Dự đoán thời gian và chi phí hoàn thành dự án phần mềm sử dụng XGBoost

Lê Thế Anh[1, 2*], Huỳnh Quyết Thắng[1], Nguyễn Thanh Hùng[1]

[1]Đại học Bách khoa Hà Nội, Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội, Việt Nam;
[2]Trường Đại học Kỹ thuật - Hậu cần Công an Nhân dân, Phường Hồ, Thuận Thành, Bắc Ninh, Việt Nam.
[*]Email: anhlt.ict@gmail.com
Nhận bài: 19/1/2024; Hoàn thiện: 15/3/2024; Chấp nhận đăng: 19/3/2024; Xuất bản: 22/4/2024.
DOI: https://doi.org/10.54939/1859-1043.j.mst.94.2024.149-158

TÓM TẮT

Với sự phát triển vượt bậc của công nghệ thông tin, quản lý chi phí và thời gian hoàn thành dự án phần mềm trở thành một vấn đề cấp thiết. Để có thể quản lý các dự án phần mềm, nhu cầu về dự đoán chi phí và thời gian hoàn thành là vô cùng quan trọng. Các phương pháp truyền thống thường sử dụng phương pháp quản lý giá trị thu được EVM để dự đoán chi phí và thời gian hoàn thành dự án. Tuy nhiên, phương pháp này thường đạt được độ chính xác không quá cao khi dữ liệu có nhiều nhiễu. Những năm gần đây, các phương pháp học máy xuất hiện như một giải pháp hữu ích cho việc tận dụng các dữ liệu trong quá khứ để dự đoán các giá trị trong tương lai. Trong nghiên cứu này, chúng tôi đề xuất sử dụng mô hình học máy XGBoost để dự đoán chi phí và thời gian hoàn thành dự án. Kết quả thực nghiệm cho thấy XGBoost có tiềm năng trong việc giải quyết bài toán này.

Từ khoá: Quản lý dự án phần mềm; EVM; XGBoost.

1. TỔNG QUAN

Sự phát triển vượt bậc của công nghệ thông tin tạo ra ảnh hưởng mạnh mẽ đến phong cách sinh hoạt và lối sống của con người. Hoạt động quản lý phần mềm cũng nằm trong phạm vi ảnh hưởng đó. Mô hình quản lý dự án phần mềm này khác với quản lý dự án truyền thống ở chỗ các dự án phần mềm nhìn chung có các pha cố định, đòi hỏi nhiều vòng kiểm tra, cập nhật và phản hồi của khách hàng.

Quản lý giá trị thu được (EVM- Earned value management) là một trong những kỹ thuật nổi tiếng để kiểm soát thời gian và chi phí của một dự án [1, 6]. Phương pháp này dựa trên một tập hợp các độ đo để đo lường và đánh giá tình trạng tổng thể của một dự án nhằm đưa ra cảnh báo sớm cho người quản trị dự án về các vấn đề của dự án. Tuy nhiên, phương pháp này có một số hạn chế như: chỉ dựa trên các chi phí trong quá khứ, dự đoán thiếu tính tin cậy trong giai đoạn sớm của dự án và không tính đến các thống kê dự báo [6] . Ba điểm hạn chế trên là lý do chính dẫn đến việc phải phát triển các phương pháp mới [1, 6]. Một trong các phương pháp đó chính là việc sử dụng các phân tích hồi quy tuyến tính hoặc phi tuyến để phát triển các mô hình hồi quy, hay còn được gọi là các mô hình tăng trưởng (GM-Growth Models) [2].

Hiện nay, có nhiều mô hình dự đoán chi phí hoàn thành dự án, cũng như các mô hình dự đoán thời điểm kết thúc của dự án. Các mô hình khác nhau đã được nghiên cứu và so sánh một cách đầy đủ nhờ Batselier, J và các cộng sự trong [4]. Một số nghiên cứu liên quan sử dụng phương pháp EVM có thể kể đến như: Khamooshi and Golafshani (2014) trong [5] đề xuất được phương pháp mới EDM đã cải thiện hơn hầu hết các phương pháp ESM, còn Elshaer (2013) trong [7] đã mở rộng được phương pháp cũ ESM rất hiệu quả trong giai đoạn sớm của dự án, nhưng kém hiệu quả trong giai đoạn sau của dự án. Nhóm tác giả Narbaev T.; De Marco A. (2014) [3] đã đề xuất phương pháp kết hợp các mô hình tăng trưởng và phương pháp EVM cho một số kết quả khả quan.

