Predictive Exploration of Method-Level Bugs: An empirical study

Under the Guidance of
Dr.B.Tirimula Rao ,M.Tech , Ph.D
Assistant Professor , HOD IT

Presented by
S.Soniya
21VV1D8801

November 16, 2023

Abstract

- This project focuses on method-level bug predictions using code metrics and historical measures.
- The aim is to evaluate the effectiveness of different prediction models and assess the predictive power of each metric.
- By collecting and analyzing 18 open-source datasets and employing various prediction models(Random forest ,SVM,Naive bayes,Logistic,Decision tree)
- The results demonstrate that the model achieves good prediction performance on large-scale software datasets.

Problem Statement

- Numerous studies have explored defect prediction at various granularity levels, but method-level prediction models are rarely utilized.
- To address Method-level bug prediction , our project seeks to comprehensively analyze method-level bug predictions using 18 open-source datasets.
- We aim to compare different prediction models, analyze the significance of each metric, and identify the best metrics for constructing method-level bug prediction models.

Existing system

- **Limited Model Variety:**
  The existing system only utilizes Random Forest (RF), Naive Bayes (NB), Logistic Regression, and Decision Tree models.

- **Metric Selection:**
  The existing system focuses on AUC, MCC, and F1 Score as performance metrics.

Proposed system

- **Model Diversity:**
  Introduces Support Vector Machine (SVM).

- **Metric Selection:**
  The proposed system employs a broader set of performance metrics, including AUC, accuracy, MCC, F1 Score, precision, and geometric mean.

- **Tool Integration::**
  Leveraging the Weka tool and integrating it seamlessly with Java, the proposed system enhances the analytical capabilities for model development, evaluation, and analysis, providing a robust platform for research.

18 large scale Datasets

Datasets	URL
ActiveMQ:5.16.0	https://github.com/apache/activemq
Avro:1.9.1	https://github.com/apache/avro
Calcite:1.24.0	https://github.com/apache/calcite
Camel:2.22.0	https://github.com/apache/camel
Cassandra:3.11.6	https://github.com/apache/cassandra
CXF:3.3.7	https://github.com/apache/cxf
Drill:1.15.0	https://github.com/apache/drill
Flink:1.8.0	https://github.com/apache/flink
Flume:1.8.0	https://github.com/apache/flume
Hbase:3.6.3	https://github.com/apache/hbase
Ignite:0.94.8	https://github.com/apache/ignite
Kafka:2.6.0	https://github.com/apache/kafka
Maven:2.5.1	https://github.com/apache/maven
Nutch:1.17	https://github.com/apache/nutch
PDFBox:2.0.20	https://github.com/apache/pdfbox
Struts:2.5.20	https://github.com/apache/struts
Wicket:8.8.0	https://github.com/apache/wicket
Zookeeper:3.5.8	https://github.com/apache/zookeeper

List of method-level Code metrics

Index	Metric name
C1	Lines of Code
C2	Number of Comment lines
C3	Number of all lines
C4	Number of Blank lines
C5	Number of Declare lines
C6	Number of Executable lines
C7	Number of Parameters
C8	Number of Statements
C9	Number of Declare Statements
C10	Number of Executable Statements
C11	Halstead-Vocabulary
C12	Halstead-Length
C13	Halstead-Difficulty
C14	Halstead-Volume
C15	Halstead-Effort
C16	Halstead-Bugs
C17	Cyclomatic complexity
C18	Number of Path
C19	MaxNesting
C20	Fan-in
C21	Fan-out

List of method-level History measures

Index	Measure name
H1	Added LOC
H2	Deleted LOC
H3	Changed LOC
H4	Number of Changes
H5	Number of Authors
H6	Number of Modified Statements
H7	Number of Modified Expressions
H8	Number of Modified Comments
H9	Number of Modified Return type
H10	Number of Modified Parameters
H11	Number of Modified Prefix
H12	Added LOC/LOC
H13	Deleted LOC/LOC
H14	Added LOC/Deleted LOC
H15	Changed LOC/Number of Changes
H16	Number of Modified Statements/ Number of Statements
H17	Number of Modified Expressions/ Number of Statements
H18	Number of Modified Comments/ LOC
H19	Number of Modified Parameters/ Number of Parameters

