Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.000106

XÂY DỰNG VÀ CÂN CHỈNH MÔ HÌNH DỰ BÁO MẬT SỐ RẦY NÂU TRÊN NỀN APACHE SPARK

Đỗ Thanh Nghị, Trần Nguyễn Minh Thư, Bùi Võ Quốc Bảo, Phạm Nguyên Khang
Khoa CNTT-TT, Trường Đại học Cần Thơ
Khu 2, Đường 3/2, Xuân Khánh, Ninh Kiều, TP. Cần Thơ
dtnghi@cit.ctu.edu.vn

TÓM TẮT— *Trong bài viết này, chúng tôi trình bày tiếp cận xây dựng mô hình dự báo dịch rầy nâu gây hại trên lúa. Mô hình máy học véc-tơ hỗ trợ và rừng ngẫu nhiên là các mô hình được sử dụng phổ biến trong dự báo do tính chính xác của chúng. Tuy nhiên, việc cân chỉnh mô hình để tìm các siêu tham số của giải thuật máy học tốn nhiều thời gian tính toán. Chúng tôi đề xuất phân tán các tác vụ cân chỉnh mô hình trên nền Apache Spark (nền tảng tính toán nhóm trên bộ nhớ trong), để rút ngắn thời gian tìm kiếm các siêu tham số của giải thuật học khi xây dựng mô hình dự báo mật số rầy nâu. Kết quả thực nghiệm cho thấy rằng phân tán công việc cân chỉnh mô hình dự báo của máy học véc-tơ hỗ trợ, rừng ngẫu nhiên trên nền Apache Spark đạt hiệu quả về thời gian khi tăng số lượng nút sử dụng trong hệ nhóm máy tính. Kết quả của mô hình tối ưu tìm được sau khi cân chỉnh mô hình dự báo chính xác mật số rầy nâu khi so sánh với các mô hình hồi quy tuyến tính, k láng giềng.*

Từ khóa— *Dự báo mật số rầy nâu, máy học véc-tơ hỗ trợ, rừng ngẫu nhiên, Apache Spark.*

I. GIỚI THIỆU

Vùng đồng bằng sông Cửu Long từ lâu được xem là trung tâm lớn về sản xuất lúa gạo, nuôi trồng, đánh bắt và chế biến thủy sản, đóng góp lớn vào xuất khẩu nông thủy sản của cả nước. Theo báo Quân đội nhân dân số ra ngày 2 tháng 1 năm 2015, nguồn tin từ Ban Chỉ đạo Tây Nam Bộ cho biết, trong năm 2014, các tỉnh vùng Đồng bằng sông Cửu Long (ĐBSCL) phấn đấu nâng kim ngạch xuất khẩu gạo và thủy sản lên 10,2 tỷ USD, tăng trên 21% so với năm 2014, góp phần đưa tổng kim ngạch xuất khẩu hàng hóa của vùng trong năm 2015 đạt 11,9 tỷ USD. Các tỉnh ĐBSCL sẽ thực hiện các biện pháp ổn định diện tích sản xuất lúa 4,2 triệu héc-ta (trong đó, 80% diện tích trồng giống lúa chất lượng cao) và 800.000 héc-ta thủy sản để phấn đấu đạt sản lượng 25 triệu tấn lúa và 3,7 triệu tấn thủy sản phục vụ tiêu dùng trong nước và chế biến xuất khẩu. Kinh tế vùng đóng vai trò rất lớn trong phát triển kinh tế của nước ta. Khi kinh tế xã hội phát triển thì cũng đi theo đó là nạn tàn phá môi trường tự nhiên, ô nhiễm, do biến đổi điều kiện khí hậu, gây ra không ít khó khăn tác động trực tiếp đến sản xuất của bà con nông dân. Theo Tạp chí cộng sản số ra ngày 29 tháng 10 năm 2013, Việt Nam được Liên hợp quốc xác định là một trong sáu quốc gia trên thế giới chịu tác động nhiều nhất của tình trạng biến đổi khí hậu toàn cầu. Trong đó, đồng bằng sông Cửu Long được xác định là một trong những vùng của Việt Nam và thế giới chịu tác động và thiệt hại nặng nề nhất do tình trạng biến đổi khí hậu và nước biển dâng. Tình trạng nước biển xâm nhập ngày càng sâu vào đất liền, làm nhiều diện tích lúa bị nhiễm mặn. Dịch bệnh phát triển trên diện rộng như dịch rầy nâu làm phá hoại lúa, tôm cá chết hàng loạt do bị nhiễm bệnh hay do tác động xấu của môi trường. Tình hình dịch hại ảnh hưởng rất lớn đến nguồn lợi kinh tế của bà con nông dân và cũng ảnh hưởng đến phát triển kinh tế, an ninh lương thực của vùng.

Chính vì lý do trên, xây dựng mô hình phục vụ công tác dự báo tình hình dịch hại rất cần thiết. Mục tiêu chính là giúp nhà nông tránh được rủi ro trong sản xuất, kịp thời ứng phó với dịch hại, bảo vệ nguồn lợi kinh tế. Nghiên cứu của [Trương et al., 11] đề xuất sử dụng công nghệ GIS và mô hình hồi quy tuyến tính để dự báo dịch rầy nâu ở Đồng Tháp. [Vũ & Huỳnh, 16] sử dụng mô hình mạng Bayes và xích Markov để dự báo mức độ nhiễm, cháy và lan truyền rầy theo thời gian. [Võ & Trần, 14], [Võ et al., 15] đề xuất ứng dụng ảnh viễn thám xác định hiện trạng sinh trưởng cây lúa cảnh báo dịch hại tỉnh tại An Giang. [Nguyễn, 16] nghiên cứu hệ thống đa tác tử và mô hình hóa khả năng ra quyết định dựa vào nhiều tiêu chí trong đánh giá rủi ro côn trùng hại lúa.