Tại Việt Nam, hiện có một số nghiên cứu liên quan đến dự báo thời gian kết thúc dự án và dự đoán chi phí hoàn thành dự án bằng cách áp dụng các hệ số hiệu suất khác nhau trong phương pháp kết hợp mô hình tăng trưởng kết hợp EVM nhằm nâng cao chất lượng dự báo [14].

Tuy nhiên, phương pháp này cũng có một số điểm hạn chế sau: chỉ dựa trên các số liệu trong quá khứ, dự đoán thiếu tin cậy trong giai đoạn sớm của dự án, không tính đến các thống kê dự báo. Ngoài việc so sánh thì ba lý do trên cũng là lý do dẫn đến việc phát triển phương pháp khác.

Trong các nghiên cứu [11-13] các tác giả đã sử dụng các phương pháp học máy trong trí tuệ nhân tạo như: mô hình logic mờ, mạng nơ-ron nhân tạo, phân tích hồi quy bội, lý luận dựa trên trường hợp, mô hình lai, mô hình mờ di truyền để giải quyết các bài toán khác nhau trong quản lý dự án. Trong các phương pháp học máy được được trình bày thì các tác giả cũng đã nghiên cứu và đề xuất sử dụng thuật toán học máy XGBoost (Extreme Gradient Boosting). Đây là một thuật toán để giải quyết bài toán học có giám sát (supervised learning) cho độ chính xác khá cao bên cạnh các mô hình Deep learing đang rất phổ hiện nay.

Nhận thấy phương pháp học máy XGBoost có nhiều ưu điểm và có khả năng để áp dụng giải quyết bài toán dự đoán chi phí và thời gian hoàn thành dự án, chúng tôi đã nghiên cứu thuật toán XGBoost để giải quyết bài toán dự đoán này. Thay vì chỉ nhận đầu vào là raw data dạng numerical (thường phải chuyển sang dạng n-vector trong không gian số thực) như các mô hình Deep Learning truyền thống thì XGBoost nhận đầu vào là tabular datasets với mọi kích thước và dạng dữ liệu bao gồm cả categorical mà dạng dữ liệu này thường được tìm thấy nhiều hơn trong business model.

Bên cạnh đó, XGBoost có tốc độ huấn luyện nhanh, có khả năng mở rộng để tính toán song song trên nhiều server, có thể tăng tốc bằng cách sử dụng GPU, nhờ vậy mà big data không phải là vấn đề của mô hình này.

Mỗi mô hình ở trên có những ưu điểm và nhược điểm riêng và được áp dụng cho các bộ dữ liệu cụ thể. Trong khuôn khổ bài báo này, chúng tôi tập trung nghiên cứu trên mô hình XGBoost áp dụng trong kỹ thuật EVM trên bộ dữ liệu các dự án trong tài liệu [15] và các dự án thực tế tại Việt Nam để nâng cao chất lượng dự đoán mức độ hoàn thành của dự án.

Nội dung tiếp theo trong bài báo được trình bày như sau: Mục 2 trình bày Cơ sở lý thuyết về quản lý dự án và kỹ thuật EVM và mô hình XGBoost; Mục 3 đề xuất áp dụng thuật toán XGBoost trong kỹ thuật EVM để nâng cao chất lượng dự đoán chi phí và thời gian hoàn thành dự án; Mục 4 trình bày thực nghiệm và đánh giá kết quả và mục 5 trình bày kết luận, đóng góp khoa học và hướng phát triển của nghiên cứu tiếp theo.

2. CƠ SỞ LÝ THUYẾT

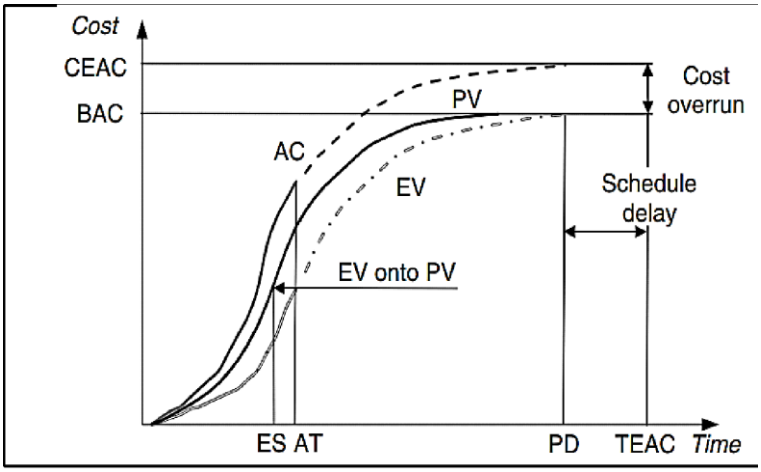
2.1. Phương pháp quản lý giá trị thu được EVM