Dataset

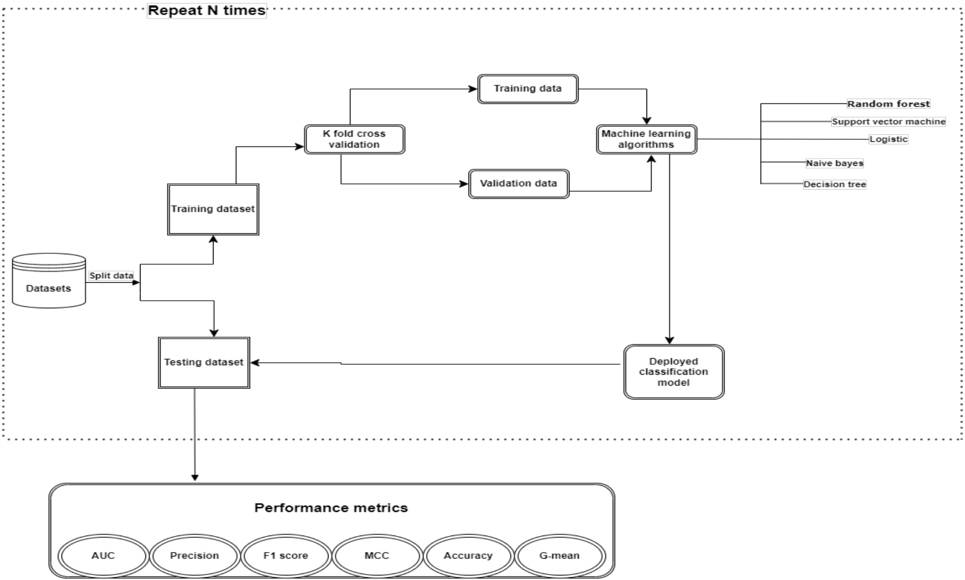
	Method name	C1	C2	C3	C4	C5	C6
0	org.apache.avro.data.Json.parseJson	7	4	11	0	1	4
1	org.apache.avro.data.Json.write	53	3	56	0	3	48
2	org.apache.avro.data.Json.read	29	3	32	0	3	26
3	org.apache.avro.data.RecordBuilderBase.validate	5	11	16	0	1	2
4	org.apache.avro.data.RecordBuilderBase.default...	3	8	11	0	1	1

	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
0	1	4	1	3	20.0	27.0	9.0	116.692059	4860.0	0.095639	2	2
1	2	41	4	37	0.0	0.0	0.0	0.000000	0.0	0.000000	11	11
2	1	27	7	20	0.0	0.0	0.0	0.000000	0.0	0.000000	13	12
3	2	3	1	2	21.0	30.0	12.5	131.769523	7875.0	0.131941	2	2
4	1	2	1	1	0.0	0.0	0.0	0.000000	0.0	0.000000	1	1

	C19	C20	C21	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12
0	1	5	5	2	2	4	1	1	4	0	0	0	0	0	0.285714
1	2	18	24	0	0	0	2	0	4	0	0	0	0	1	0.000000
2	3	11	18	0	0	0	1	0	0	0	0	0	0	1	0.000000
3	1	45	3	2	1	3	1	1	1	4	0	0	0	0	0.400000
4	0	10	4	13	20	33	3	1	26	3	4	0	0	0	4.333333

	H13	H14	H15	H16	H17	H18	H19	bug-prone
0	0.285714	1.00	4.0	1.000000	0.0	0.000000	0.0	not bug-prone
1	0.000000	0.00	0.0	0.097561	0.0	0.000000	0.0	not bug-prone
2	0.000000	0.00	0.0	0.000000	0.0	0.000000	0.0	not bug-prone
3	0.200000	2.00	3.0	0.333333	0.8	0.000000	0.0	not bug-prone
4	6.666667	0.65	11.0	13.000000	1.0	1.333333	0.0	bug-prone