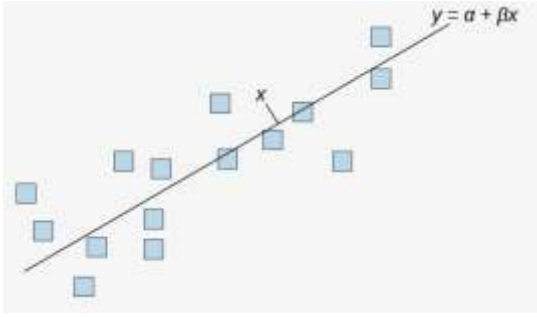
Trong phạm vi của nghiên cứu này, chúng tôi trình bày kết quả thu được từ việc áp dụng công nghệ khám phá tri thức và khai mở dữ liệu [Fayyad et al., 96] trong phân tích và dự báo mật số rầy nâu gây hại trên lúa. Chúng tôi tiến hành điều tra thu thập số liệu từ các mùa vụ trước, thực hiện các thao tác tiền xử lý và làm sạch dữ liệu. Bước tiếp theo thực hiện xây dựng mô hình phi tuyến, máy học véc-tơ hỗ trợ (Support Vector Machines – SVM [Vapnik, 1995]), rừng ngẫu nhiên (Random Forests – RF [Breiman, 01]), để dự báo mật số rầy nâu. Chúng tôi đề xuất phân tán các tác vụ cân chỉnh mô hình dự báo trên nền tảng tính toán nhóm trên bộ nhớ trong, Apache Spark [Zaharia et al., 10], [Apache Software Foundation, 14]. Kết quả thực nghiệm cho thấy rằng phân tán công việc cân chỉnh mô hình dự báo của máy học véc-tơ hỗ trợ, rừng ngẫu nhiên trên nền Apache Spark đạt hiệu quả về thời gian khi tăng số lượng nút sử dụng trong hệ nhóm máy tính. Kết quả của mô hình tối ưu tìm được sau khi cân chỉnh mô hình dự báo chính xác mật số rầy nâu khi so sánh với các mô hình hồi quy tuyến tính [Hastie et al., 01], k láng giềng [Fix & Hodges, 52].

Phần còn lại của bài viết được tổ chức như sau: phần 2 trình bày tóm tắt về các mô hình dự báo mật số rầy nâu; phần 3 trình bày cân chỉnh mô hình với Apache Spark; kết quả thực nghiệm được trình bày trong phần 4 trước khi kết luận và hướng phát triển được trình bày trong phần 5.

II. CÁC MÔ HÌNH DỰ BÁO

Hồi quy là phương pháp toán học được áp dụng thường xuyên trong thống kê để phân tích mối liên hệ giữa các hiện tượng kinh tế xã hội. Xét tập dữ liệu gồm m phần tử $x_1, x_2, ..., x_m$ trong không gian n chiều (biến độc lập, thuộc tính), có giá trị tương ứng của biến phụ thuộc (cần dự báo) là $y_1, y_2, ..., y_m$. Phân tích hồi quy là phân tích thống kê để xác định mối quan hệ giữa biến phụ thuộc y với một hay nhiều biến độc lập x.

A. Hồi quy tuyến tính



Hình 1. Hồi quy tuyến tính

Hồi quy tuyến tính được sử dụng rộng rãi trong thực tế do tính đơn giản. Mô hình hồi quy tuyến tính mô tả mối quan hệ tuyến tính giữa biến phụ thuộc y với một hay nhiều biến độc lập x. Mô hình hồi quy tuyến tính có dạng:

$$y = \alpha + \beta x \tag{1}$$

với α là chặn (intercept), β là độ dốc (slope)

Các tham số α, β của mô hình được ước lượng từ dữ liệu quan sát (tập dữ liệu huấn luyện) bằng phương pháp bình phương bé nhất (least squares):

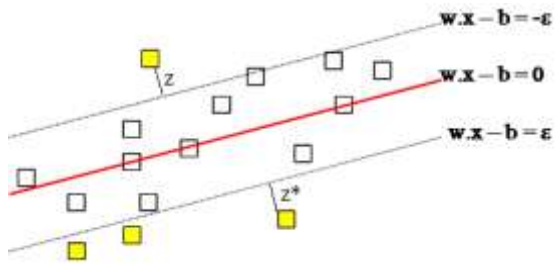
$$Min\left(\sum_{i=1}^{m}[y_i - (\alpha + \beta x_i)]^2\right) \tag{2}$$

Giá trị dự báo cho phần tử mới x dựa vào công thức (3):

$$\hat{y} = \alpha + \beta x \tag{3}$$

B. Máy học véc-tơ hỗ trợ

Máy học véc-tơ hỗ trợ (SVM) được đề xuất bởi Vapnik từ năm 1995 là mô hình học hiệu quả và phổ biến cho vấn đề phân lớp, hồi quy tuyến tính và phi tuyến. Xét bài toán hồi quy như hình 2.



Hình 2. Máy học véc-tơ hỗ trợ cho vấn đề hồi quy

Giải thuật SVM tìm siêu phẳng tối ưu (xác định bởi véc-tơ pháp tuyến w và độ lệch của siêu phẳng b), đi qua tất cả các phần tử dữ liệu với độ lệch chuẩn là ε (dựa trên 2 siêu phẳng hỗ trợ, $w.x - b = \varepsilon$ và $w.x - b = -\varepsilon$). Những phần tử nằm phía ngoài siêu phẳng hỗ trợ được coi như lỗi. Khoảng cách lỗi được biểu diễn bởi $z_i \geq 0$ (với x_i nằm phía trong của 2 siêu phẳng hỗ trợ của nó thì khoảng cách lỗi tương ứng $z_i = 0$, còn ngược lại thì $z_i > 0$ là khoảng cách từ điểm x_i đến siêu phẳng hỗ trợ tương ứng của nó). Huấn luyện máy học SVM cho xử lý vấn đề hồi quy dẫn đến việc giải bài toán quy hoạch toàn phương (4) như sau:

$$min\ \Psi(w, b, z^*, z) = (1/2)\ ||w||^2 + c\sum_{i=1}^{m}(z_i^* + z_i)$$

$$s.t. \tag{4}$$

$$w.x_i - b - y_i - z_i^* \leq \varepsilon$$

$$w.x_i - b - y_i + z_i \geq -\varepsilon$$

$z_i^*, z_i \geq 0$ *(i=1, 2, …, m)*

với hằng c > 0 được sử dụng để chỉnh độ rộng lề và lỗi.