Những điểm đặc trưng chủ yếu của việc triển khai thực hiện thuật quản lý giá trị thu được bao gồm: một bản kế hoạch dự án (được lập trước khi khởi công) xác định công việc phải hoàn thành, giá trị kinh phí dự kiến, giá trị kinh phí (tức chi phí) theo dự toán (tức là kế hoạch trước khởi công).

EVM là một công cụ hiệu quả được sử dụng để dự đoán thời gian và chi phí hoàn thành dự án dựa vào tình trạng hiện tại của dự án. EVM có các tham số chính như sau [14]:

- PV (Planned Value): Là dự toán ngân quỹ chi phí cho công việc theo tiến độ BCWS (Budgeted Cost of Work Scheduled), đại diện ước tính ban đầu cho công việc theo kế hoạch. Đây là thông tin tiêu biểu cung cấp cho ngân sách của dự án và được tính toán bằng cách ước tính các thành phần liên quan. Một cách rõ ràng hơn, nó được cung cấp bởi các thủ tục đánh giá chi phí theo kế hoạch và được kết hợp với EVM để tăng cường hiệu năng cho bộ công cụ có nhiệm vụ yêu cầu, sắp xếp các công việc liên quan.
- EV (Earned Value): Là chi phí công việc được thực hiện BCWP (Budgeted Cost of Work Performed), đại diện cho lượng công việc đã hoành thành cho đến thời điểm đánh giá, được biểu thực dựa trên ngân sách ban đầu cho công việc đó.
- AC (Actual Cost): Là chi phí thực tế cho công việc đã thực hiện ACWP (Actual Cost of Work Performed), là hao phí thực tế phải bỏ ra để hoàn thành phần công việc, đã được thực hiện xong, vào đúng thời điểm báo cáo.

- ES (Earned Schedule): Thời gian theo kế hoạch.
- BAC (Budget at Completion): Ngân quỹ hoàn thành dự án.



Hình 1. *Các tham số của kỹ thuật quản trị giá trị thu được.*

Hiệu suất dự án về mặt thời gian và chi phí, được xác định bằng cách so sánh các tham số chính PV, AC, EV và ES nhằm đưa ra kết quả đo lường hiệu suất như sau:

- CPI (Cost Performance Index): Chỉ số hiệu suất chi phí. Công thức tính CPI = EV/AC;
- SPI (Schedule Performance Index): Chỉ số hiệu suất kế hoạch. Công thức tính SPI = EV/PV;
- SPI (t): Chỉ số hiệu suất kế hoạch điều chỉnh. Công thức tính SPI(t) = ES/AT;

Dự đoán chi phí hoàn thành dự án (CEAC – Cost Estimate at Completion) được tính theo công thức sau:

$$CEAC = AC + PCWR = AC + \frac{BAC - EV}{PF} \quad (1)$$

Trong đó:

- AC: Chi phí thực tế ở thời điểm hiện tại (tức là thời gian thực tế AT);
- PCWR: Chi phí dự kiến cho các công việc còn lại (là ước lượng cho tương lai);

Cách tính giá trị PCWR phụ thuộc vào hệ số hiệu suất PF (Performance Factor), thể hiện giả định tạo ra cho hiệu suất mong muốn của các công việc trong tương lai, như sau:

- PF = 1: Hiệu suất tương lai được mong đợi dựa trên đường kế hoạch cơ sở;
- PF = CPI: Hiệu suất tương lai được mong đợi dựa trên hiệu suất chi phí hiện tại;
- PF = SPI hoặc SPI(t): Hiệu suất tương lai được mong đợi dựa trên hiệu suất về thời gian hiện tại.
- PF = SCI hoặc SCI(t): Hiệu suất tương lai được mong đợi dựa trên hiệu suất về thời gian và chi phí hiện tại. SCI là chỉ số chi phí kế hoạch, được tính theo công thức SCI = SPI * CPI và SCI(t) = SPI(t) * CPI.

Thời gian dự đoán hoàn thành dự án (TEAC – Time Estimate at Completion) được tính [14]:

$$TEAC = AT + PDWR \quad (2)$$

Trong đó:

- AT: Thời gian hiện tại;
- PDWR: Khoảng thời gian dự kiến của các công việc còn lại, cách tính cũng dựa trên hệ số PE.