Architecture



F1,Mcc,Auc values using Random forest

Datasets	Type1			Type2			Type3		
	F1	MCC	AUC	F1	MCC	AUC	F1	MCC	AUC
ActiveMQ	0.663	0.313	0.718	0.751	0.483	0.845	0.784	0.555	0.876
Avro	0.632	0.243	0.730	0.709	0.420	0.780	0.724	0.432	0.808
Calcite	0.559	0.144	0.599	0.617	0.262	0.688	0.644	0.297	0.710
Camel	0.619	0.235	0.650	0.711	0.429	0.797	0.722	0.446	0.811
Cassandra	0.613	0.224	0.665	0.700	0.394	0.791	0.742	0.471	0.835
CXF	0.752	0.210	0.711	0.824	0.415	0.865	0.836	0.463	0.885
Drill	0.623	0.245	0.702	0.680	0.391	0.769	0.706	0.427	0.802
Flink	0.629	0.068	0.679	0.687	0.235	0.748	0.714	0.298	0.779
Flume	0.626	0.243	0.649	0.661	0.311	0.782	0.693	0.368	0.800
Hbase	0.635	0.284	0.685	0.698	0.401	0.781	0.711	0.418	0.796
Ignite	0.811	0.055	0.708	0.818	0.145	0.875	0.828	0.227	0.881
Kafka	0.649	0.096	0.678	0.677	0.187	0.769	0.710	0.268	0.818
Maven	0.696	0.073	0.721	0.761	0.310	0.868	0.778	0.347	0.886
Nutch	0.590	0.158	0.690	0.664	0.331	0.766	0.712	0.419	0.806
PDFBox	0.683	0.3628	0.773	0.724	0.432	0.786	0.762	0.528	0.858
Struts	0.600	0.255	0.677	0.733	0.458	0.825	0.760	0.502	0.849
Wicket	0.616	0.125	0.698	0.731	0.387	0.852	0.763	0.439	0.869
Zookeeper	0.639	0.276	0.682	0.708	0.429	0.788	0.695	0.389	0.793

- **TYPE1 :** Prediction models based on code metrics;
- **TYPE2 :** Prediction models based on history measures;
- **TYPE3 :** Prediction models based on both code metrics and history measures

prediction model development

- we proceed to implement various prediction models. This involves employing sophisticated algorithms such as Random Forest, Support Vector Machine, Logistic Regression, Naive Bayes, and Decision tree.

  Each of these models brings distinct strengths and capabilities, allowing us to thoroughly evaluate their performance.

- To assess the effectiveness of the prediction models, we employ a diffrent evaluation metrics. These include the F1 Score, AUC-ROC, Matthews Correlation Coefficient (MCC), Precision, G-mean, and Accuracy.

  These are used to evaluate and compare the effectiveness of different predictive models