Giải bài toán quy hoạch toàn phương (4) sẽ thu được siêu phẳng hồi quy (*w, b*) của SVM. Dự báo cho phần tử mới đến *x* dựa trên siêu phẳng (*w, b*) được tính theo công thức (5):

$$predict(x) = (w.x - b) \quad (5)$$

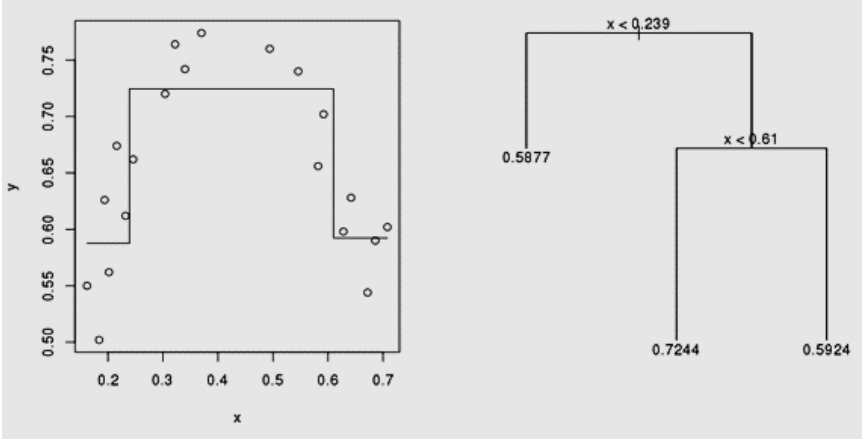
Máy học SVM có thể sử dụng các hàm nhân khác nhau để giải quyết lớp các bài toán phân lớp phi tuyến [Cristianini & Shawe-Taylor, 00]. Để xử lý các vấn đề phân lớp phi tuyến, không cần bất kỳ thay đổi nào hơn từ giải thuật mà chỉ cần thay thế hàm nhân tuyến tính trong công thức bằng các hàm nhân khác. Có 2 hàm nhân phi tuyến phổ biến là:

- Hàm đa thức bậc d: $K\langle x_i, x_j\rangle = \left(\langle x_i \times x_j\rangle + 1\right)^d$ (6)
- Hàm cơ sở bán kính (Radial Basic Function – RBF): $K\langle x_i, x_j\rangle = e^{-g\|x_i - x_j\|^2}$ (7)

Mô hình máy học SVM cho kết quả cao, ổn định, chịu đựng nhiễu tốt và phù hợp với các bài toán phân lớp, hồi quy. Nhiều ứng thành công của SVM đã được công bố trong nhiều lĩnh vực như nhận dạng ảnh, phân loại văn bản và sinh-tin học [Guyon, 99].

C. Rừng ngẫu nhiên

Cây quyết định đề xuất bởi [Breiman et al., 84], [Quinlan, 93] là mô hình máy học tự động sử dụng rất nhiều trong phân tích dự báo và khai mở dữ liệu do tính đơn giản và hiệu quả. Hình 6 minh họa một ví dụ của cây quyết định thu được bằng cách học từ tập dữ liệu, để dự đoán giá trị biến phụ thuộc y từ biến x. Mô hình rất dễ hiểu bởi vì chúng ta có thể rút trích luật quyết định tương ứng với nút lá có dạng IF-THEN được tạo ra từ việc thực hiện AND trên các điều kiện theo đường dẫn từ nút gốc đến nút lá. Các luật quyết định dễ hiểu với người sử dụng.



Hình 3. Mô hình cây quyết định cho vấn đề hồi quy

Giải thuật học từ dữ liệu là quá trình xây dựng cây bắt đầu từ nút gốc đến nút lá. Giải thuật thực hiện phân hoạch đệ quy tập dữ liệu theo các biến độc lập thành các phân vùng siêu chữ nhật rời nhau mà ở đó các phần tử dữ liệu $x_i, x_j, \ldots, x_k$ của cùng phân vùng (nút lá) có các $y_i, y_j, \ldots, y_k$ là tương tự nhau trong vấn đề hồi quy. Giải thuật học mô hình cây quyết định từ dữ liệu gồm 2 bước lớn: xây dựng cây, cắt nhánh để tránh học vẹt. Quá trình xây dựng cây được làm như sau:

- Bắt đầu từ nút gốc, tất cả các dữ liệu học ở nút gốc,

- Nếu các phần tử dữ liệu tại 1 nút là tương tự nhau thì nút đang xét được cho là nút lá, giá trị dự báo của nút lá chính là giá trị trung bình của các $\{y_i, \ldots, y_k\}$ của các phần tử trong nút lá.

- Nếu dữ liệu ở nút quá hỗn loạn (các giá trị $\{y_i, \ldots, y_k\}$ rất khác nhau) thì nút được cho là nút trong, tiến hành phân hoạch dữ liệu một cách đệ quy bằng việc chọn 1 biến để thực hiện phân hoạch tốt nhất có thể.

Một biến được cho là tốt được sử dụng để phân hoạch dữ liệu sao cho kết quả thu được cây nhỏ nhất. Việc lựa chọn này dựa vào các heuristics: chọn biến sinh ra các nút lá sớm nhất. Để đánh giá và chọn biến khi phân hoạch dữ liệu, giải thuật CART của [Breiman et al., 84] ước lượng độ đo hỗn loạn thông tin tại phân vùng *D* dựa trên độ lệch chuẩn như trong (8) với μ là giá trị trung bình của các giá trị *y* trong *D*.

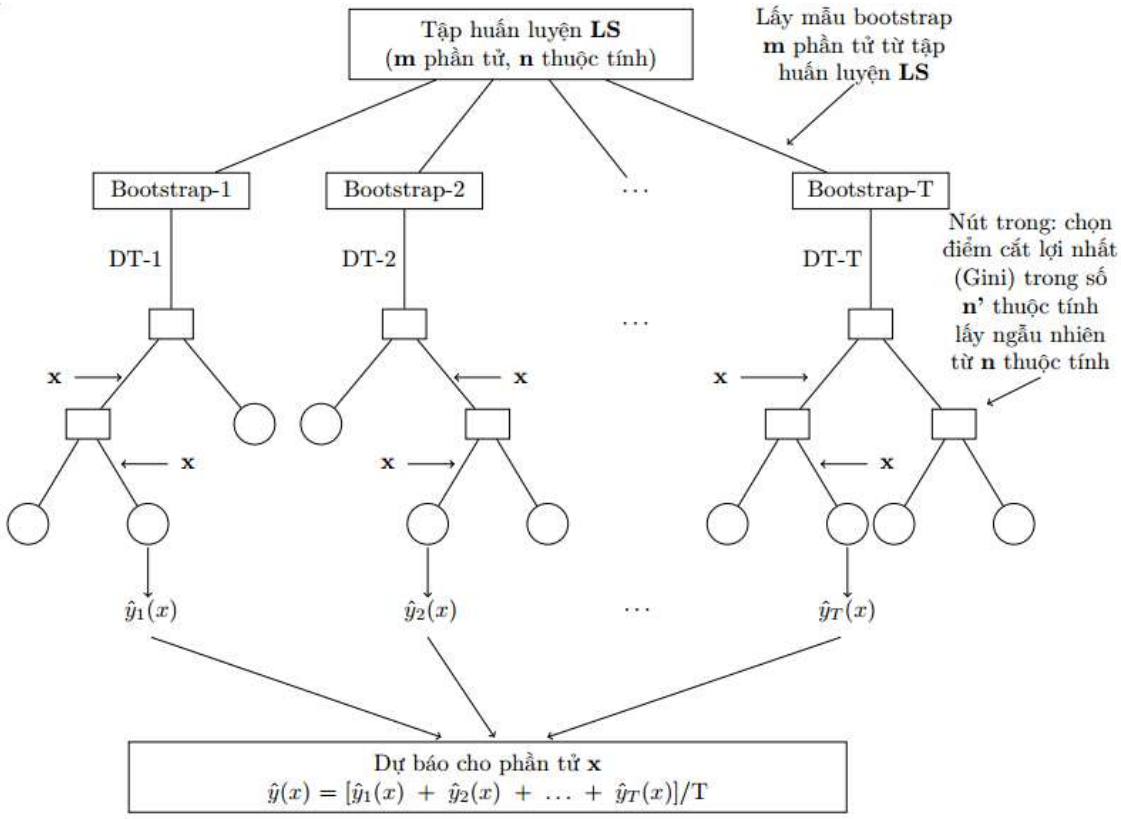
$$S(D) = \sum_{i=1}^{k} \frac{(y_i - \mu)^2}{k} \tag{8}$$

