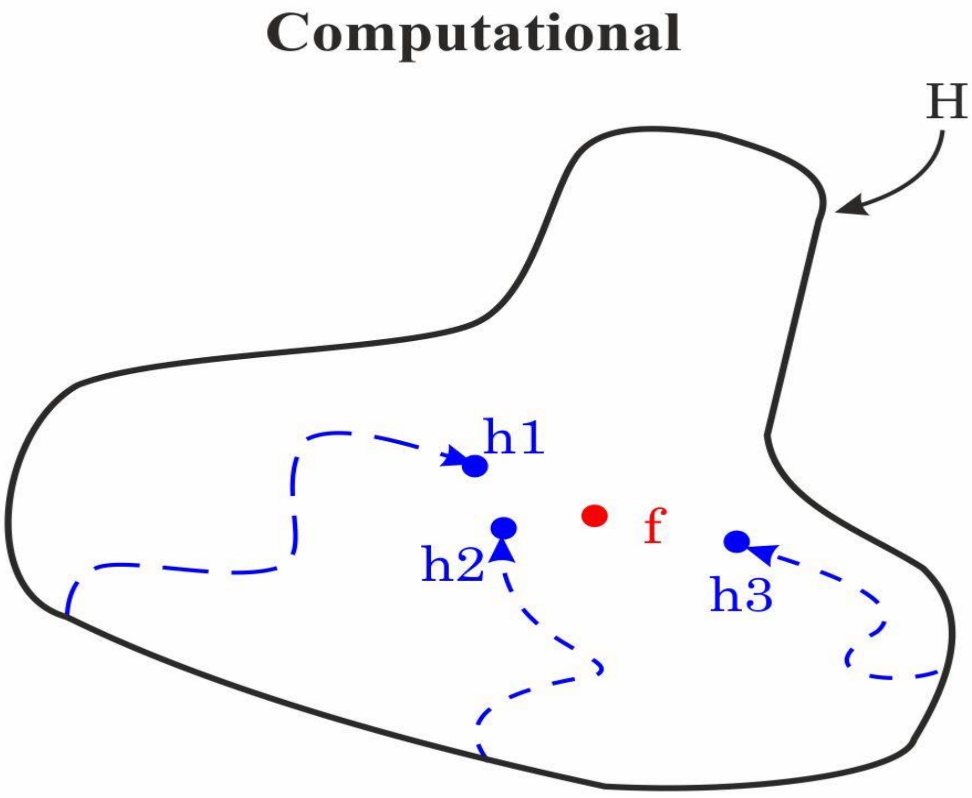
  - The performance measures SA, Δ is the unique global error indicators added in our study for assessing the prediction model performance.

SEET: SOFTWARE DEVELOPMENT EFFORT ESTIMATION USING ENSEMBLE TECHNIQUES

Benala, T. R., & Mall, R. (2018). SEET: Software Development Effort Estimation using Ensemble Techniques. ACM SIGSOFT Software Engineering Notes, 43(3)- July Issue.

ENSEMBLES

Fundamental reasons why an ensemble may work better than a single prediction model



SEET METHOD

- A foundation-centered approach was adopted to investigate the performance of the base techniques such as LSSVR, ELM, and MLP and their homogeneous ensembles.

- ELM is a state-of-art machine learning algorithm based on single-hidden layer feed-forward neural networks (SLFNs).

- The results of the ELM-based SDEE have not been reported in the relevant literature.

TECHNIQUES

LSSVR:

$$\text{Minimize } \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{i=1}^{l} E_i^2$$

$$\text{Subject to } y(x) = w^T \varphi(x) + c + E_{i,}\ i = 1,2, \dots, n$$

Where E_i is the error value of instance i and γ is the cost funciton

$$\text{The decision Function : } f(x) = \text{sign}\left(\sum_{s=1}^{N_s} \alpha_s y_s K(x, x_s) + c\right)$$

ELM:

$$\text{Minimize } \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{i=1}^{l} E_i^2$$

$$\text{Subject to } h_i(x)w = y_i - E_i$$

$$w = h^{\dagger} y$$

MLP:

$$y = \hat{f}(x) = f\left(\sum_{j=1}^{n} W_j f_j\left(\sum_{i=1}^{m} w_{ij} x_i + b_j\right) + B\right)$$

where $x \in \Re^m$, $f(.)$ is the transfer function, w_{ij} is the connection weight between the ith input node and jth hidden node, n is the number of hidden nodes, W_j is the weight on the link between the jth hidden node and output neuron, b_j is the bias of the jth hidden node, and B is the bias.

SEET

- The ensembles were designed using the combination of four variants of the base techniques created using the bagging technique.

- The linear combination rule incorporated the SA as a weight factor to determine the prediction accuracy of the ensemble.

- In this study, to the best of our knowledge, the SA was used as a weight factor for the first time.

- The values of EF for <LSSVR, ELM, MLP, RBF, and CART > were <0.39, 0.45, 0.26, 0.25, and 0.20>, respectively.
- The results indicated that LSSVR, ELM, and MLP performed more suitably than RBF and CART.
- Therefore, the aforementioned three techniques were selected as base learning techniques to develop the ensemble.