Result analysis

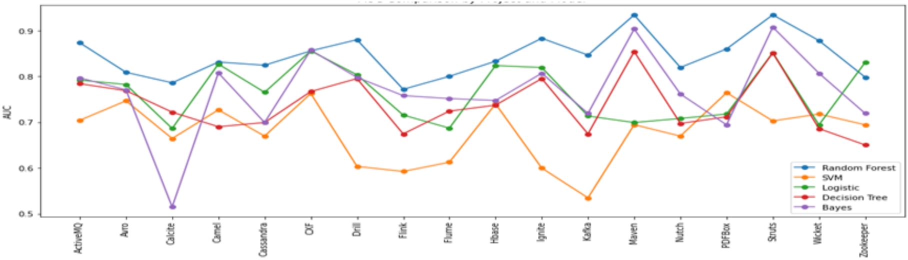
Dataset	Model	AUC	Precision	MCC	Accuracy	F-measur	G-mean	Dataset	Model	AUC	Precision	MCC	Accuracy	F-measur	G-mean
ACTIVE MQ	Random forest	**0.8732**	**0.7798**	**0.5563**	**0.7862**	**0.7859**	0.7235	AVRO	Random forest	**0.8087**	0.7564	0.526	0.7562	0.7842	0.761
	SVM	0.7039	0.7539	0.3744	0.7245	0.7211	0.6256		SVM	0.7464	**0.822**	**0.5783**	**0.7903**	**0.7743**	0.7451
	Logistic	0.7917	0.7283	0.4069	0.7334	0.7219	0.6847		Logistic	0.7821	0.7557	0.4899	0.758	0.755	0.7386
	Decision tree	0.7838	0.7489	0.4622	0.7447	0.7463	**0.7339**		Decision tree	0.7686	0.7742	0.5308	0.7741	0.7741	**0.7653**
	Bayes	0.7966	0.7216	0.4101	0.7198	0.7208	0.7031		Bayes	0.7704	0.746	0.4672	0.7338	0.7248	0.7177
Calcite	Random forest	**0.7858**	0.7352	0.4274	**0.7291**	0.7143	**0.6899**	Camel	Random forest	**0.8311**	0.8689	**0.4781**	0.8369	0.8408	**0.8305**
	SVM	0.6641	0.6364	0.3561	0.6435	0.6014	0.6094		SVM	0.7267	0.7899	0.3382	0.7967	0.7808	0.635
	Logistic	0.6861	0.6775	0.3271	0.6735	0.6751	0.665		Logistic	0.8267	**0.8789**	0.3781	**0.8969**	**0.8808**	0.5805
	Decision tree	0.7217	**0.7391**	**0.4555**	0.709	0.7313	0.7314		Decision tree	0.6899	0.8667	0.3334	0.8857	0.8726	0.5782
	Bayes	0.5148	0.535	0.125	0.5883	**0.7465**	0.5017		Bayes	0.8074	0.8772	0.3034	0.6858	0.7421	0.7291
Cassandra	Random forest	**0.8242**	0.7295	**0.459**	**0.7294**	**0.7294**	**0.7295**	CXF	Random forest	**0.8555**	0.8645	**0.4633**	**0.8778**	**0.8661**	**0.7786**
	SVM	0.6694	0.6891	0.3501	0.6631	0.6565	0.6616		SVM	0.7625	0.7465	0.3127	0.7613	0.7439	0.6816
	Logistic	0.765	0.6993	0.3938	0.6949	0.6929	0.6943		Logistic	**0.8555**	0.8476	0.3313	0.8698	0.8529	0.5776
	Decision tree	0.6991	0.6843	0.3686	0.6843	0.6843	0.6843		Decision tree	0.7667	0.8365	0.3	0.8503	0.8424	0.5971
	Bayes	0.6991	**0.7998**	0.412	0.7055	0.705	0.7052		Bayes	0.8572	**0.8681**	0.3982	0.7856	0.8126	0.7592
Drill	Random forest	**0.8799**	**0.7928**	0.5847	**0.7918**	**0.7916**	**0.7919**	Flink	Random forest	**0.7716**	**0.7737**	**0.3405**	**0.7807**	0.7378	0.5852
	SVM	0.6033	0.6478	0.3237	0.6489	0.6126	0.6162		SVM	0.5922	0.7477	0.2476	0.7603	0.7065	0.5529
	Logistic	0.8033	0.7476	0.4937	0.7459	0.7456	0.7462		Logistic	0.7154	0.7448	0.3246	0.7663	**0.7456**	0.6215
	Decision tree	0.795	0.7666	0.5332	0.7665	0.7665	0.7661		Decision tree	0.6742	0.7368	0.3163	0.752	0.7419	0.6368
	Bayes	0.7976	0.6979	0.3915	0.6745	0.6933	0.6938		Bayes	0.7578	0.743	0.3314	0.7172	0.7267	**0.6762**