Nếu sử dụng biến A phân hoạch D kích thước m thành 2 tập con D_1 (kích thước m_1) và D_2 (kích thước m_2), độ hỗn loạn sau khi phân hoạch được tính như công thức (9):

$$S_A(D) = \frac{m_1}{m} S(D_1) + \frac{m_2}{m} S(D_2) \tag{9}$$

Biến được chọn phân hoạch dữ liệu là biến cho giá trị độ hỗn loạn sau khi phân hoạch là nhỏ nhất.

Mô hình cây quyết định sau khi xây dựng thường không mạnh với nhiễu và dễ dẫn đến học vẹt. Tức là mô hình có tính tổng quát thấp, chỉ cần dữ liệu kiểm tra có thay đổi một ít so với dữ liệu học thì cây quyết định dự báo sai. Để khắc phục khuyết điểm này, Breiman cũng đề nghị các chiến lược cắt nhánh trong giải thuật CART. Có 2 lựa chọn hoặc postpruning (cắt nhánh cây sau khi xây dựng cây) hay prepruning (dừng sớm quá trình phân nhánh). Trong thực tế, postpruning được sử dụng nhiều hơn prepruning. Tuy nhiên độ phức tạp của việc cắt nhánh sau khi xây dựng cây rất phức tạp, sử dụng các chiến lược để ước lượng lỗi sinh ra bởi mô hình sau khi cắt nhánh.



Hình 4. Mô hình rừng ngẫu nhiên cho vấn đề hồi quy

Trong phân tích thành phần lỗi của giải thuật học, Breiman đã chỉ ra trong [Breiman, 96], lỗi bao gồm 2 thành phần là bias và variance. Thành phần lỗi bias là khái niệm về lỗi của mô hình học (không liên quan đến dữ liệu học) và thành phần lỗi variance là lỗi do tính biến thiên của mô hình so với tính ngẫu nhiên của các mẫu dữ liệu học. Dựa trên cách phân tích hiệu quả của giải thuật học, Breiman đã đề xuất giải thuật học rừng ngẫu nhiên [Breiman, 01], tạo ra một tập hợp các cây quyết định không cắt nhánh, mỗi cây được xây dựng trên tập mẫu bootstrap (lấy mẫu có hoàn lại từ tập học), tại mỗi nút phân hoạch tốt nhất được thực hiện từ việc chọn ngẫu nhiên một tập con các thuộc tính. Lỗi tổng quát của rừng phụ thuộc vào độ chính xác của từng cây thành viên trong rừng và sự phụ thuộc lẫn nhau giữa các cây thành viên. Giải thuật rừng ngẫu nhiên xây dựng cây không cắt nhánh nhằm giữ cho thành phần lỗi bias thấp (thành phần lỗi bias là thành phần lỗi của giải thuật học, nó độc lập với tập dữ liệu học) và dùng tính ngẫu nhiên để điều khiển tính tương quan thấp giữa các cây trong rừng. Giải thuật máy học rừng ngẫu nhiên (hình 4) có thể được trình bày ngắn gọn như sau:

- Từ tập dữ liệu học *LS* có *m* phần tử và *n* biến (thuộc tính), xây dựng *T* cây quyết định một cách độc lập nhau
- Mô hình cây quyết định thứ *t* được xây dựng trên tập mẫu Bootstrap thứ *t* từ tập học *LS*
- Tại nút trong, chọn ngẫu nhiên *n'* biến (*n'*<<*n*) và tính toán phân hoạch tốt nhất dựa trên *n'* biến này
- Cây được xây dựng đến độ sâu tối đa không cắt nhánh.

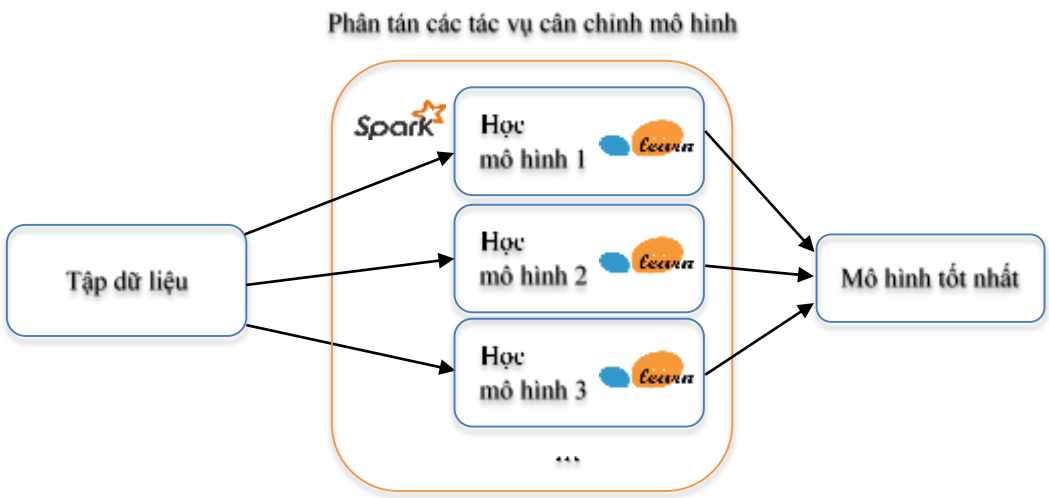
Kết thúc quá trình xây dựng T mô hình cơ sở, kết quả dự báo một phần tử mới đến x, chính là giá trị trung bình dự báo các mô hình cơ sở trên x.