Để dự đoán thời gian hoàn thành dự án có thể sử dụng một trong ba phương pháp sau [4]:

- Phương pháp dựa trên PV:

$$TEAC1_{PV} = PD - TV \quad (3)$$

$$TEAC2_{PV} = \frac{PD}{SPI} \quad (4)$$

$$TEAC3_{PV} = \frac{PD}{SCI} \quad (5)$$

- Phương pháp dựa trên ED:

$$ED = AT * SPI$$

$$TEAC_{ED} = AT + \frac{max(PD, AT) - ED}{PF} \quad (6)$$

- Phương pháp dựa trên ES:

$$TEAC_{ES} = AT + \frac{PD - ES}{PF} \quad (7)$$

2.2. Mô hình XGBoost

2.2.1. Tổng quan XGBoost

XGBoost (viết tắt của "Extreme Gradient Boosting") là một mô hình học máy sử dụng kỹ thuật Gradient Boosting dựa trên mô hình cơ bản là "Tập hợp cây quyết định" (Decision Tree Ensembles) để giải quyết các bài toán supervised learning trong học máy. Để hiểu về XGBoost, chúng ta phải làm rõ các khái niệm: (i)"Ensemble method" trong học máy, (ii)kỹ thuật "boosting" và "Gradient boosting", (iii)Decision Tree Ensembles.

2.2.2. Các tham số quan trọng trong mô hình XGBoost

- booster: Sử dụng booster nào, mặc định là "gbtree" tức gradient boosted tree.

- nthread: Số lượng luồng song song được sử dụng để chạy XGBoost.

- verbosity: Độ dài của việc in tin nhắn hệ thống, mặc định thường là 1.

- num_feature: Số lượng feature được sử dụng trong boosting, được đặt bằng số lượng tối đa feature bởi XGBoost b, Các tham số cho "boosting" cây

- gamma: Sự giảm thiểu "loss" tối thiểu cần thiết để tạo một phân vùng sâu hơn trên một nút lá của cây.

- max_depth: Độ sâu tối đa của cây. Việc tăng giá trị này sẽ làm cho mô hình phức tạp hơn và có nhiều khả năng bị overfit. Giá trị mặc định thường là 6.

- min_child_weight: Tổng trọng lượng instance tối thiểu cần thiết ở một nút con. Nếu bước phân vùng cây dẫn đến một nút lá có tổng trọng lượng instance nhỏ hơn min_child_weight, thì quá trình xây dựng sẽ từ bỏ việc phân vùng tiếp theo.

- max_delta_step: Bước delta tối đa mà mô hình cho phép mỗi output tại các nút lá. Nếu giá trị được đặt thành 0, tức là không có ràng buộc. Nếu nó được đặt thành giá trị dương, nó có thể giúp thực hiện bước cập nhật trọng số thận trọng hơn.

- subsample: Tỉ lệ tạo subsample trong quá trình huấn luyện. Nếu đặt thành 0,5 có ý nghĩa là XGBoost sẽ lấy mẫu ngẫu nhiên một nửa dữ liệu huấn luyện trước khi tạo cây. Và điều này sẽ ngăn chặn việc overfit.

- reg_alpha: Sử dụng Chính quy hoá L1 trên trọng số. Việc tăng giá trị này sẽ làm cho mô hình trở nên thận trọng hơn.

- reg_lambda: Sử dụng Chính quy hoá L2 trên trọng số.

- max_bin: Số lượng tối đa các thùng chứa các giá trị của feature được giới hạn.

2.2.3. Tổng kết về XGBoost

XGBoost là một mô hình học máy được phát triển dựa trên thuật toán Gradient Boosting - thuật

toán mạnh mẽ nhất trong Machine Learning. Được cải tiến mạnh mẽ từ cơ sở này, XGBoost trở nên phổ biến, đặc biệt là trong các cuộc thi Machine Learning trên Kaggle. Điều làm nên hiệu suất ấn tượng và khả năng tính toán của XGBoost nằm ở ba yếu tố:

- Kết hợp tốt các kỹ thuật để tránh overfit như: subsample hàng, cột, áp dụng Regularization L1 và L2,...