Result analysis

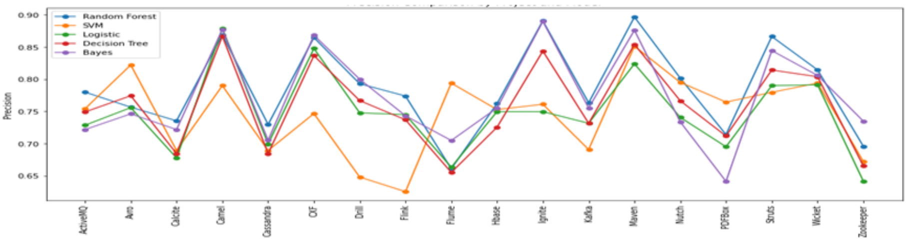
Dataset	Model	AUC	Precision	MCC	Accuracy	F-measure	G-mean	Dataset	Model	AUC	Precision	MCC	Accuracy	F-measure	G-mean
Flume	Random forest	**0.8**	0.6607	0.3882	**0.6926**	0.6698	0.6601	Hbase	Random forest	**0.833**	**0.7621**	0.472	**0.7624**	**0.766**	**0.7649**
	SVM	0.6124	0.6251	0.2308	0.6	0.595	0.6141		SVM	0.7398	0.7527	0.493	0.7413	0.7527	0.7352
	Logistic	0.6868	0.6631	0.3302	0.6631	0.6632	0.6601		Logistic	0.8234	0.7495	**0.4986**	0.7493	0.7492	0.749
	Decision tree	0.724	0.655	**0.3891**	0.6842	0.6849	**0.6853**		Decision tree	0.737	0.7253	0.45	0.7253	0.7253	0.7253
	Bayes	0.7514	**0.7046**	0.3424	0.6825	**0.6884**	0.6752		Bayes	0.7473	0.6467	0.3125	0.6666	0.6841	0.6857
Ignite	Random forest	**0.883**	**0.8906**	0.2278	**0.875**	0.8189	0.6363	Kafka	Random forest	**0.8464**	**0.7632**	**0.4155**	**0.7747**	**0.7584**	0.6825
	SVM	0.5995	0.7609	0.2135	0.8101	0.866	0.4461		SVM	0.5345	0.6906	0.3559	0.6501	0.7142	0.5034
	Logistic	0.8192	0.7494	**0.4985**	0.7493	0.7492	0.7491		Logistic	0.7133	0.7314	0.3559	0.7252	0.728	0.6801
	Decision tree	0.7098	0.8436	0.2956	0.8482	**0.8458**	0.6097		Decision tree	0.752	0.7314	0.3559	0.7252	0.728	0.6801
	Bayes	0.7582	0.8898	0.3881	0.7031	0.7544	**0.7806**		Bayes	0.7539	0.7552	0.3669	0.6538	0.6686	**0.6988**
Maven	Random forest	**0.934**	**0.8965**	**0.5668**	**0.9012**	**0.888**	0.6897	Nutch	Random forest	**0.8193**	**0.8016**	**0.4698**	**0.8095**	**0.7915**	0.6836
	SVM	0.6943	0.8512	0.3527	0.8667	0.8355	0.5429		SVM	0.6694	0.795	0.4402	0.8015	0.7769	0.6537
	Logistic	0.699	0.8238	0.3255	0.8197	0.8217	0.6484		Logistic	0.708	0.7406	0.3407	0.7301	0.7346	0.6737
	Decision tree	0.8097	0.854	0.4414	0.854	0.854	0.7083		Decision tree	0.697	0.7664	0.407	0.7619	0.7639	**0.705**
	Bayes	0.9033	0.876	0.5218	0.8626	0.8681	**0.7784**		Bayes	0.7613	0.7336	0.3109	0.6825	0.6976	0.6712
PDFBox	Random forest	**0.8598**	0.7138	**0.5629**	0.7225	**0.7638**	0.713	Struts	Random forest	**0.9342**	**0.8662**	**0.6187**	**0.863**	**0.8638**	**0.8632**
	SVM	0.7642	**0.7643**	0.4576	**0.7456**	0.7025	**0.7226**		SVM	0.7026	0.7795	0.4876	0.7559	0.7345	0.7007
	Logistic	0.7502	0.6949	0.3869	0.6925	0.6922	0.693		Logistic	0.8304	0.7899	0.5584	0.7916	0.79	0.7747
	Decision tree	0.7176	0.7116	0.4231	0.7115	0.7115	0.7115		Decision tree	0.8513	0.8141	0.6086	0.8095	0.8107	0.8083
	Bayes	0.694	0.6411	0.2714	0.6274	0.6229	0.6311		Bayes	0.9074	0.8443	0.6576	0.8219	0.8235	0.8367
Wicket	Random forest	**0.878**	**0.8143**	0.4308	**0.8256**	0.8015	0.6602	Zookeeper	Random forest	**0.798**	0.6956	0.3887	0.6979	0.6961	0.6897
	SVM	0.6687	0.7938	0.319	0.8035	0.7545	0.5478		SVM	0.6652	0.6718	0.3272	0.6619	0.6755	0.6674
	Logistic	0.7695	0.7912	0.3946	0.8079	0.7939	0.6558		Logistic	0.6689	0.6413	0.2663	0.6338	0.6355	0.6345
	Decision tree	0.6857	0.8037	**0.4374**	0.8145	**0.8071**	0.6927		Decision tree	0.65	0.6651	0.3163	0.6619	0.663	0.6591
	Bayes	0.8067	0.8057	0.432	0.7615	0.7753	**0.7427**		Bayes	0.7189	**0.7346**	**0.4453**	**0.7183**	**0.7194**	**0.7267**