III. CÂN CHỈNH MÔ HÌNH VỚI APACHE SPARK

Như trình bày trong [Breiman, 01], rừng ngẫu nhiên học nhanh, chịu đựng nhiễu tốt và không bị tình trạng học vẹt. Mô hình rừng ngẫu nhiên dự báo với độ chính xác cao khi so sánh với các thuật toán học có giám sát hiện nay như máy học SVM [Vapnik, 95], Adaboost [Freund & Schapire, 99]. Chính vì lý do đó, máy học véc-tơ hỗ trợ và rừng ngẫu nhiên được sử dụng phổ biến trong cộng đồng khám phá tri thức và khai thác dữ liệu [Wu & Kumar, 09]. Tuy nhiên, để có được mô hình dự báo tốt, các nhà phân tích số liệu cần phải thực hiện bước cân chỉnh mô hình trong quá trình huấn luyện mô hình dự báo.

Một mô hình máy học thường chịu sự tác động của nhiều tham số. Với trường hợp của mô hình máy học véc-tơ hỗ trợ, quá trình huấn luyện cần điều chỉnh 2 siêu tham số là tham số của hàm nhân và hằng $c > 0$ được sử dụng để chỉnh độ rộng lề và lỗi. Tương tự, quá trình huấn luyện mô hình dự báo rừng ngẫu nhiên cũng cần điều chỉnh 4 tham số như: số thuộc tính ngẫu nhiên được sử dụng để tính phân hoạch tại nút trong của cây quyết định, số phần tử ít nhất để thực hiện phân hoạch tại nút trong (điều kiện dừng sớm quá trình phân hoạch của cây quyết định), độ sâu tối đa của cây quyết định và tổng số cây cần xây dựng trong rừng.

Rất khó biết được giá trị các tham số là bao nhiêu được sử dụng để có thể thu được mô hình tốt nhất nếu không thực hiện thử sai nhiều giá trị khác nhau cho các tham số. Quá trình cân chỉnh mô hình dựa trên tập các giá trị khác nhau của các tham số, mỗi bộ tham số là tổ hợp của các tham số, được dùng để xây dựng một mô hình trên tập dữ liệu huấn luyện và đánh giá kết quả dự báo trên tập dữ liệu kiểm tra. Mô hình có kết quả kiểm tra tốt nhất sẽ được lựa chọn. Tiến trình cân chỉnh mô hình thường mất rất nhiều thời gian đặc biệt là khi mô hình có độ phức tạp cao (nhiều tham số) hoặc dữ liệu học lớn, khi phải xử lý trên một máy tính đơn. Tuy nhiên, chúng ta có thể thấy rằng nhiều tác vụ trong kịch bản điều chỉnh tham số cho mô hình máy học là độc lập. Đây là điều kiện lý tưởng cho việc thực hiện song song các tác vụ này. Chúng tôi đề xuất phân tán các tác vụ cân chỉnh mô hình trên nền tảng tính toán nhóm trên bộ nhớ trong Apache Spark [Zaharia et al., 10], [Apache Software Foundation, 14] và thư viện spark-sklearn [Hunter & Bradley, 16], để rút ngắn thời gian tìm kiếm các tham số tối ưu của giải thuật học khi xây dựng mô hình dự báo.



Hình 5. Phân tán các tác vụ cân chỉnh mô hình trên nền Spark cluster sử dụng spark-sklearn

Thư viện spark-sklearn cho phép phân phối tải công việc cho một cụm máy Spark. Mỗi máy tính trong cụm thực hiện giải thuật học để huấn luyện mô hình dự báo sử dụng các bộ tham số và gửi trả về kết quả thu được tương ứng với từng bộ tham số. Nhờ đó, chúng ta có thể chọn được bộ tham số tối ưu cho mô hình dự báo. Hình 5 minh họa quá trình cân chỉnh mô hình trên cụm máy tính Apache Spark.

IV. KẾT QUẢ THỰC NGHIỆM

Để tiến hành đánh giá hiệu quả của các mô hình dự báo mật số rầy nâu, chúng tôi tiến hành cài đặt tất cả các chương trình dự báo bằng ngôn ngữ trong ngôn ngữ Python có sử dụng gói thư viện Scikit-learn [Pedregosa et al., 2011]. Thư viện Scikit-learn cung cấp các giải thuật để xây dựng mô hình hồi quy tuyến tính (LM [Hastie et al., 01]), k láng giềng (kNN [Fix & Hodges, 52]), máy học véc-tơ hỗ trợ cho hồi quy (SVR), rừng ngẫu nhiên (RF).

Chúng tôi cài đặt 1 nhóm gồm 4 máy tính, trong đó có 3 máy tính PC, CPU Intel Core i5-4570 3.2 GHz (4 core), 4 GB RAM và 1 máy tính PC, CPU Intel Core i7-4790, 3.6 GHz (4 core), 16 GB RAM. Tất cả các máy đều cài đặt hệ điều hành Linux (Ubuntu 14.04 LTS), cài đặt nền Apache Spark [Zaharia et al., 10], [Apache Software Foundation, 14], gói thư viện spark-sklearn [Hunter & Bradley, 16], để thực hiện phân tán công việc cân chỉnh mô hình

dự báo của máy học véc-tơ hỗ trợ, rừng ngẫu nhiên trên nền Apache Spark đạt hiệu quả về thời gian. Kết quả của mô hình tối ưu tìm được sau khi cân chỉnh mô hình dự báo được so sánh với mô hình hồi quy tuyến tính, *k* láng giềng.

