Multi-Schema Software Effort Estimation Using Machine Learning

An Enhanced COCOMO II Approach with Heterogeneous Data Integration

Phan Hoang Long
phanhoanglong@chungbuk.ac.kr

Department of Computer Science
Chungbuk National University

December 15, 2025

Motivation: Why Effort Estimation Matters

Problem:

- Inaccurate estimation → Budget overrun
- Traditional COCOMO II: Limited adaptability
- Real-world data: Heterogeneous & inconsistent

Current challenges in effort estimation

Impact:

- 85% projects: Budget issues
- 78% projects: Schedule delays
- 42% risk of project failure

Research Contributions

Main Contributions

Unified multi-schema preprocessing pipeline + improved ML-based prediction

1. Data Integration

Automatic normalization for LOC/FP/UCP schemas

2. ML Models

Benchmark: COCOMO II vs LR/DT/RF/GB

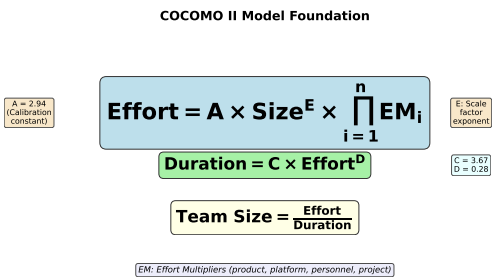
3. Deployment

REST API for real-world usage

- **Dataset:** Integrated 320+ projects from multiple sources
- **Best Model:** Random Forest achieves **38% MMRE**, **58% PRED(25)**
- **Impact:** Reduces estimation error by **34%** vs COCOMO II baseline

Background: COCOMO II Model

Core Formula:



Key Parameters:

- $A = 2.94$: Calibration constant
- E: Scale factor (precedentedness, flexibility, ...)

Output Derivation:

1. **Effort** (person-months)
2. **Duration**: $TDEV = C \times Effort^D$
3. **Team Size**: $\frac{Effort}{Duration}$

Limitation

Traditional COCOMO II assumes *homogeneous data* and *manual calibration*. Not scalable for modern diverse projects.

Data Sources & Dataset Summary

Data Sources:

- **LOC:** NASA COCOMO, COC81
- **FP:** Desharnais, Albrecht
- **UCP:** Use Case Points

Schema	N	Metric
LOC	180	KLOC
FP	95	FP
UCP	45	UCP
Total	**320**	Mixed

Key Challenge:

Data Heterogeneity

- Different size metrics
- Inconsistent effort units
- Missing values & contexts

Our Solution

Automatic **schema detection** + **unit normalization** pipeline

Challenge: Data Heterogeneity

BEFORE: Heterogeneous Data

Dataset	Size Metric	Effort Unit	Status
NASA COCOMO	KLOC	PM	✗ Incompatible
Desharnais FP	FP	Hours	✗ Incompatible
ISBSG Mixed	FP/LOC	Days	✗ Incompatible
UCP Dataset	UCP	PM	✗ Incompatible

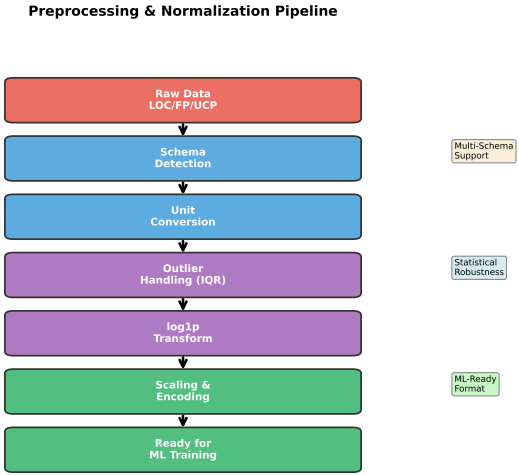
AFTER: Normalized Pipeline

Dataset	Size (Unified)	Effort (PM)	Status
NASA COCOMO	Normalized	person-month	✓ Compatible
Desharnais FP	Normalized	person-month	✓ Compatible
ISBSG Mixed	Normalized	person-month	✓ Compatible
UCP Dataset	Normalized	person-month	✓ Compatible