Visualizing Results

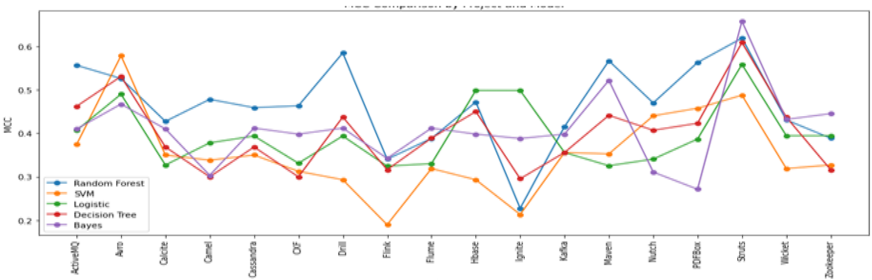
AUC VALUES COMPARISON



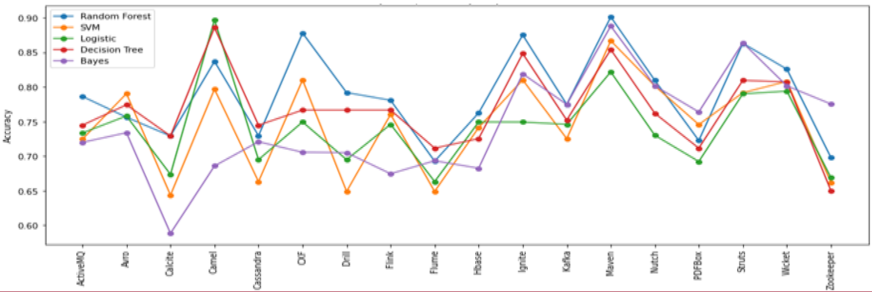
PRECISION VALUES COMPARISON



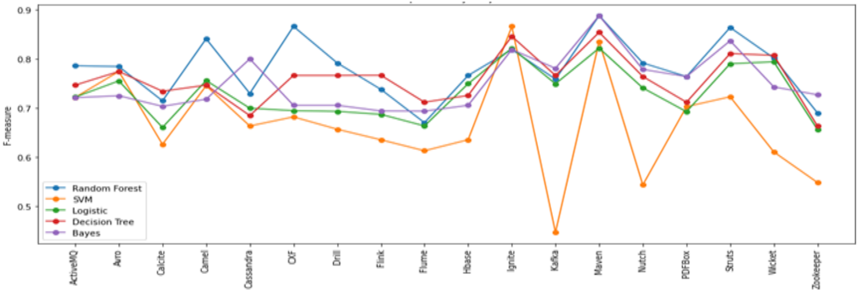
MCC VALUES COMPARISON



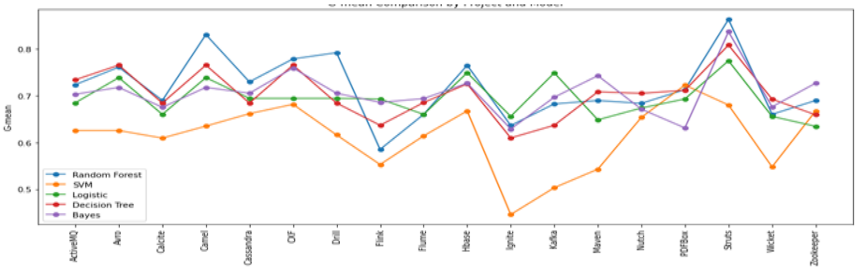
ACCURACY VALUES COMPARISON



F1 VALUES COMPARISON



G-MEAN VALUES COMPARISON



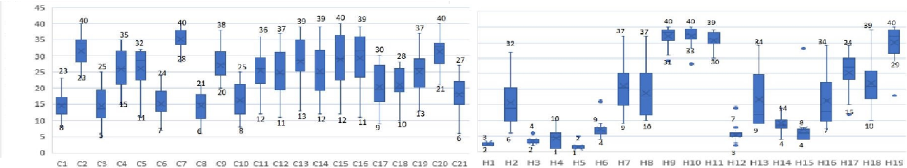
Result Analysis

- From the results ,Random Forest consistently outperforms other models across all 18 datasets in terms of AUC.
- Show how models perform differently on various datasets, underlining the need to choose models based on each dataset's unique features.
- Among the evaluation metrics, the AUC demonstrates the highest performance, indicating strong discriminative ability.
- Information Gain (IG) as a metric to evaluate how effectively each attribute contributes to making accurate predictions, thereby assessing the predictive power of individual features.

Rank of each metrics

Metric	Rank	Metric	Rank	Metric	Rank	Metric	Rank
H5	1.5	C1	15.3	H7	21.4	C13	27.4
H1	2.5	C6	15.9	H18	21.7	C15	28.0
H3	3.5	H2	16.0	C19	24.2	C16	29.0
H4	4.3	C10	16.1	C12	24.2	C2	31.2
H12	5.4	H13	17.0	C11	25.0	C20	32.3
H6	7.2	H16	17.3	H17	25.0	C7	34.1
H15	8.5	C21	17.7	C14	25.4	H19	34.6
H14	10.2	H8	19.7	C4	25.9	H11	35.3
C8	15.0	C17	20.5	C5	26.1	H10	36.8
C3	15.3	C18	21.2	C9	26.6	H9	37.0

Rank distributions for each code metric and history measures



Random forest AUC values with selected metrics

Datasets	TOP 1%	TOP 5%	TOP10%	TOP20%	TOP30%
ActiveMQ	0.786	0.799	0.807	0.822	0.838
Avro	0.673	0.655	0.751	0.773	0.788