A. Chuẩn bị tập dữ liệu

Chúng tôi đã thu thập dữ liệu tại địa bàn Trung An, Quận Thốt Nốt, Thành Phố Cần Thơ. Tập dữ liệu thu được là kết quả điều tra tại 840 địa điểm (phần tử), với 24 thuộc tính khác nhau. Sau khi tiền xử lý, loại bỏ các thuộc tính không dùng trong dự báo như: số thứ tự, mã ruộng, các thuộc tính có dữ liệu nhiễu và số liệu điều tra sai lệch cũng được bỏ qua như: ngày điều tra, ngày sạ, tuổi lúa. Chúng tôi thu được 12 thuộc tính, trong đó có 11 thuộc tính dự báo dùng để xây dựng mô hình dự báo mật số rầy (thuộc tính phụ thuộc, có giá trị từ 0 đến 12900). Các thuộc tính dự báo bao gồm:

1. Kinh độ
2. Vĩ độ
3. Giống lúa
4. Mật độ sạ (kg/ha)
5. Nhiệt độ không khí (độ C)
6. Ẩm độ không khí (%)
7. Mực nước ruộng (cm)
8. So màu lá lúa (số màu: 1/2/3/4/5/6)
9. Mật số cỏ (cây/m^2)
10. Số chồi/m^2
11. Số lá/m^2

B. Xây dựng và cân chỉnh mô hình dự báo

Thí nghiệm thực hiện xây dựng các mô hình dự báo sử dụng tập dữ liệu có được để dự báo mật số rầy nâu từ 11 thuộc tính dự báo. Chúng tôi sử dụng nghi thức kiểm thử hold-out bằng cách lấy ngẫu nhiên 2/3 tập dữ liệu (560 dòng) làm tập huấn luyện các mô hình dự báo và 1/3 còn lại (280 dòng) làm tập kiểm tra kết quả dự báo. Kết quả dự báo được đánh giá trên tiêu chí trung bình lỗi tuyệt đối (Mean Absolute Error - MAE). Chúng tôi chỉ sử dụng tập huấn luyện để điều chỉnh các tham số của các mô hình. Các tham số này được lựa chọn sao cho đạt tiêu chí lỗi thấp nhất.

Xây dựng mô hình hồi quy tuyến tính (LM) không cần phải điều chỉnh bất kỳ tham số nào.

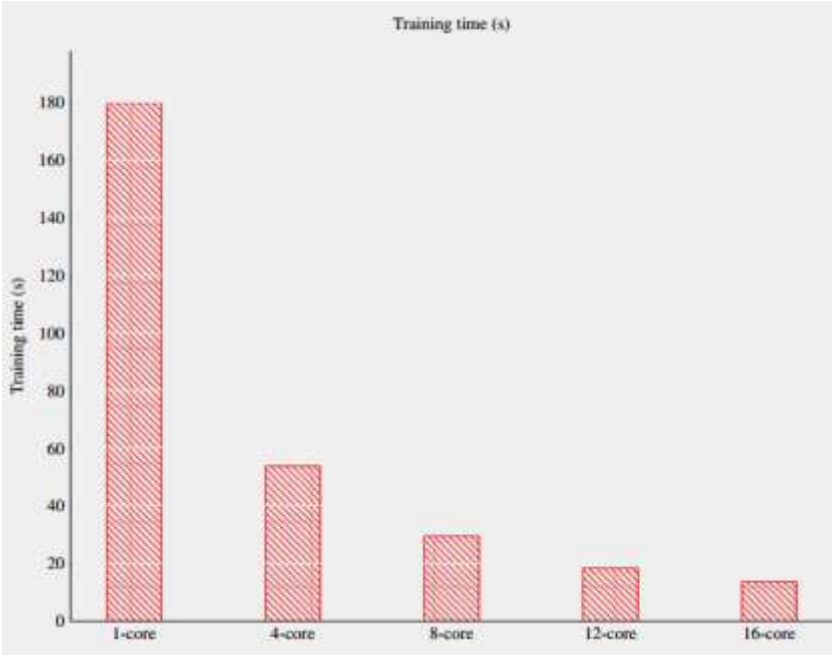
Mô hình *k* láng giềng sử dụng giá trị $k = 1, 2, ..., 10$. Kết quả dự báo chính xác nhất khi $k = 5$.

Mô hình máy học véc-tơ hỗ trợ cho hồi quy (SVR), chúng tôi đề xuất sử dụng hàm nhân phi tuyến RBF bởi vì tính tổng quát của hàm RBF so với các hàm nhân phi tuyến khác [Lin, 03]. Giải thuật máy học véc-tơ hỗ trợ xây dựng mô hình dự báo cần điều chỉnh 3 siêu tham số: tham số γ của hàm nhân RBF, tham số ε và hằng số c được sử dụng để chỉnh độ rộng lề và lỗi. Chúng tôi đề xuất tìm kiếm bộ 3 siêu tham số tối ưu trong lưới các giá trị như sau:

Bảng 1. Bảng giá trị của các siêu tham số cần điều chỉnh cho mô hình dự báo của máy học véc-tơ hỗ trợ

Tham số	Giá trị	Số giá trị
γ	0.0001, 0.00025, 0.0005, 0.00075, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1	17
ε	0.1, 0.15, 0.2, 0.25, 0.3	5
c	1, 10, 100, 1000, 10000, 100000	6
Tổng bộ 3 siêu tham số (γ, ε, c) khác nhau		510

Từ bảng 1, giải thuật máy học véc-tơ hỗ trợ cần thử tất cả 510 bộ 3 siêu tham số (γ, ε, c) khác nhau để chọn ra được mô hình tối ưu (cho lỗi thấp nhất). Chúng tôi sử dụng nền tảng Apache Spark, khảo sát thời gian huấn luyện 510 mô hình khác nhau theo sự thay đổi số lượng core được dùng trên hệ nhóm 4 máy tính như trình bày. Kết quả thu được trình bày trên hình 6.



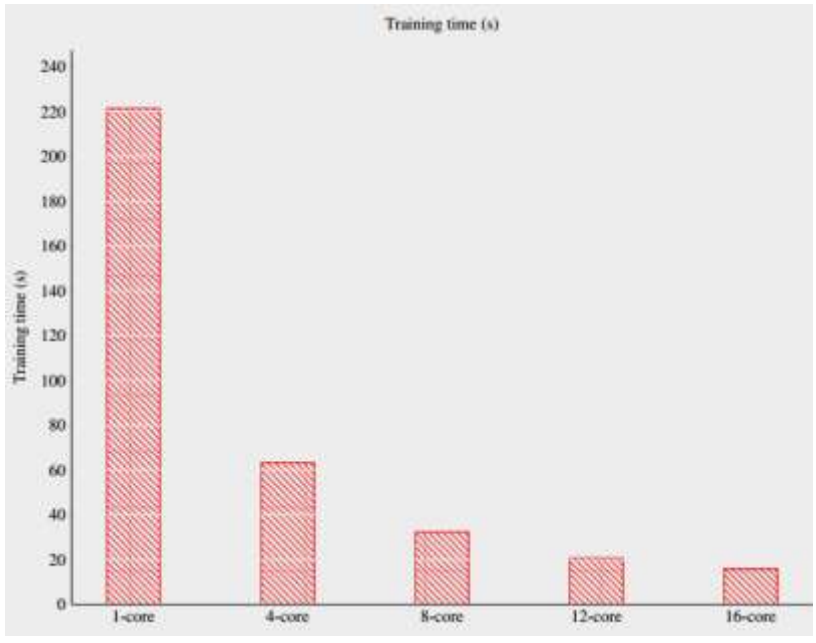
Hình 6. Thời gian huấn luyện 510 mô hình dự báo máy học véc-tơ hỗ trợ

Kết quả thu được bộ 3 siêu tham số tối ưu để huấn luyện mô hình dự báo máy học véc-tơ hỗ trợ là: $\gamma = 1$, $\varepsilon = 0.1$ và $c = 100000$.