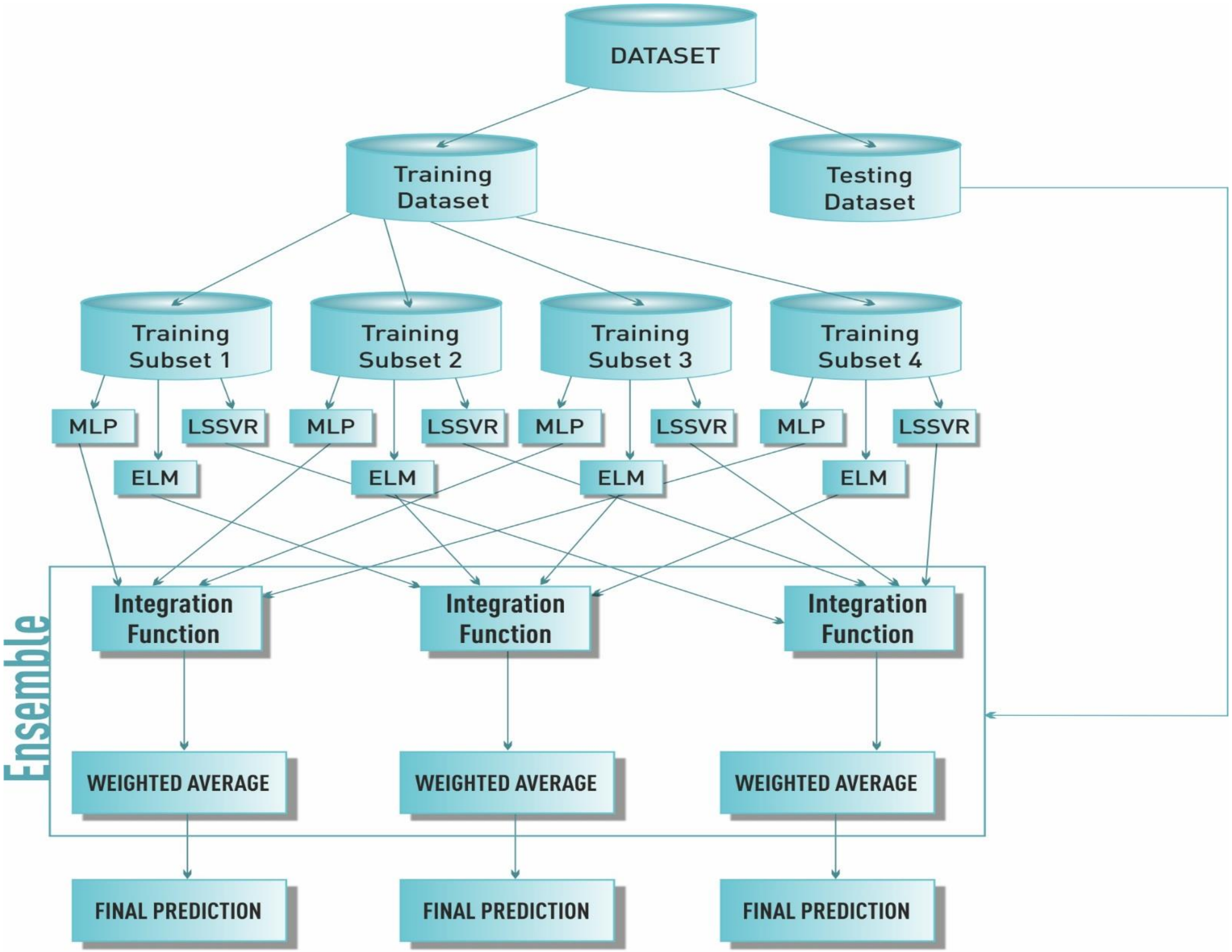
- Four variants of each base learner were generated. The variants were annotated as LSSVR1, LSSVR2, LSSVR3, LSSVR4, ELM1, ELM2, ELM3, ELM4, MLP1, MLP2, MLP3, and MLP4.

- In this study, the weights were initially computed, and the final prediction, as indicated in the proposed algorithm, was then processed.

- The experimental study was conducted using the publicly available PROMISE repository test suite.

- The ensemble techniques performed more suitably than the individual base techniques in most cases.

SEET



The system architecture of SEET

EXPERIMENT SETUP

- **Integration Function**

$$\hat{f}_{int}(.) = \sum_{i=1}^{4} w_i(x) * f_i(x) \qquad w_i(x) = \frac{SA_i}{\sum_i SA_i}$$

Where $w_i(x)$ and $f_i(x)$ are the weights and predictions of the learning techniques, respectively.

- **Goal of experiment**
  - compare prediction accuracy of SEET model and RMSE ensemble model.
- **Performance Criteria**

MMRE, PRED(0.25), SA, EF.

$$MRE = \frac{|\text{Actual effort} - \text{predcited effort}|}{\text{Actual effort}}$$

$$= \frac{y_i - \hat{y}_i}{y_i}$$

$$MMRE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{y_i}$$

$PRED(0.25) = \frac{k}{n}$, Where n denoted the total number of projects and k represents the number of projects whose MRE is less than or equal to 0.25.

EXPERIMENT SETUP

- Performance Criteria

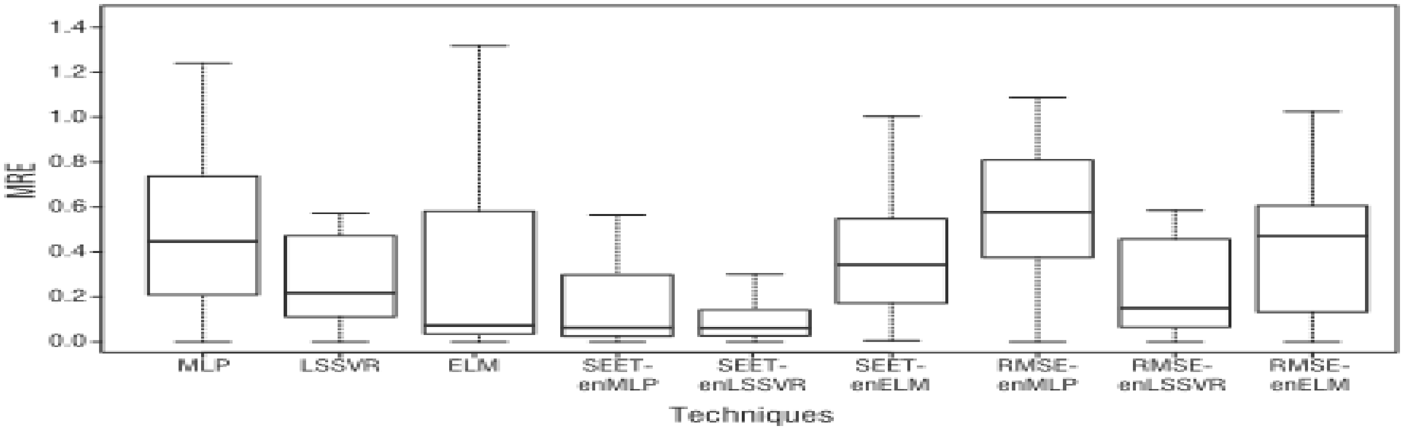
$$SA_{p_i} = \left(1 - \frac{MAR_{p_i}}{MAR_{p_0}}\right) * 100$$