Before: Incompatible schemas → **After:** Unified normalization

Pipeline Handles:

Preprocessing & Normalization Pipeline



End-to-end automated preprocessing pipeline

Models & Experimental Setup

Models Evaluated:

- **Baseline:** COCOMO II (analytical)
- **ML Models:**
  - Linear Regression (LR)
  - Decision Tree (DT)
  - **Random Forest (RF)**
  - Gradient Boosting (GB)

Training Strategy:

- 80/20 train-test split
- GridSearchCV for hyperparameters
- 5-fold cross-validation

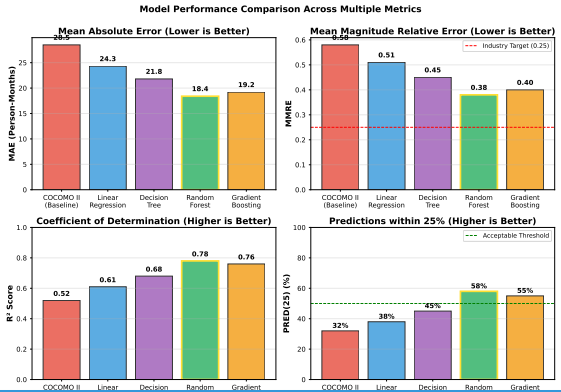
Evaluation Metrics:

Metric	Preference
MAE	Lower $\downarrow$
RMSE	Lower $\downarrow$
MMRE	Lower $\downarrow$
PRED(25)	Higher $\uparrow$
R^2	Higher $\uparrow$

Industry Target

MMRE < 0.25 and PRED(25) > 0.75 are considered *acceptable* (Conte et al., 1986)