Tương tự, giải thuật rừng ngẫu nhiên xây dựng mô hình dự báo cần điều chỉnh 4 tham số: tham số *n'* số thuộc tính ngẫu nhiên được sử dụng để tính phân hoạch tại nút trong của cây quyết định, *minobj* số phần tử ít nhất để thực hiện phân hoạch tại nút trong (điều kiện dừng sớm quá trình phân hoạch của cây quyết định), *maxdepth* độ sâu tối đa của cây quyết định, *T* tổng số cây cần xây dựng trong rừng. Chúng tôi đề xuất tìm kiếm bộ 4 tham số tối ưu trong lưới các giá trị như sau:

Bảng 2. Bảng giá trị của các tham số cần điều chỉnh cho mô hình dự báo của rừng ngẫu nhiên

Tham số	Giá trị	Số giá trị
n'	1, 2, 3, 4, 5, 6	6
minobj	1, 3, 5, 7, 9, 11	6
maxdepth	5, 10, không giới hạn	3
T	50, 100, 200	3
Tổng bộ 4 tham số (*n', minobj, maxdepth, T*) khác nhau		324



Hình 7. Thời gian huấn luyện 324 mô hình dự báo rừng ngẫu nhiên

Từ bảng 2, giải thuật rừng ngẫu nhiên cần thử tất cả 324 bộ 4 tham số (*n'*, *minobj*, *maxdepth*, *T*) khác nhau để chọn ra được mô hình dự báo chính xác nhất. Hình 7 trình bày thời gian huấn luyện 324 mô hình khác nhau theo sự thay đổi số lượng core được dùng trên hệ nhóm 4 máy tính. Kết quả thu được bộ 4 tham số tối ưu để huấn luyện mô hình dự báo rừng ngẫu nhiên là: *n' = 3*, *minobj=5*, *maxdepth=10* và *T=100*.

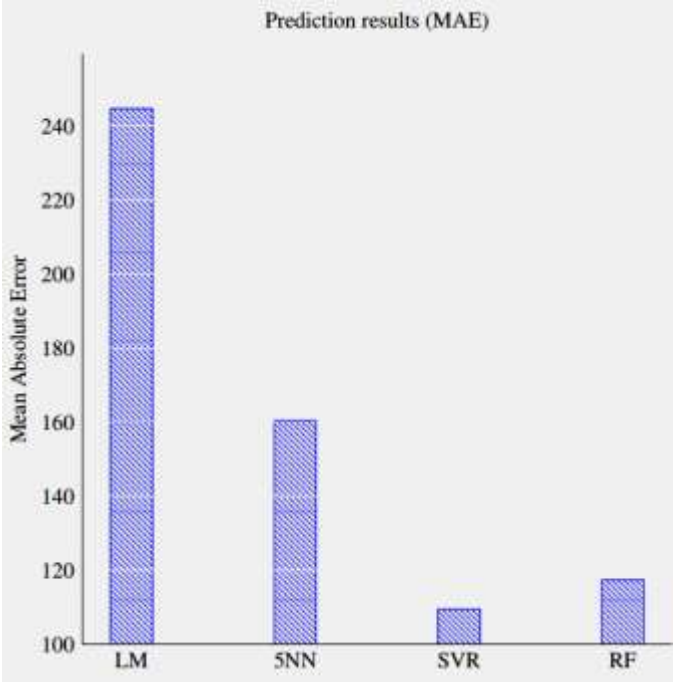
Từ kết quả cần chỉnh mô hình dự báo của máy học véc-tơ hỗ trợ và rừng ngẫu nhiên trên nền Apache Spark, chúng ta có thể thấy rằng thời gian cân chỉnh để tìm bộ tham số cho các mô hình dự báo giảm gần như tuyến tính khi số lượng core được sử dụng tăng lên trong hệ nhóm máy tính. Điều này cho thấy được hiệu quả về mặt thời gian khi phân tán công việc cân chỉnh mô hình dự báo của máy học véc-tơ hỗ trợ, rừng ngẫu nhiên trên nền Apache Spark.

C. Kết quả dự báo

Từ khi tìm được bộ siêu tham số tối ưu sau bước cân chỉnh các mô hình, chúng tôi sử dụng các mô hình tối ưu này để dự báo mật số rầy nâu. Bảng 3, hình 8 trình bày kết quả dự báo thu được từ các mô hình được đánh giá theo tiêu chí trung bình lỗi tuyệt đối. Kết quả dự báo trong bảng 3 với lỗi thấp nhất được in đậm, lỗi thấp thứ hai được gạch dưới. So sánh các kết quả thu được từ các mô hình dự báo, có thể thấy rằng mô hình hồi quy tuyến tính (LM) cho lỗi dự báo cao nhất. Trong khi các mô hình dự báo phi tuyến chứng tỏ nhiều ưu thế hơn. Mô hình *5* láng giềng (*5*NN) cho kết quả dự báo chính xác hơn mô hình hồi quy tuyến tính. Đặc biệt là mô hình hồi quy máy học véc tơ hỗ trợ (SVR) và rừng ngẫu nhiên (RF) dự báo mật số rầy chính xác hơn rất nhiều so với mô hình hồi quy tuyến tính và *5* láng giềng. Mô hình SVR và RF giảm tỷ lệ lỗi hơn 2 lần khi so với LM, khoảng 1.4 lần so với *5*NN.