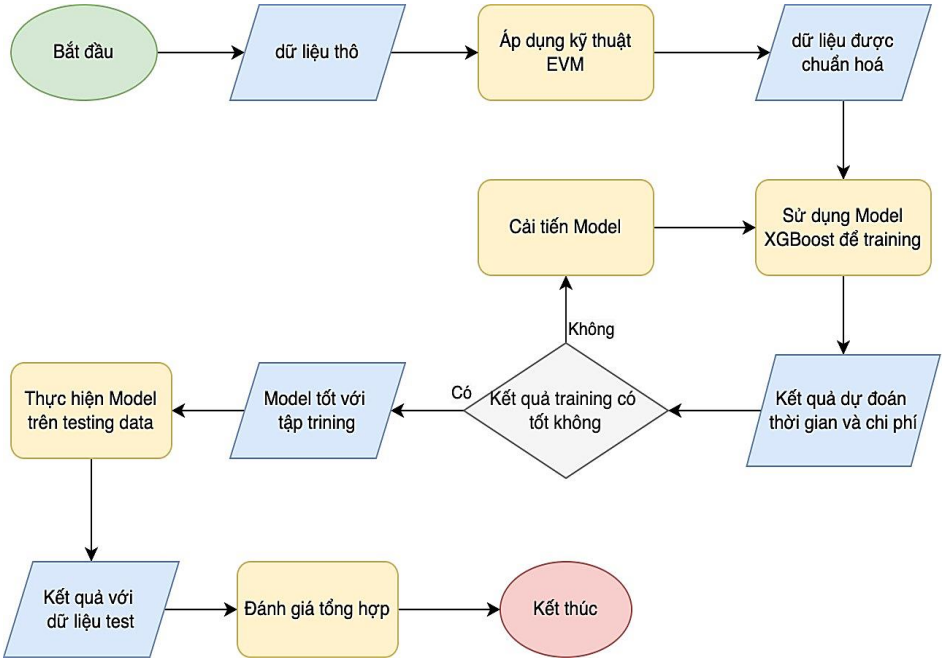
- Khả năng tận dụng tài nguyên hệ thống: tính toán song song trên CPU/GPU, tính toán phân tán trên nhiều server, tính toán khi tài nguyên bị giới hạn, cache optimization để tăng tốc training.

- Cuối cùng là khả năng xử lý missing data value, tiếp tục training bằng mô hình đã được xây dựng trước đó để tiết kiệm thời gian.

3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Tổng quan giải pháp

Từ dữ liệu đầu vào là dữ liệu thô (raw data) của các dự án phần mềm, áp dụng kỹ thuật EVM để đưa dữ liệu về dạng được chuẩn hoá. Với dữ liệu được chuẩn hoá, làm dữ liệu đầu vào cho mô hình học máy (machine learning model) sử dụng thuật toán XGBoost, đưa ra kết quả là các dự đoán về thời gian và chi phí hoàn thành dự án phần mềm. Để trực quan hoá tổng quan giải pháp, dưới đây là lưu đồ luồng giải pháp được thực hiện như sau:



Hình 2. *Lưu đồ luồng giải pháp.*

3.2. Xây dựng mô hình sử dụng XGBoost và EVM cho bài toán Dự đoán thời gian và chi phí hoàn thành dự án

3.2.1. Bài toán dự đoán chi phí và thời gian của dự án

Đầu vào bài toán: Cho các dự án với các thông tin biết trước như sau:

- Kế hoạch dự án (các giá trị dự kiến PV tại các thời điểm báo cáo của dự án từ khi bắt đầu đến khi kết thúc dự án);

- Thời điểm báo cáo dự án AT;

- Chi phí thực tế AC của dự án tại các thời điểm báo cáo từ khi bắt đầu tính đến thời điểm hiện tại;

- Giá trị thu được EV của dự án tại các thời điểm báo cáo từ khi bắt đầu tính đến thời điểm hiện tại.

Đầu ra bài toán: Kết quả dự đoán tổng chi phí và thời gian hoàn thiện dự án.

3.2.2. Xây dựng mô hình

Tiến hành xây dựng mô hình XGBoost theo các bước sau:

- Bước 1: Chuyển đổi các file dữ liệu sang định dạng csv, chia thành 2 tập train và test.
- Bước 2: Tiến hành xây dựng mô hình dựa vào thư viện XGBoost.

Dự đoán TOTAL_AC:

– Input: XT, AC

– Output: TOTAL_AC

- Dự đoán TOTAL_AT:

– Input: XT, AT

– Output: TOTAL_AT

- Bước 3: Điều chỉnh tham số.