$$EF = \frac{PRED(0.25)}{1+MMRE}$$

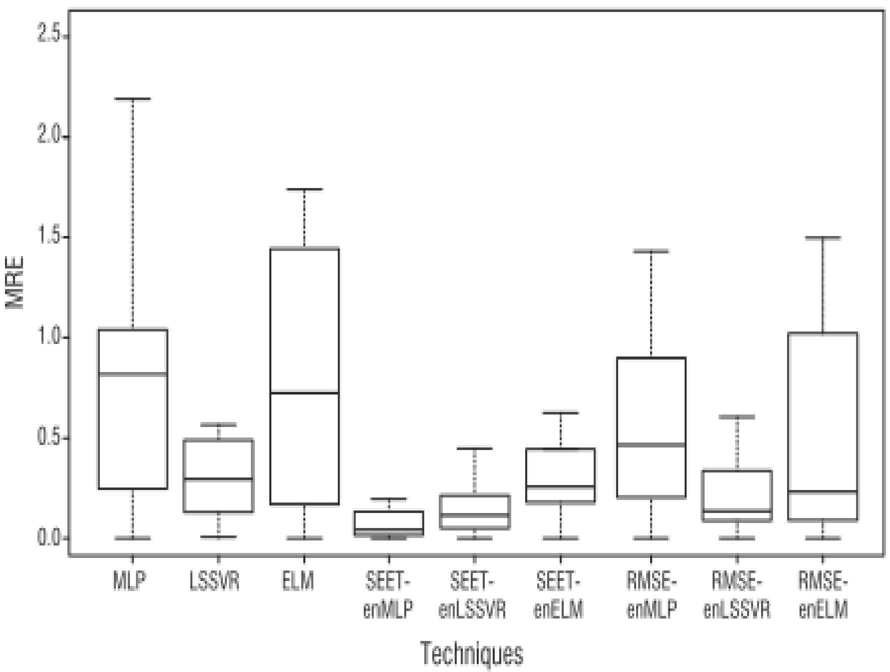
EXPERIMENT SETUP

Technique	Kernel Funciton	Regularisation Parameter
LSSVR	RBF Kernel	γ, σ set to $\left\{B^{\frac{1}{2}}\right\}$ $where\ B\ is: -5:1:14$

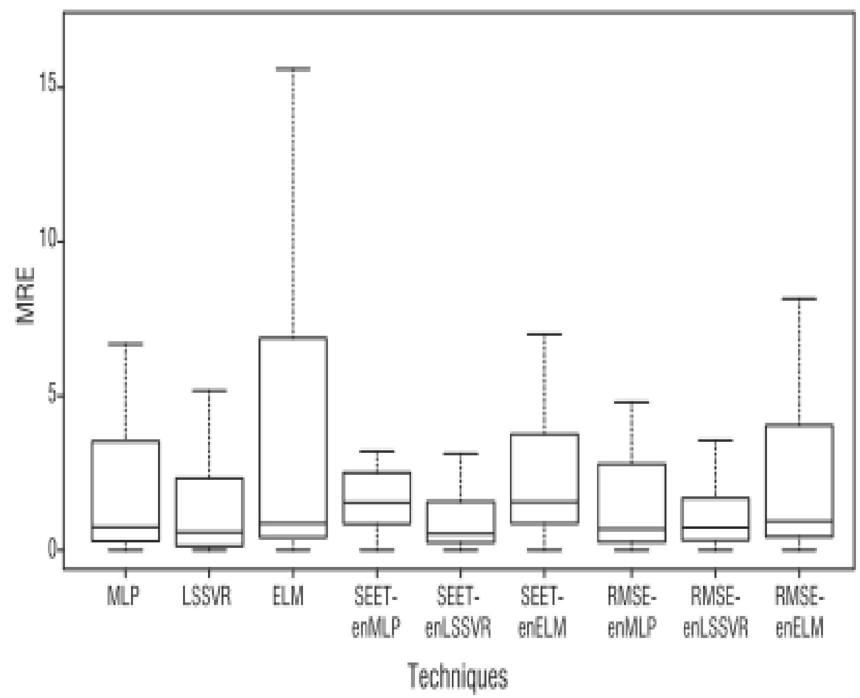
Technique	Weight/ No of hidden Nodes	Activation Funciton Learning algo
MLP	[0,1]/ 3 to 5 hidden nodes Learning rate={0.1,0.2,0.3,0.4,0.5} Momentum={0.1,0.2,0.3,0.4,0.5}	Sigmoid/ Back Propogation
ELM	[0,1]/ 2 to 10 nodes	Sigmoid/ Moore – Penroose Inverse