Bảng 3. Kết quả dự báo mật số rầy nâu của các mô hình

Mô hình dự báo	**Tỷ lệ lỗi (MAE)**
Hồi quy tuyến tính (LM)	244.6689
k láng giềng (*k*NN, *k=5*)	160.4421
Máy học véc-tơ hỗ trợ (SVR) $\gamma = 1$, $\varepsilon=0.1$ và *c=100000*	**109.508**
Rừng ngẫu nhiên (RF) *n' = 3, minobj=5, maxdepth=10* và *T=100*	<u>117.396</u>



Hình 8. So sánh kết quả dự báo mật số rầy nâu của các mô hình

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa trình bày trong bài viết về tiếp cận xây dựng và cân chỉnh nhanh mô hình máy học véc-tơ hỗ trợ và rừng ngẫu nhiên trên nền tảng Apache Spark, để dự báo hiệu quả mật số rầy nâu. Quá trình xây dựng mô hình máy học véc-tơ hỗ trợ và rừng ngẫu nhiên mất nhiều thời gian điều chỉnh các bộ tham số, để tạo ra mô hình dự báo tối ưu. Chúng tôi đề xuất phân tán các tác vụ cân chỉnh hai mô hình dự báo này trên nền tảng tính toán nhóm trên bộ nhớ trong, Apache Spark. Thực nghiệm cho thấy rằng phân tán công việc cân chỉnh mô hình dự báo của máy học véc-tơ hỗ trợ, rừng ngẫu nhiên trên nền Apache Spark đạt hiệu quả về thời gian khi tăng số lượng nút sử dụng trong hệ nhóm

máy tính. Kết quả của mô hình tối ưu tìm được sau khi cân chỉnh mô hình dự báo chính xác mật số rầy nâu khi so sánh với các mô hình hồi quy tuyến tính, *k* láng giềng.

Tiếp cận dự báo được đề xuất trong bài là tổng quát cho các vấn đề về dự báo dịch hại. Trong tương lai, chúng tôi sẽ nghiên cứu áp dụng cho các vấn đề dự báo tương tự như dự báo bệnh đạo ôn, sâu cuốn lá và các vấn đề tương tự.

TÀI LIỆU THAM KHẢO

[1] Apache Software Foundation, "Apache Spark™ Lightning-fast cluster computing", Retrieved 4 March 2014, http://spark.apache.org.

[2] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.: "*Classification and Regression Trees*", Wadsworth International, 1984.

[3] Breiman, L.: "Bagging predictors", *Machine Learning* vol.24(2):123–140, 1996.

[4] Breiman, L.: "Arcing classifiers", *The Annals of Statistics* vol.26(3):801-849, 1998.

[5] Breiman, L.: "Random forests", *Machine Learning* vol.45(1):5-32, 2001.

[6] Chang, C. C., Lin, C. J., "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp.1-27, 2011 http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] Cristianini, N., Shawe-Taylor, J., "*An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*", Cambridge University Press, New York, NY, USA, 2000.

[8] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: "From Data Mining to Knowledge Discovery in Databases", in *AI Magazine* vol.17(3):37-54, 1996.

[9] Freund, Y., Schapire, R., "A short introduction to boosting", *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp.771-780, 1999.

[10] E. Fix, J. Hodges. Discriminatoiry Analysis: Small Sample Performance. *Technical Report 21-49-004, USAF School of Aviation Medicine, Randolph Field, USA*, (1952).

[11] Guyon, I., Web page on svm applications, 1999, http://www.clopinet.com/isabelle/Projects/SVM/applist.html.

[12] Hastie, T., Friedman, J-H., Tibshirani, R.: "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*", Springer, 2001.

[13] Hunter, T. and Bradley, J.: "Auto-scaling scikit-learn with Spark", databricks, 2016.

[14] Ihaka, R., Gentleman, R.: "R: A language for data analysis and graphics", *Journal of Computational and Graphical Statistics*, vol.5(3):299-314, 1996.

[15] Lin, C., "A practical guide to support vector classification", 2003.

[16] Nguyễn, NG-V., "Hệ thống đa tác tử và mô hình hóa khả năng ra quyết định dựa vào nhiều tiêu chí trong đánh giá rủi ro côn trùng hại lúa", *CNTT trong hỗ trợ ra quyết định về giáo dục, nông nghiệp, thủy sản và môi trường vùng ĐBSCL* 2016, NXB ĐHCT, Trang: 70-85.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research* vol.12:2825-2830, 2011.

[18] Quinlan, J.R.: "*C4.5: Programs for Machine Learning*", Morgan Kaufmann, 1993.

[19] Trương, C-Q., Võ, Q-M., Trần, T-T., Trần, T-Đ., "Ứng dụng GIS dự báo trung hạn khả năng nghiễm rầy nâu trên lúa – Trường hợp nghiên cứu ở Đồng Tháp", *Tạp chí Khoa học Trường Đại học Cần Thơ*, Số 17a: 103-109, 2011.

[20] Vapnik, V.: "*The Nature of Statistical Learning Theory*", Springer-Verlag, 1995.

[21] Võ, Q-M., Trần, T-H., "Cảnh báo dịch hại lúa ở Đồng bằng sông Cửu long trên cơ sở sử dụng ảnh viễn thám MODIS", *Số tạp chí Chuyên Đề Nông Nghiệp* (2014), Trang: 124-132.

[22] Võ, Q-M., Huỳnh, TT-H., Trần, T-H., "Ứng dụng ảnh viễn thám xác định hiện trạng sinh trưởng cây lúa cảnh báo dịch hại tỉnh An Giang", *Số Chuyên Đề CNTT2015*, Trang: 203-211.

[23] Vũ, D-L., Huỳnh, X-H., "Dự báo mức độ nhiễm, cháy và lan truyền rầy theo thời gian", *CNTT trong hỗ trợ ra quyết định về giáo dục, nông nghiệp, thủy sản và môi trường vùng ĐBSCL* 2016, NXB ĐHCT, Trang: 102-118.

[24] Wu X. and Kumar V.: "*Top 10 Algorithms in Data Mining*", Chapman & Hall/CRC, 2009.

[25] Zaharia, M., Chowdhury, M., Franklin, M., Shenker, S., Stoica, I.: "Spark: Cluster Computing with Working Sets", USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), 2010.

BUILDING PREDICTION MODELS OF BROWN PLANTHOPPER USING APACHE SPARK

Do Thanh Nghi, Tran Nguyen Minh Thu, Bui Vo Quoc Bao, Pham Nguyen Khang

ABSTRACT— *In this paper, we present the approach for building the prediction models of brown planthopper. The support vector machines and random forests are the most powerful prediction models. Hyperparameter optimization is the problem of choosing a set of hyperparameters for a learning algorithm, usually spending very long time to get the highest accuracy. Our proposal is to distributes the most repetitive tasks of model tuning on a PC cluster using Apache Spark. This aim is to reduce runtime of hyperparameter optimization for building the prediction models of brown planthopper. The numerical test results show that hyperparameter tuning of support vector machines and random forests in parallel way improve runtime scaling to the number of nodes used in Spark cluster. The support vector machines model and random forests model using optimized hyperparamters are the most accurate brown planthopper prediction, compared to the linear regression model and k nearest neighbors.*