Sử dụng thư viện GridSearchCv để tìm ra "best score" và "best params" trong quá trình điều chỉnh tham số. Sau khi thử nghiệm nhiều lần, ta có được bộ "best params" đối với từng model như sau:

Đối với Model dự đoán giá trị TOTAL_AC:

- xgb_grid.best_score: 0.7512836219709136
- xgb_grid.best_params: {
’base_score’: 0.5,
’booster’: ’gbtree’,
’colsample_bylevel’: 1,
’colsample_bynode’: 1,
’colsample_bytree’: 1,
’gamma’: 0,
’importance_type’: ’gain’,
’learning_rate’: 0.05,
’max_delta_step’: 0,
’max_depth’: 5,
’min_child_weight’: 1,
’missing’: None,
’n_estimators’: 1000,
’n_jobs’: 1,
’nthread’: None,
’objective’: ’reg:linear’,
’random_state’: 0,
’reg_alpha’: 0,
’reg_lambda’: 1,
’scale_pos_weight’: 1,
’seed’: None, ’silent’: None,
’subsample’: 0.8,
’verbosity’: 1
}

Đối với Model dự đoán giá trị TOTAL_AT:

- xgb_grid.best_score: 0.7839018869382778
- xgb_grid.best_params: {
’base_score’: 0.5,
’booster’: ’gbtree’,

'colsample_bylevel': 1,
'colsample_bynode': 1,
'colsample_bytree': 1,
'gamma': 1,
'importance_type': 'gain',
'learning_rate': 0.09,
'max_delta_step': 0,
'max_depth': 5,
'min_child_weight': 1,
'missing': None,
'n_estimators': 100,
'n_jobs': 1,
'nthread': None,
'objective': 'reg:linear',
'random_state': 0,
'reg_alpha': 0,
'reg_lambda': 1,
'scale_pos_weight': 1,
'seed': None, 'silent': None,
'subsample': 0.8,
'verbosity': 1
}

- Bước 4: Thử nghiệm trên tập test, lưu lại kết quả dự đoán các dự án.

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Cài đặt và môi trường

- Lập trình trên môi trường: Google Colaboratory.

- Ngôn ngữ lập trình: Python.

- Sử dụng 1 số thư viện hỗ trợ xử lý toán học và thống kê: Numpy, Glob, Pandas; gói thư viện xử lý đồ họa Matplotlib và thư viện hỗ trợ cài đặt, điều chỉnh, lưu trữ Model như Sklearn, XGBoost, Joblib.

4.2. Dữ liệu thực nghiệm

Dữ liệu thực nghiệm được thu thập từ [14] và các dự án từ trang web *projectmanagement.ugent.be*. Dữ liệu thu thập được bao gồm 120 dự án với thông tin về chi phí và thời tiến độ hoàn thành được bao cáo theo các mốc thời gian. Chúng tôi chia bộ dữ liệu thành 2 phần: 110 bộ dùng để huấn luyện và 10 bộ dùng để đánh giá.

4.3. Độ đo sai số tuyệt đối trung bình MAPE

Mape là sai số trung bình của kết quả dự đoán và kết quả thực tế trên toàn bộ tập dữ liệu kiểm thử.

$$MAPE = \frac{\sum_{i=1}^{n}\left(\left|\frac{y_i - f_i}{y_i}\right|\right)}{n} \quad (8)$$

Trong đó:

- *MAPE*: Sai số của kết quả dự đoán so với kết quả thực tế;

- y_i: Chi phí (thời gian) thực tế;

- f_i: Kết quả dự đoán chi phí (thời gian) của model;

- n: Số lượng dự án tiến hành kiểm thử.

4.4. Kết quả thực nghiệm

4.4.1. Dự đoán chi phí hoàn thành dự án

Bảng 1. *Kết quả dự đoán chi phí hoàn thành dự án.*

Dự án	Chi phí thực tế	Dự đoán tại thời điểm 25%	Sai số (%)	Dự đoán tại thời điểm 50%	Sai số (%)	Dự đoán tại thời điểm 75%	Sai số (%)
DA 1	2563675,3	3450390,25	34,59	1980104,88	22,76	2546159,5	0,68
DA 2	2512524	2055766	18,18	1898775,38	24,43	2451478,75	2,43
DA 3	955929,2	776732,31	18,75	988522,56	3,41	988522,56	3,41
DA 4	175030,7	274502,94	56,83	284032,19	62,28	284032,19	62,28
DA 5	2590796,7	3450390,25	33,18	3450390,25	33,18	2173816	16,09
DA 6	186107	218371,66	17,34	284032,19	52,62	284032,19	52,62
DA 7	1868796,3	864973,88	53,71	1158857,25	37,99	1737244,38	7,04
DA 8	308343,8	519852,75	68,6	533800,69	73,12	533800,69	73,12
DA 9	967988,8	1055922,88	9,08	697695,31	27,92	1003905,13	3,71
DA10	646473,6	385749	40,33	568628,38	12,04	728710	12,72
Sai số trung bình MAPE			35,05		34,97		23,41