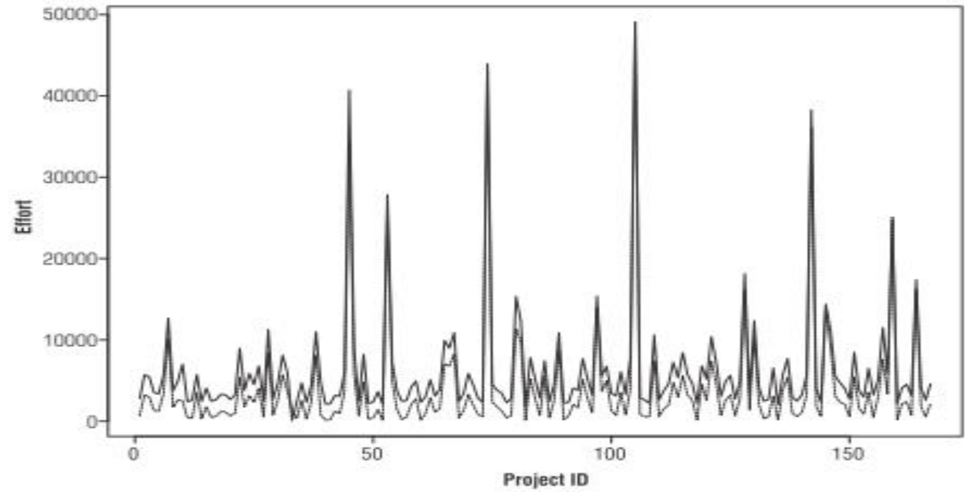
2a) Desharnais dataset



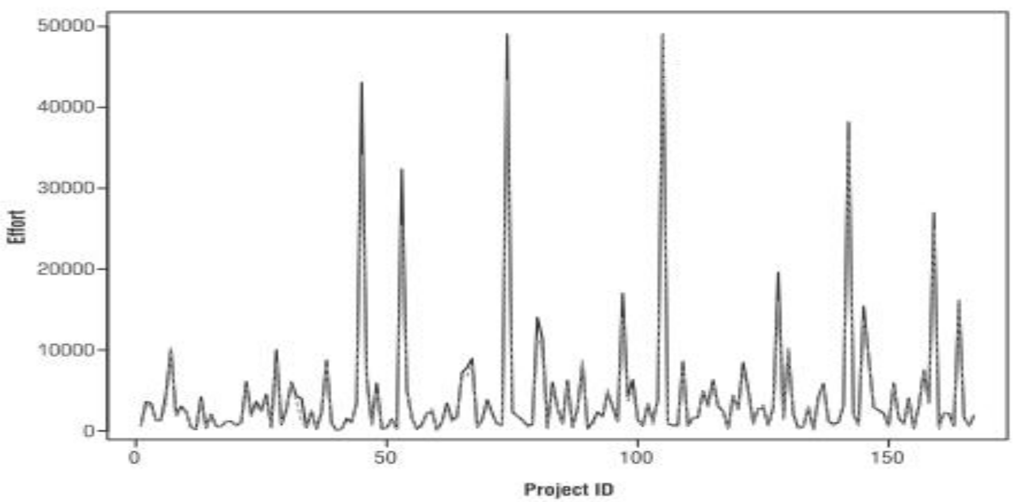
2b) Albrecht dataset



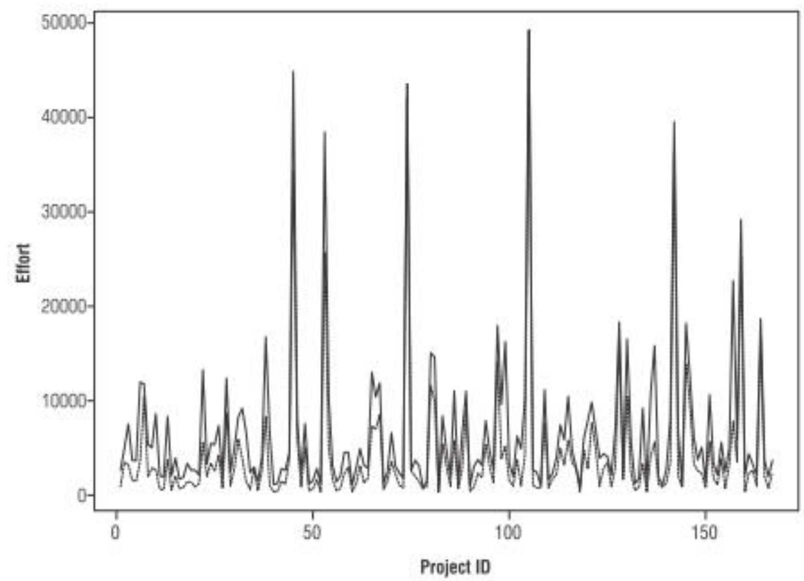
2c) Nasa dataset



a) LSSVR technique



b) SEET-enLSSVR technique



c) RMSE-enLSSVR technique

COMPARISON OF PREDICTION ACCURACY OF ENSEMBLE TECHNIQUES

Win-Loss-Draw	SEET Model	RMSE Model
SEET Model	0-0-6	4-2-0
RMSE Model	2-4-0	0-0-6

Performance Metrics: Evaluation Function (EF)

CONCLUSION

- In this study we introduce SA based integration function for designing ensemble technique.

- The machine learning techniques MLP, LSSVR, and ELM are utilized to build homogeneous ensemble.

- The ensembles are designed by combining four variants of the base techniques created using the bagging approach.

Conclusion

- The linear combination rule incorporates SA as a weight factor in proposed SEET model and RMSE as weight factor in RMSE ensemble model to determine the prediction accuracy.

- The SEET ensemble model performed better in most cases compared with the individual base learners and RMSE ensemble model.

On the Value of Random Vector Functional Link Neural Network in Software Development Effort Estimation

Benala. T.R., Dehuri, S., & Mall, R. (2012). Functional link artificial neural networks for software cost estimation. International Journal of Applied Evolutionary Computation (IJAEC), 3(2), 62-82.

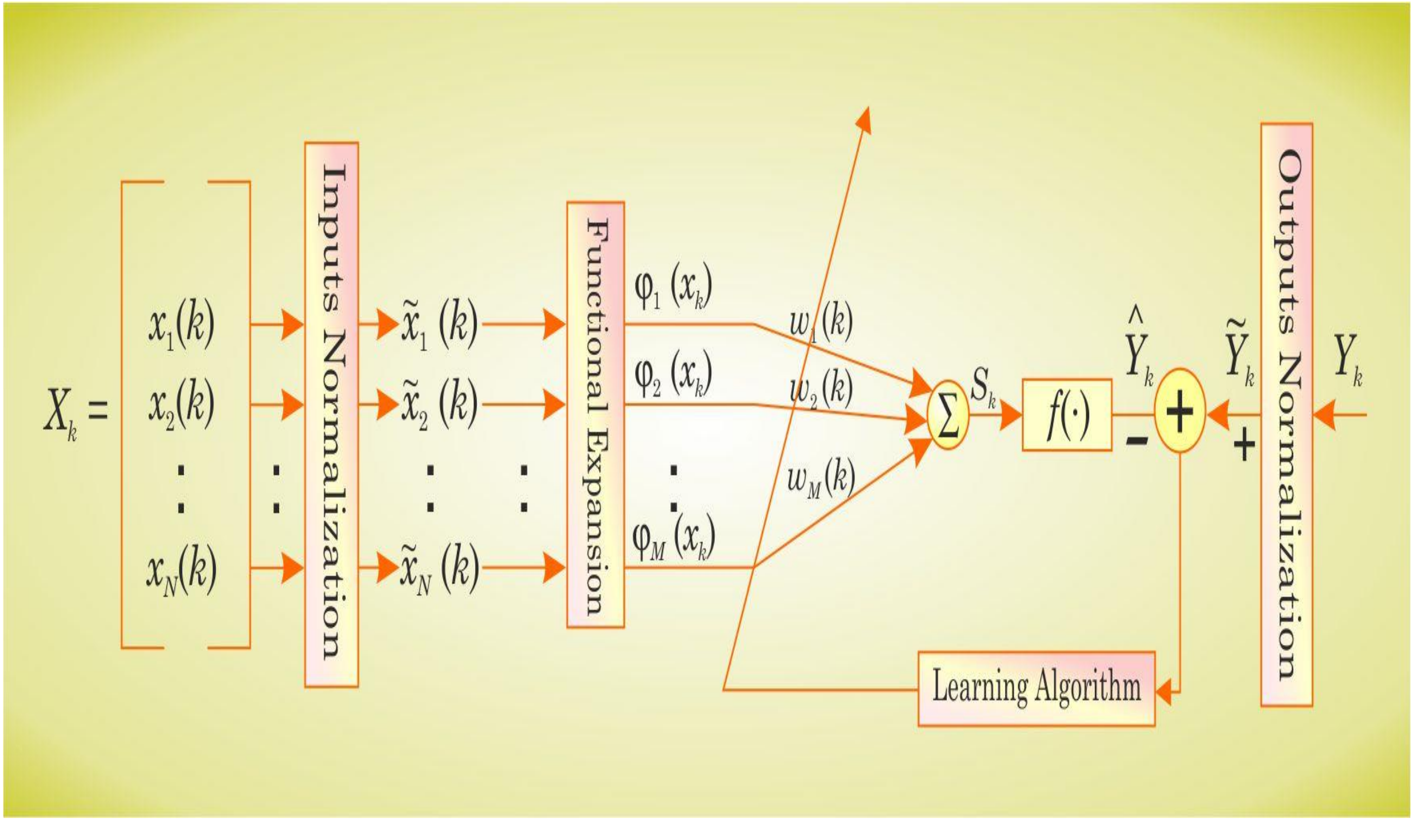
FUNCTIONAL LINK ARTIFICIAL NEURAL NETWORK

- Functional link artificial neural network (FLANN) is higher-order neural networks introduced by Klassen and Pao in 1988.