Calcite	0.627	0.648	0.661	0.679	0.724
Camel	0.781	0.821	0.812	0.682	0.838
Cassandra	0.734	0.776	0.767	0.760	0.789
CXF	0.817	0.848	0.836	0.845	0.853
Drill	0.638	0.703	0.738	0.759	0.767
Flink	0.691	0.728	0.714	0.723	0.753
Flume	0.703	0.673	0.753	0.765	0.786
Hbase	0.677	0.707	0.753	0.787	0.795
Ignite	0.787	0.818	0.846	0.873	0.875
Kafka	0.730	0.716	0.735	0.757	0.788
Maven	0.815	0.821	0.845	0.861	0.879
Nutch	0.703	0.693	0.705	0.741	0.774
PDFBox	0.624	0.698	0.713	0.731	0.771
Struts	0.765	0.796	0.806	0.807	0.822
Wicket	0.780	0.815	0.823	0.840	0.863
Zookeeper	0.725	0.723	0.744	0.759	0.769

- Just using the top 30% ranked metrics, all the derived models have substantial performance with a AUC larger than 0.7; Even only using the top 1 ranked metric, most of the derived models can achieve a good performance on method-level bug prediction.

Conclusion

- This overall study presents how systematically build models for predicting bug-prone methods, and provides empirical evidence to best select metrics to build method-level bug prediction models.
- No single model consistently excels across all datasets. The study underscores the importance of considering dataset characteristics in model selection, providing valuable insights for practical bug prediction applications.

Reference

[1] Ran Moa, Shaozhi Wei a, Qiong Feng b, Zengyang Li "An exploratory study of bug prediction at the method level "

[2] Yasutaka Kamei , Emad Shihab ,"Defect Prediction: Accomplishments and Future Challenges," IEEE Conference Publication — IEEE Xplore, Mar. 01, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7476771

Thankyou