Kết quả thực nghiệm cho thấy sai số dự đoán chi phí hoàn thành theo phương pháp học máy XGBoost tại các giai đoạn sớm (25%), giai đoạn giữa (50%), giai đoạn muộn (75%) của 10 dự án là từ 73,12 đến 0,68. Nhiều dự án thì tại giai đoạn muộn của dự án phương pháp này cho kết quả dự đoán chi phí hoàn thành khá chính xác, tuy nhiên, còn có những dự án phương pháp học máy XGBoost cho kết quả có sai số quá lớn. Sai số trung bình của phương pháp XGBoost tại giai đoạn muộn của dự án là nhỏ nhất (23,41%) và tại giai đoạn sớm của dự án là lớn nhất (35,05%).

4.4.2. Dự đoán thời gian hoàn thành dự án

Bảng 2. *Kết quả dự đoán thời gian hoàn thành dự án.*

Dự án	Thời gian thực tế (tháng)	Dự đoán tại thời điểm 25%	Sai số (%)	Dự đoán tại thời điểm 50%	Sai số (%)	Dự đoán tại thời điểm 75%	Sai số (%)
DA 1	6,08	7,45	22,63	6,94	14,23	8,24	35,49
DA 2	21,08	21,02	0,29	20,25	3,95	19,98	5,19
DA 3	10,08	8,22	18,43	7,9	21,65	10,49	4,08
DA 4	10,95	7,74	29,25	9,27	15,28	11,46	4,67
DA 5	9,95	8,22	17,33	10,88	9,34	11,18	12,39
DA 6	13,68	9,3	32,04	11,3	18,18	12,43	9,16
DA 7	12,91	14,17	9,71	13,32	3,19	13,34	3,34
DA 8	5,52	8,87	60,90	4,66	15,46	5,29	4,05
DA 9	23,35	20,33	12,91	27,47	17,66	29,66	27,05
DA10	9,78	12,6	28,85	12,17	24,47	11,39	16,50
MAPE			23,23		14,34		12,19

Kết quả thực nghiệm dự đoán thời gian hoàn thành dự án theo phương pháp học máy XGBoost tại các giai đoạn sớm (25%), giai đoạn giữa (50%), giai đoạn muộn (75%) của 10 dự án là từ 60,9 đến 0,29. Nhiều dự án có kết quả dự đoán thời gian hoàn thành khá chính xác, tuy nhiên, đặc biệt vẫn có dự án dự đoán còn sai đến 60,9%. Sai số trung bình của phương pháp XGBoost tại giai đoạn muộn của dự án là nhỏ nhất (12,19%) và tại giai đoạn sớm của dự án là lớn nhất (23,23%).

5. KẾT LUẬN

Nhìn chung, kết quả thực nghiệm cho chúng ta thấy tiềm năng của việc ứng dụng thuật toán XGBoost nói riêng và các kĩ thuật học máy nói chung trong bài toán dự đoán chi phí và thời gian hoàn thành dự án phần mềm. Sử dụng phương pháp học máy XGBoost để dự đoán thời gian hoàn thành dự án cho kết quả chính xác hơn sử dụng để dự đoán chi phí hoàn thành dự án. Cụ thể, đối với mô hình dự đoán chi phí, độ chính xác của mô hình chỉ dao động trong khoảng từ 65% đến 75%. Trong khi đó, độ chính xác của mô hình dự đoán thời gian lại cao hơn rõ rệt, từ khoảng 76% đến 87%. Điều này có thể do bản chất của chính "thời gian" và "chi phí". Trên thực tế, việc dự đoán chính xác con số chi phí hoàn thành dự án rất khó khăn, còn đối với "thời gian" sẽ dễ dàng hơn và cho kết quả có khả năng chính xác cao hơn.

Trên thực tế, các dữ liệu thu thập được đều là dữ liệu thực của các doanh nghiệp tự nguyện cung cấp dưới dạng open source, lượng dữ liệu để huấn luyện còn tương đối nhỏ làm ảnh hưởng đến độ chính xác của mô hình. Trong tương lai, chúng tôi sẽ cố gắng thu thập thêm dữ liệu và cải tiến thuật toán để xây dựng được mô hình dự đoán chuẩn xác hơn.