- The FLANN captures non-linear input–output relationships by expanding the input vector (cost drivers) and then processing the final output layer, thus predicts the output (effort in Person-Months).

- The input vector (cost driver) is expanded using Chebyshev polynomial and the implicit hidden unit is created. The weighted summation approximates the software development effort.

Architecture of FLANN

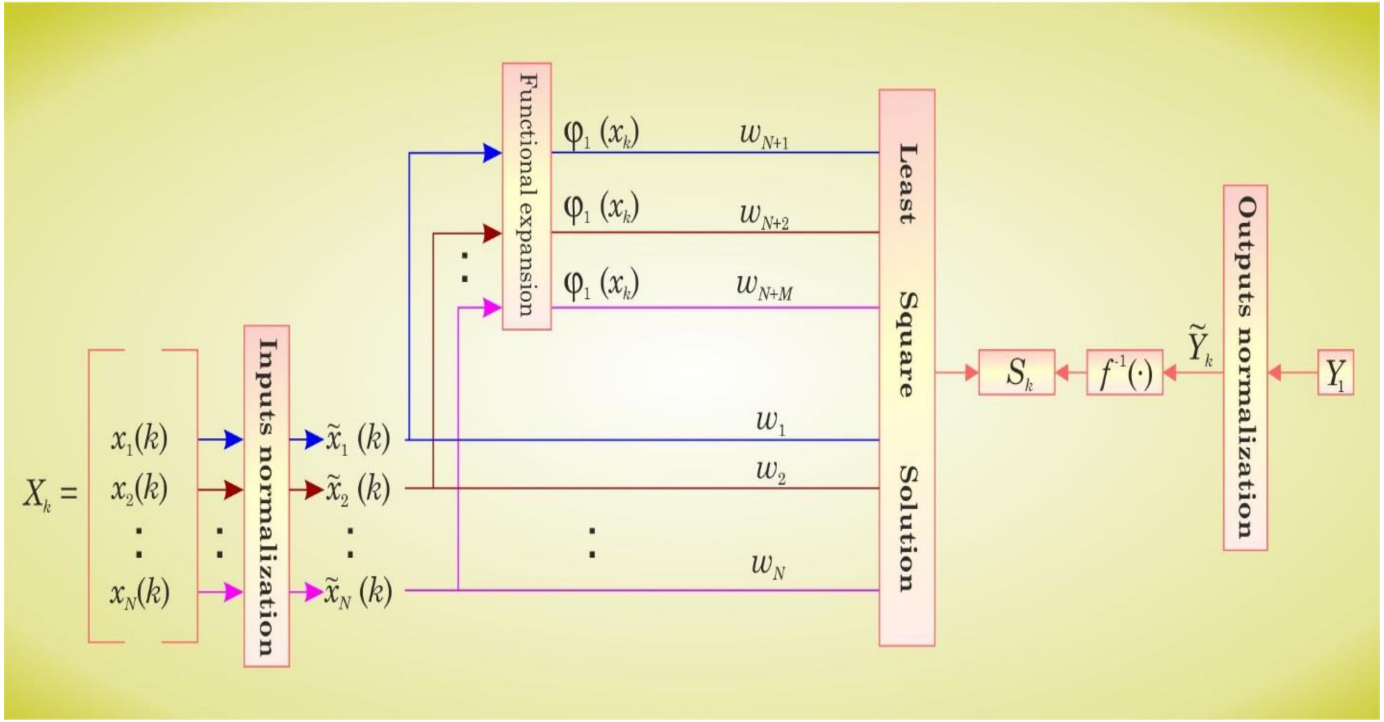


RANDOM VECTOR FUNCTIONAL LINK ARTIFICIAL NEURAL NETWORK

- RVFL is a semi-randomized version of FLANN proposed by Pao et al. (1994) that integrates the benefit of random weights and functional link.

- The RVFL network, a special case of single layer feed forward neural network (SLFN), comprises of three layers: input layer, hidden layer (enhancement layer), and output layer with direct links from the input layer to the output layer.

RVFL Architecture



- The random weights between input layer and hidden layer helps the basis function at hidden layer to avoid saturation point.

- The direct communication from input to output layer aids in regularization of hidden layer output.

- The two-way communication between input and output layer enhances the generalization capability of RVFL architecture.

PROPOSED METHODOLOGY

Given

$FE(.)$: Chebyshev polynomial Functional expansion

$logsig(.)$: logistic sigmoid activation function of the output layer

λ : The ridge regression regularization coefficient

<u>Initialization</u>:

- Regularization parameter $\lambda = \left\{B^{\frac{1}{2}}\right\}$,
- where $B = -5:1:14$
- The weights between input and hidden layer are selected randomly between [-1, +1] using uniform probability distribution.
- Each attribute is normalized using min-max normalization technique
- $\tilde{x}_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$ where $i = 1,2, \ldots, N$ and $j = 1,2, \ldots, k$

The input matrix of RVFL network is:

$$\mathcal{H} = [\widetilde{X}_k, H]^T$$

Where $\mathcal{H} = [\widetilde{X}_k, H]$is the concatenation of input unit $(\widetilde{X}_k)$ and hidden unit (H).

$$H = \begin{pmatrix} \varphi_1(x(1)) & \varphi_2(x(1)) & \dots & \varphi_M(x(1)) \\ \varphi_1(x(2)) & \ddots & \dots & \varphi_M(x(2)) \\ \vdots & \dots & \ddots & \vdots \\ \varphi_1(x(N)) & \varphi_2(x(N)) & \dots & \varphi_M(x(N)) \end{pmatrix}$$

The Hidden Unit is generated by applying of Chebyshev polynomial on input unit.

Recursive formulae of Chebyshev polynomials $(-1 \leq x \leq 1)$

$$A_0(x) = 1$$

$$A_1(x) = x$$

$$A_n(x) = 2xA_{n-1}(x) - A_{n-2}(x), n \geq 2.$$

- Apply the inverse of the output layer nodes activation function to determine the input $\mathrm{O}(\mathrm{i})$ of the output layer node.

$$\mathrm{O}(\mathrm{i}) = \mathrm{logsig}^{-1}(\tilde{\mathrm{y}}) = \ln(\tilde{\mathrm{y}}/(1-\tilde{\mathrm{y}}))$$

$$S(i) = f^{-1}(\hat{y}_i)$$

$$\mathcal{H} \times W = \mathrm{S}$$

- The optimal connection weights between input layer and output layer are obtained using closed form ℓ_2 norm ridge regression.

$$\mathrm{W} = \left(\mathcal{H}^{\mathrm{T}}\mathcal{H} + \lambda\mathrm{I}\right)^{-1}\mathcal{H}^{\mathrm{T}}\mathrm{S}$$

- Predicted effort at training stage is given by:

$$\hat{y}_i = \left(\frac{1}{1 - e^{-\mathcal{H}_i^T W}}\right) \times (y_{max} - y_{min}) + y_{min}$$

Where Y_{max} and Y_{min} are the maximum and minum effort of the training dataset.