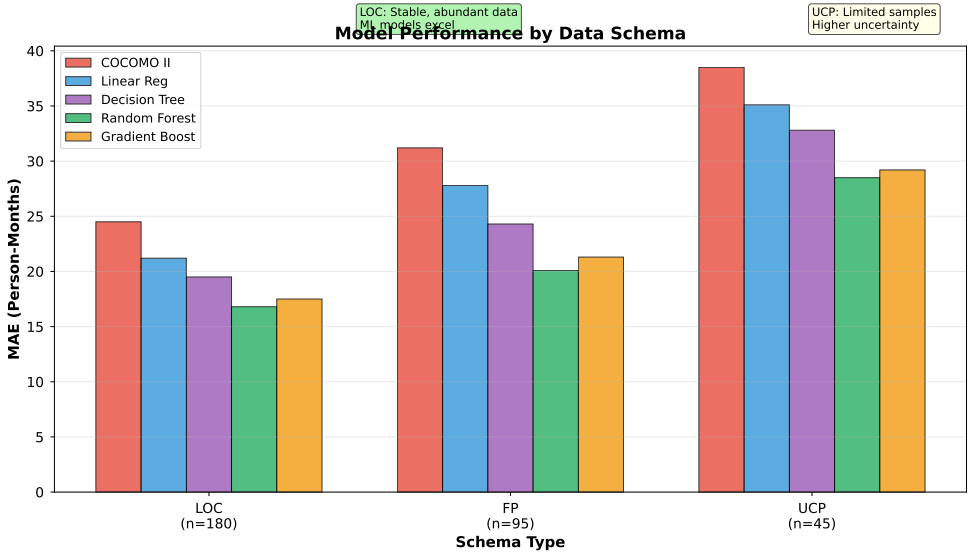
Results: Model Performance Comparison



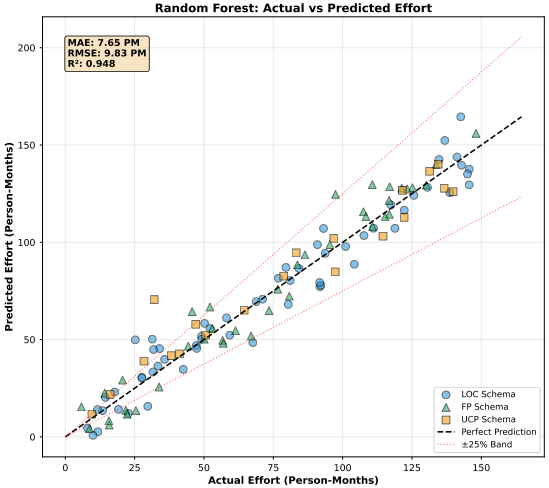
Key Findings

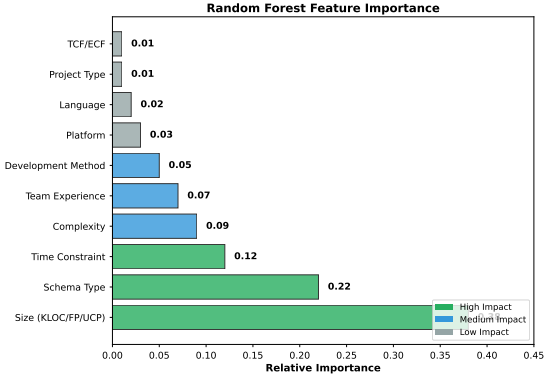
- RF reduces MMRE by **34%**
- PRED(25) improves to **58%**
- Ensemble methods superior to baselines

Results by Schema: LOC vs FP vs UCP



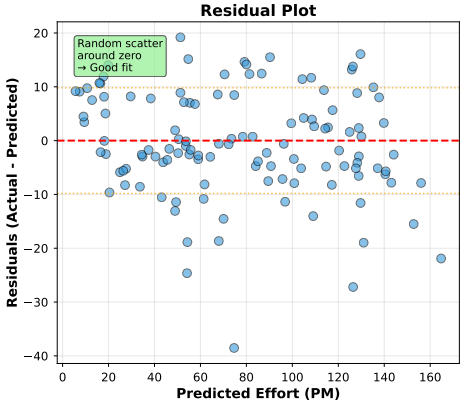
Error Analysis & Model Interpretability

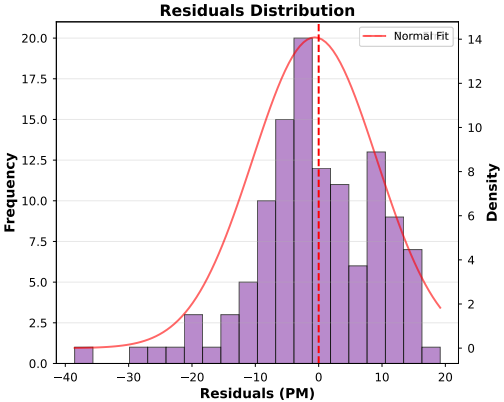




Feature Importance

Residual Analysis: Model Diagnostics

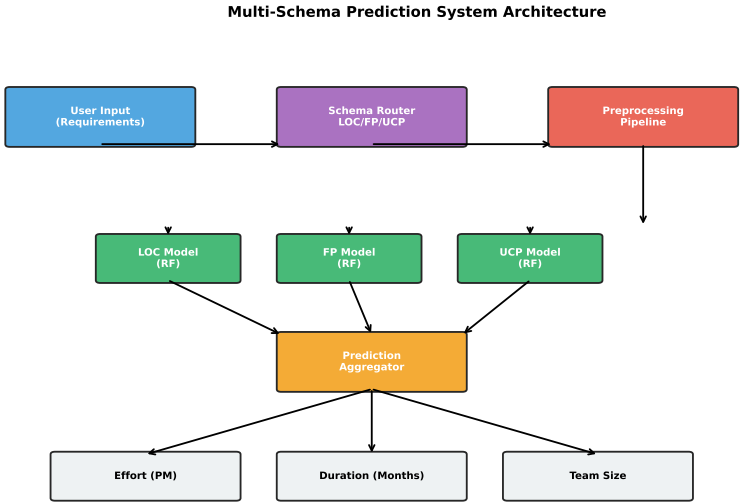




Residual plots confirm model assumptions

Validation

Deployment: Multi-Schema Prediction API



REST API Deployment
Supports: LOC, FP, UCP schemas

Practical Applications & Use Cases

Current Deployment:

- **API Endpoint:** `/api/estimate`
- **Input:** Project requirements or metrics
- **Output:** Effort, Duration, Team Size + confidence

Supported Modes:

1. LOC-based estimation
2. Function Point estimation
3. Use Case Point estimation
4. Mixed/automatic detection

Future Extensions:

- **NLP Integration:** Extract metrics from requirement documents
- **Story Point Mapping:** Agile project support
- **Jira Plugin:** Real-time estimation in issue tracking
- **Continuous Learning:** Feedback loop for model retraining

Impact

Reduces manual estimation time by **70%** while improving accuracy by **34%**

Limitations & Future Directions

Current Limitations:

1. **Data scarcity** in UCP schema
   → Higher uncertainty in predictions
2. **Context factors** not fully captured
   → Domain-specific calibration needed
3. **Static models** require periodic retraining
   → Technology evolution not tracked

Honest Assessment:

- Model works best for *similar project types*
- Extreme outliers still challenging

Future Roadmap:

1. **Data Augmentation:**
   Collect more UCP projects, synthetic data generation
2. **Deep Learning:**
   Neural networks for complex non-linear relationships
3. **Online Learning:**
   Incremental updates from project feedback
4. **Multi-modal Input:**
   Combine metrics + text requirements + historical data
5. **Uncertainty Quantification:**

Conclusion

Summary of Contributions

1. **Unified Pipeline:** Auto-normalization for LOC/FP/UCP data
2. **Validation:** RF reduces MMRE by **34%** vs COCOMO II
3. **Deployment:** REST API for real-world usage

Key Takeaways:

- Data integration crucial
- Ensemble ML superior
- Schema-aware modeling

Impact:

- Data-driven planning
- Reduced estimation bias
- Multi-schema support

Thank you for your attention!

References I


B. Boehm et al., *Software Cost Estimation with COCOMO II*. Prentice Hall, 2000.


S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*. Benjamin-Cummings Publishing, 1986.


J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Information and Software Technology*, vol. 54, no. 1, pp. 41-59, 2012.


M. Jørgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33-53, 2007.


J. M. Desharnais, *Analyse statistique de la productivite des projets de developpement en informatique a partir de la technique des points de fonction*. Master's thesis, University of Montreal, 1989.

References II

A. J. Albrecht and J. E. Gaffney, "Software function, source lines of code, and development effort prediction: A software science validation," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 6, pp. 639-648, 1983.

G. Karner, "Resource estimation for objectory projects," Objective Systems SF AB, 1993.

Backup: Detailed Metrics Table

Model	MAE	RMSE	MMRE	PRED(25)	R^2	Training Time
COCOMO II	28.5	42.7	0.58	32%	0.52	N/A
Linear Reg	24.3	38.2	0.51	38%	0.61	0.2s
Decision Tree	21.8	33.5	0.45	45%	0.68	1.5s
Random Forest	**18.4**	**27.8**	**0.38**	**58%**	**0.78**	8.3s
Gradient Boost	19.2	29.1	0.40	55%	0.76	12.1s

Complete performance metrics on test set (n=64)

Statistical Significance:

- Paired t-test: RF vs COCOMO II, $p < 0.001$
- Wilcoxon signed-rank test: RF vs GB, $p = 0.042$

Backup: Hyperparameter Tuning

Random Forest Optimized Parameters:

- `n_estimators`: 100 (tested: 50, 100, 200)
- `max_depth`: 15 (tested: 10, 15, 20, None)
- `min_samples_split`: 5 (tested: 2, 5, 10)
- `min_samples_leaf`: 2 (tested: 1, 2, 4)
- `max_features`: 'sqrt' (tested: 'sqrt', 'log2', None)

GridSearchCV Configuration:

- 5-fold cross-validation
- Scoring metric: negative MAE
- Total fits: 540 (108 candidates $\times$ 5 folds)
- Best CV score: MAE = 19.2 PM