TÀI LIỆU THAM KHẢO

[1]. Simion, Cezar-Petre, and Irinel Marin. *"Project cost estimate at completion: earmed value management versus earned schedule-based regression models. A comparative analysis of the models application in the construction projects in Romania"*. Economic Computation & Economic Cybernetics Studies & Research 52.3, (2018).

[2]. Nannini, G., R.D.H, Warburton, and A. De Marco. *"Improving the accuracy of project estimates at completion using the Gompertz function"*. International Research Network on Organizing by Projects (IRNOP), UTS ePRESS, Sydney: NSW, pp.1-15, (2017).

[3]. Narbaev T .; De Marco A. *"Combination of Growth Model and Earned Schedule to Forecast Project Cost at Completion"*. In: Journal of Construction engineering and management, vol. 140 n. 1, Article number 04013038-. - ISSN 0733-9364, (2014).

[4]. Batselier, J., & Vanhoucke, M. *"Evaluation of deterministic state-of-the-art forecasting approaches for project duration based on earning value management"*. International Journal of Project Management, 33 (7), 1588-1596, (2015).

[5]. Khamooshi, H., Golafshani, H. EDM: *"Earned Duration Management, a new approach to schedule performance management and measurement"*. Int. J. Proj. Manag. 32, 1019–1041, (2014).

[6]. Quentin WeFleming, Joel MeKoppelman. *"Earned Value Project Management"*. Project Management Institute Newtown Square, Pennsylvania USA, (2015).

[7]. Elshaer, R. *"Impact of sensitivity information on the prediction of project's duration using earning schedule method"*. International Journal of Project Management, 31 (4), 579-588, (2013).

[8]. Mukherjee, I., & Routroy, S. *"Comparing the performance of neural networks developed by using Levenberg – Marquardt and Quasi-Newton with the gradient descent algorithm for modeling a multiple response grinding process"*. Expert Systems with Applications, 39 (3), 2397-2407, (2012).

[9]. Bottou, L. *"Large-scale machine learning with stochastic gradient descent"*. In Proceedings of COMPSTAT'2010 (pp. 177-186). Physica-Verlag HD, (2010).

[10]. D. Dong, and T. McAvoy, *"Nonlinear principal component analysis — based on principal curves and neural networks",* Computers & Chemical Engineering, vol. 20, no. 1, pp. 65-78, (1996).

[11]. R. S. FAN, Y. LI, and T. T. MA, *"Research and application of project settlement overdue prediction based on xgboost intelligent algorithm",* in Sustainable Power and Energy Conference (iSPEC), IEEE, pp. 1213-1216, (2019).

[12]. H. H. Elmousalami, *"Comparison of artificial intelligence techniques for project conceptual cost prediction: A case study and comparative analysis",* IEEE Transactions on Engineering Management, vol. 68, no. 1, pp. 183-196, (2021).

[13]. YAN, Hongyan, et al, *"Investment estimation of prefabricated concrete buildings based on XGBoost machine learning algorithm"*, Advanced Engineering Informatics, vol 54, no. c: 101789, (2022).

[14]. L. T. Anh, N. T. Hung, H. Q. Thang, and N. V. Can, *"Calibrating the future performance factor PF in the EVM-GM method of evaluating software project completion: testing and evaluation",* in National

Conference XXI: Some selected issues of Information and Communications Technology, Thanh Hoa, Vietnam, pp. 137-143, (2018).

[15]. Batselier, J., Vanhoucke, M., available at: http://www.projectmanagement.ugeunt.be/research/data/realdata

[16]. Ruder, Sebastian, *"An overview of gradient descent optimization algorithms"*, (2016).

ABSTRACT

Predict software project completion time and cost using XGBoost

Nowadays, with the rapid development of information technology, managing costs and time to complete software projects has become an urgent issue. To be able to manage software projects, the need to predict costs and completion times is extremely important. Traditional methods often use EVM earned value management to predict project costs and completion times. However, this method often does not achieve very high accuracy when the data has a lot of noise. In recent years, machine learning methods have emerged as a useful solution for leveraging past data to predict future values. In this study, we propose to use the XGBoost machine learning model to predict project costs and completion time. Experimental results show that XGBoost has the potential to solve this problem.

Keywords: Software project management; EVM; XGBoost.