- For testing dataset $T = \{\acute{x}_i, \acute{y}_i\}_{i=1}^{L}$, the predicted effort at testing stage is given by:

$$\hat{\acute{y}}_i = \left(\frac{1}{1 - e^{-\acute{\mathcal{H}}_i^T W}}\right) \times (\acute{y}_{max} - \acute{y}_{min}) + \acute{y}_{min}$$

Where $\acute{Y}_{max}$ and $\acute{y}_{min}$ are the maximum and minum effort of the testing dataset respectively.

EXPERIMENT SETUP

- **Goal of experiment**
  - compare prediction accuracy of SEET model and RMSE ensemble model
- **Performance Criteria**

MMRE, PRED(0.25), SA, EF.

$$\mathrm{MRE} = \frac{|\text{Actual effort} - \text{predcited effort}|}{\text{Actual effort}}$$

$$= \frac{y_i - \hat{y}_i}{y_i}$$

$$\mathrm{MMRE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{y_i}$$

$$\mathrm{PRED}(0.25) = \frac{k}{n}$$

Where n denoted the total number of projects and k represents the number of projects whose MRE is less than or equal to 0.25.

EXPERIMENT SETUP

Performance Criteria

$$\mathrm{MAR_j} = \frac{\sum_{i=1}^{n} \left| e_j - \hat{e}_j \right|}{n}$$

e_j: Actual Effort ; $\hat{e}_j$: Predicted Effort

$$\mathrm{SA}_{p_i} = \left(1 - \frac{\mathrm{MAR}_{p_i}}{\mathrm{MAR}_{p_0}}\right) * 100$$

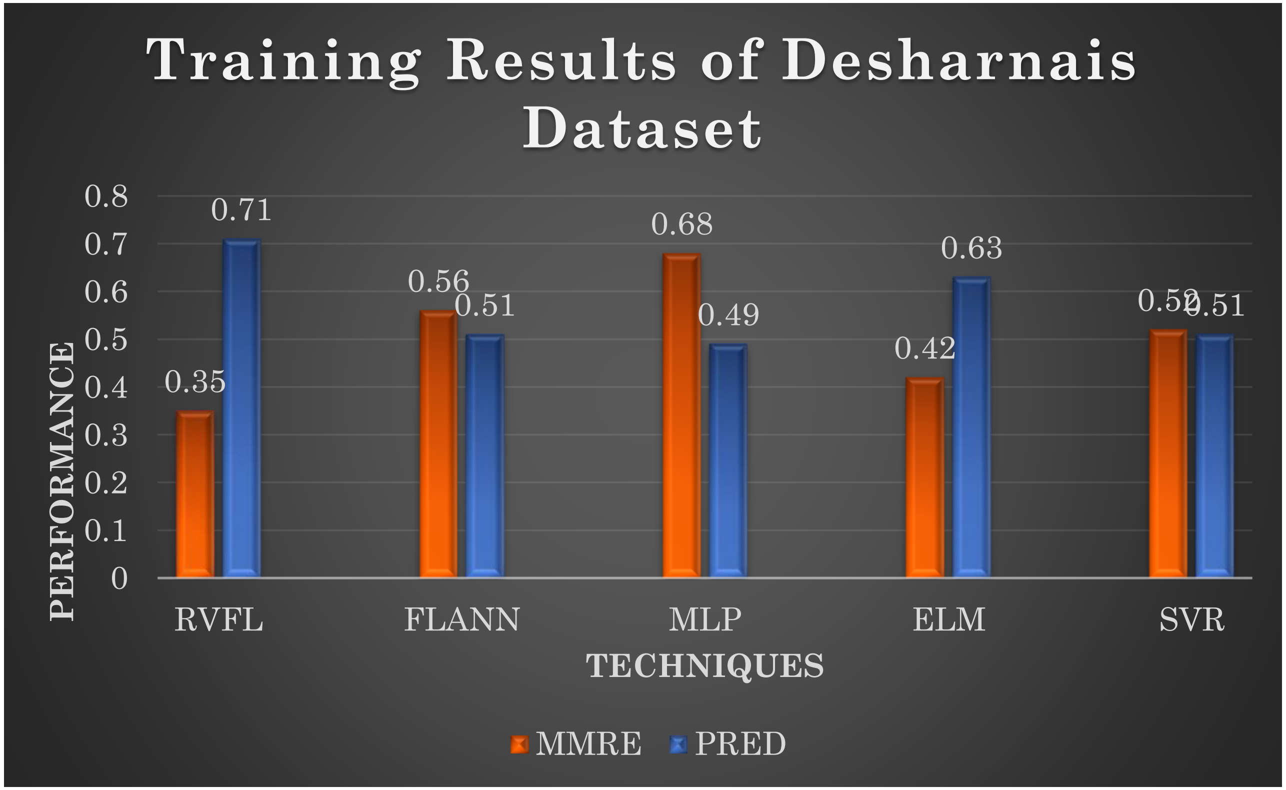
$$\Delta = \frac{\overline{\mathrm{MAR}}_{p_{randomguess}} - \mathrm{MAR}_{p_j}}{\mathrm{SD}_{p_{randomguess}}}$$

Experiment setup

Technique	Kernel Funciton	Regularisation Parameter
SVR	RBF Kernel	Penalty factor:C, Width: γ set to$\{2^{-i}\| i = -10, -4 \ldots .0, .., 4,10\}$.

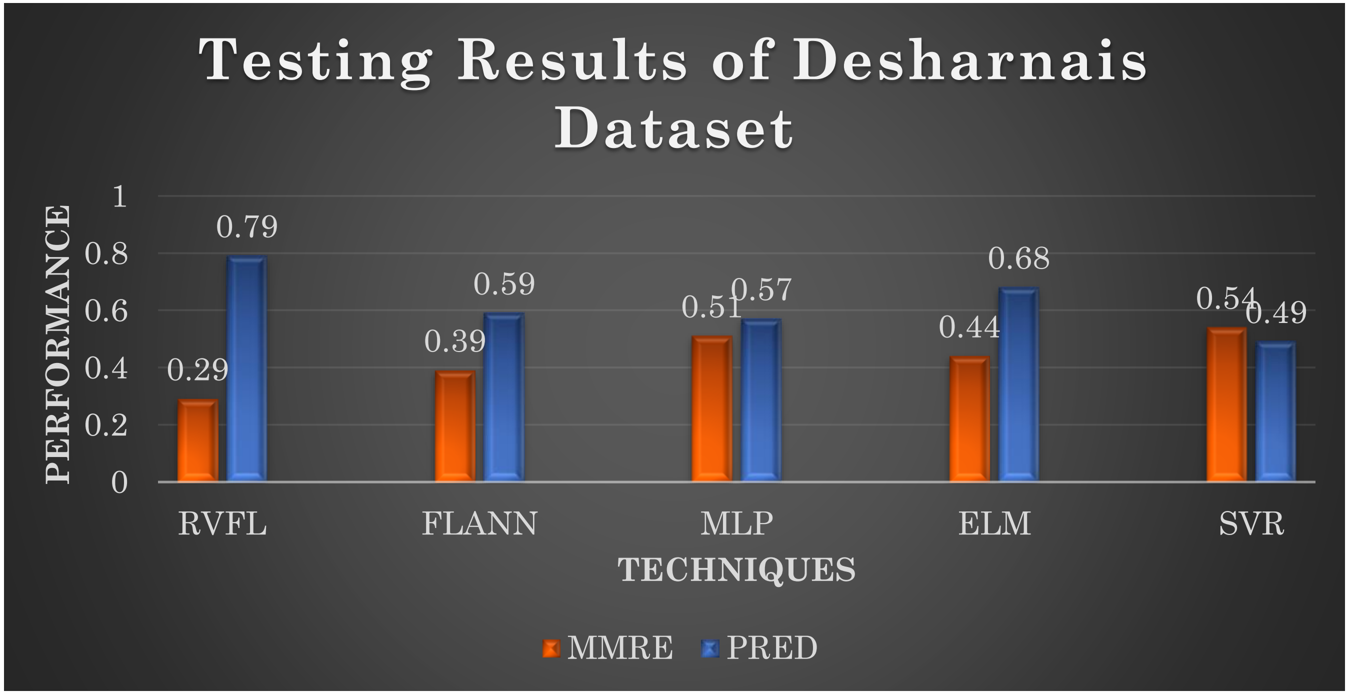
Technique	Weight/ No of hidden Nodes	Activation Funciton Learning algo
MLP	[0,1]/ 3 to 5 hidden nodes Learning rate={0.1,0.2,0.3,0.4,0.5} Momentum={0.1,0.2,0.3,0.4,0.5}	Sigmoid/ Back Propogation
ELM	[0,1]/ 3 to 5 hidden nodes	Sigmoid/ Moore – Penroose Inverse
FLANN	[0,1]/ Chebyshev Polynomial	Sigmoid/ Back propagation

Accuracy Comparisons for the training results on Desharnais dataset



RESULT ANALYSIS

Accuracy Comparisons for the testing results on Desharnais dataset



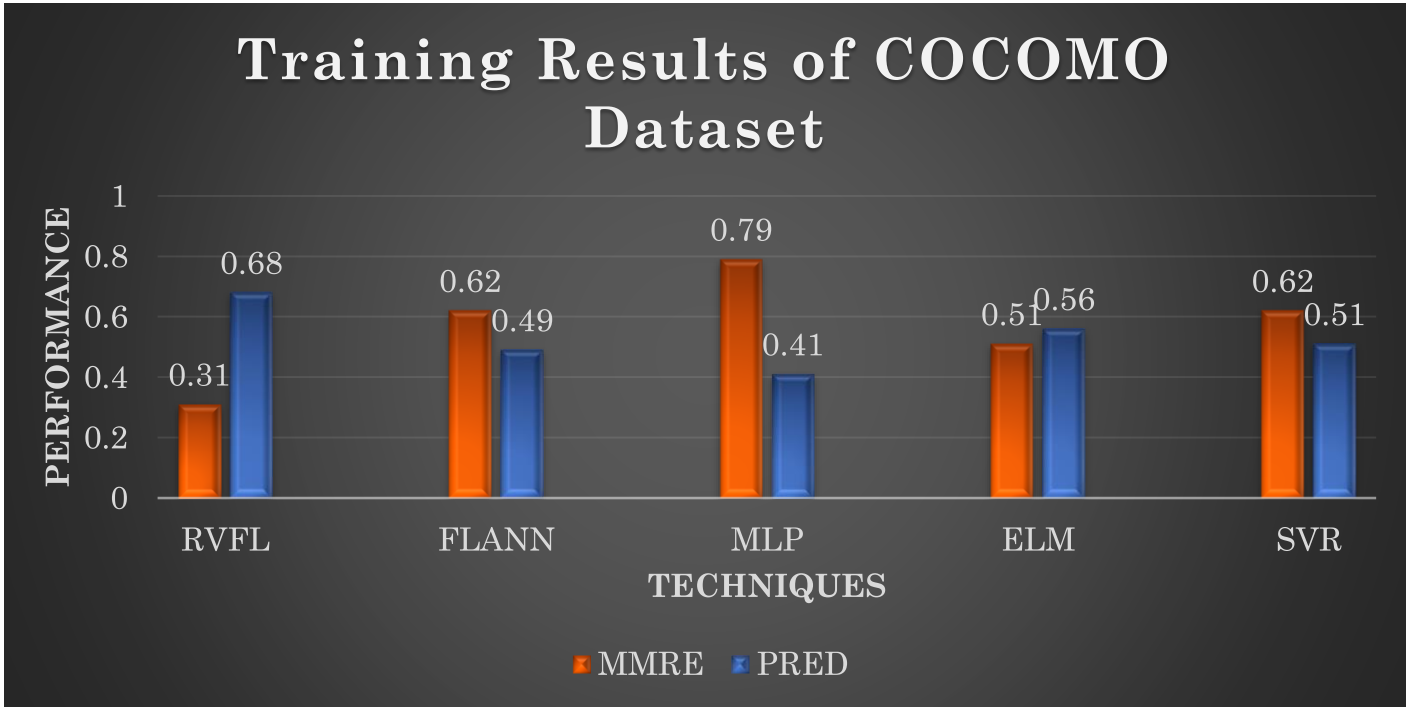
RESULT ANALYSIS

Percentage of Improvement of RVFL model for Desharnais Dataset

Techniques	MMRE		PRED	
	Training (%)	Testing (%)	Training (%)	Testing (%)
RVFL vs. FLANN	60	34	39	33
RVFL vs. ANN	94	75	44	34
RVFL vs. ELM	20	51	13	14
RVFL vs. SVR	29	86	39	38
Average Improvement	50.75	61.50	33.75	29.75

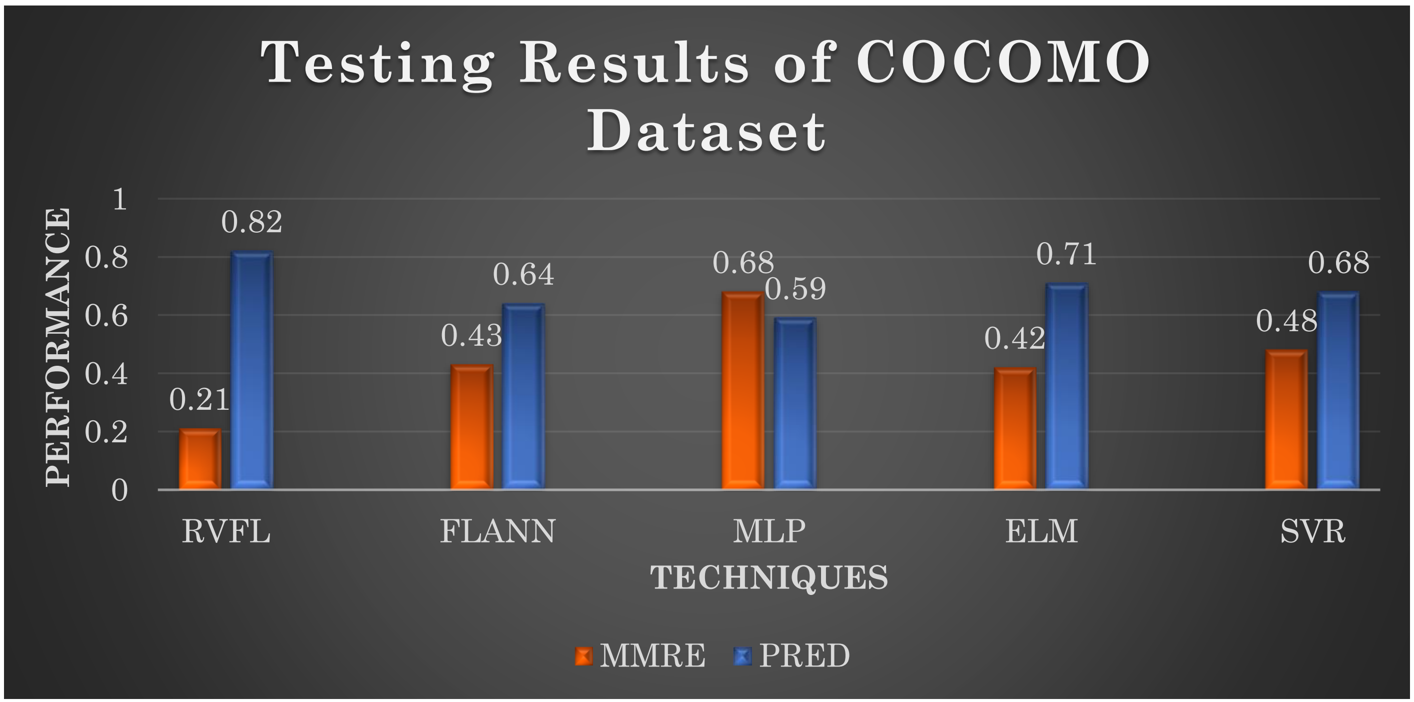
RESULT ANALYSIS

Accuracy Comparisons for the training results on COCOMO dataset



RESULT ANALYSIS

Accuracy Comparisons for the testing results on COCOMO dataset



RESULT ANALYSIS

Percentage of Improvement of RVFL model for COCOMO Dataset

Tecniques	MMRE		PRED	
	Training (%)	Testing (%)	Training (%)	Testing (%)
RVFL vs. FLANN	**37**	**38**	**38**	**28**
RVFL vs. ANN	**75**	**119**	**65**	**39**
RVFL vs. ELM	**13**	35	21	15
RVFL vs. SVR	**37**	54	33	21
Average Improvement	**40.5**	61.5	39.25	25.75

FUTURE RESEARCH

- In DABE, we have focused on a fixed value of control parameters and single objective function.We plan to incorporate DE-based Muiltiobjective optimization techniques for analogy-based SDEE.

- We would examine the performance of the proposed DABE model under simultaneous feature weight optimization and project Selection.

- MMRE and PRED were used for objective function design and criticized as biased error indicators. Objective function consisting of SA and Δ must be constructed and its performance investigated.

FUTURE RESEARCH

- Further studies on EEE include designing homogeneous and heterogeneous ensembles by using various machine learning techniques exhaustively.

- Applying both linear and nonlinear combination rules and validating the correctness of the model by using rigorous statistical tests.

- Application of evolutionary computation, computational intelligence, and swarm intelligence-based nonlinear combination rules.

Future Research

- Future research for the RVFL-SDEE technique includes considering six popular activation functions namely, sigmoid, sine, hardlin, tribas, radbas, sign.

- The research work can be extended to evaluate different closed-form based RVFL configurations listed as follows (Zhang and Suganthan 2016):

- RVFL with and without bias in the output neuron.

- RVFL with and without direct link from input layer to output layer.

REFERENCES

- Brooks Jr, F. P. (2003). Three great challenges for half-century-old computer science. Journal of the ACM (JACM), 50(1), 25-26.
- Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology, 54(1), 41-59.
- Pfleeger, S. L., & Atlee, J. M. (1998). Software engineering: theory and practice. Pearson Education India.
- Fred, B., & Brooks, P. (1975). The mythical man month. In Proceedings of the international conference on Reliable software (p. 193).
- Kitchenham, B. A. (2004). Procedures for performing systematic reviews. Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1.
- Kitchenham, B. A. (2007). Guidelines for performing systematic literature reviews in software engineering, Tech. Rep., EBSE-2007-001, UK (July 2007). URL:
- <http//www.dur.ac.uk/ebse/>.

REFERENCES

- Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation, 13(2), 398-417.
- Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5), 945-958.
- Shin, M., & Goel, A. L. (2000). Empirical data modeling in software engineering using radial basis functions. IEEE Transactions on Software Engineering, 26(6), 567-576.
- Klasser MS, Pao YH (1988) Characteristics of the functional link net: a higher order delta rule net. IEEE proceedings of 2nd annual international conference on neural networks, San Diago, CA.
- Pao, Y. H., Park, G. H., & Sobajic, D. J. (1994). Learning and generalization characteristics of the random vector functional-link net. Neurocomputing, 6(2), 163-180.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324.

REFERENCES

- Kitchenham, B. A. (1996). Software metrics: measurement for software process improvement. Blackwell Publishers, Inc.
- Kocaguneli, E., Menzies, T., & Keung, J. W. (2012). On the value of ensemble effort estimation. IEEE Transactions on Software Engineering, 38(6), 1403-1416.
- Kitchenham, B., Madeyski, L., Budgen, D., Keung, J., Brereton, P., Charters, S., ... & Pohthong, A. (2017). Robust statistical methods for empirical software engineering. Empirical Software Engineering, 22(2), 579-630.
- Shepperd, M., &MacDonell, S. (2012). Evaluating prediction systems in software project estimation. Information and Software Technology, 54(8), 820-827.

THANK YOU