# DỰ BÁO SUY THẬN MÃN TÍNH VỚI MÁY HỌC TRÊN NỀN TẢNG ĐIỆN TOÁN ĐÁM MÂY

## PREDICTION OF CHRONIC KIDNEY DISEASE WITH CLOUD-BASED MACHINE LEARNING

PHẠM VĂN ÂU[1a], HUỲNH CHÍ HIẾU[1]
*[1]Trường Trung Cấp Kỹ Thuật-Nghiệp Vụ Cái Bè,*
*[a] Tác giả liên hệ: auphamphi@gmail.com*

***Nhận bài (Received): 23/3/2023; Phản biện (Reviewed): 28/3/2023; Chấp nhận đăng (Accepted): 05/5/2023***

**TÓM TẮT**

*Có rất nhiều bệnh nguy hiểm và nguy cơ gây tử vong trong đó có bệnh suy thận mãn tính. Nếu người bệnh sớm phát hiện dấu hiệu, chẩn đoán chuẩn xác và chữa trị kịp thời sẽ hạn chế đến mức thấp khả năng mắc bệnh và ngăn chặn bệnh diễn tiến nhiều hơn và giảm được khả năng tử vong do bệnh gây ra. Từ trước cho đến nay có rất nhiều mô hình để dự đoán các loại bệnh và dự đoán suy thận mãn tính bằng phương pháp xây dựng từng mô hình khác nhau[1][2][3]. Trong nghiên cứu này chúng tôi tập trung vào việc xây dựng các mô hình là, Cây quyết định, Bagging, Boosting. Kết quả thực nghiệm trên bộ dữ liệu Chronic KIdney Disease L.Jerlin Rubini cho thấy Boosting có hiệu suất dự đoán cao hơn so với các mô hình dự đoán mà chúng tôi sử dụng.*

***Từ khóa:*** *Cây quyết định, Bagging, Boosting, máy học, bệnh thận mãn tính.*

**ABSTRACT**

*There are many dangerous and fatal diseases, including chronic kidney disease. If the disease is early detected, accurately diagnosed and timely treated, this will will limit the likelihood of disease and prevent the disease from developing and reduce the possibility of death caused by the disease. So far, there have been many models to predict diseases and predict chronic kidney failure by building different models. In this study, we focus on building models, namely, Decision Tree, Bagging, and Boosting. Results obtained from experiments conducted on the Chronic KIdney Disease L.Jerlin Rubini dataset show that Boosting has a higher predictive performance than the prediction models we are using.*

***Keywords:*** *Decision tree, Bagging, Boosting, machine learning, chronic kidney disease.*

## 1. GIỚI THIỆU

Bệnh thận mãn đặc biệt nguy hiểm đối với nhóm người bệnh đái tháo đường, huyết áp cao và bệnh lý cầu thận. Trên thế giới hiện có khoảng trên 850 triệu người mắc bệnh thận mãn và khoảng trên 3 triệu người bị mắc bệnh đang được điều trị thay thế thận. Xây dựng ứng dụng giúp người bệnh có thể phát hiện sớm, tiên lượng với độ chính xác cao, hạn chế tỷ lệ tử vong do bệnh thận mãn tính. Ngoài ra, các ứng dụng được phát triển trên nền điện toán đám mây

giúp giảm chi phí về hạ tầng cơ sở vật chất, thiết bị lưu trữ, xử lý dữ liệu, đồng thời giúp người dùng có thể truy cập mọi lúc, mọi nơi trên các ứng dụng.

Để đưa ra dự báo với độ chính xác (có thể) cao nhất và để tìm ra mô hình dự báo khả thi là điều cần thiết. Tuy nhiên, các mô hình dự đoán hiện nay chưa thể hiện tính tổng quát. Vì vậy, xây dựng một mô hình dự báo khả thi, tiết kiệm, có tính khái quát cao là mong muốn không chỉ của các nhà nghiên cứu mà còn của tất cả mọi người trong cuộc sống. Phương pháp kết hợp nhiều mô hình dự đoán riêng biệt để cho độ chính xác cao hơn gọi là phương pháp Ensemble.

Trong nghiên cứu này, chúng tôi sử dụng ba mô hình Cây quyết định, Bagging, Boosting. Trong đó, chúng tôi sẽ triển khai các kỹ thuật của ba mô hình với ngôn ngữ R với bộ dữ liệu bệnh thận mãn tính trên dịch vụ đám mây Amazon. Mô hình dự báo theo phương pháp Ensemble, cụ thể là Boosting có độ chính xác cao hơn so với các mô hình riêng lẻ trước đây.

Cấu trúc của bài viết được chia thành 3 Phần. Phần 2 là nội dung (gồm Cây quyết định, Bagging, Boosting, Dịch vụ đám mây Amazon và thực nghiệm với bộ dữ liệu Chronic KIdney Disease L.Jerlin Rubini). Phần 3 là Bàn luận và kết luận: trình bày một bản tóm tắt các kết quả đạt được.

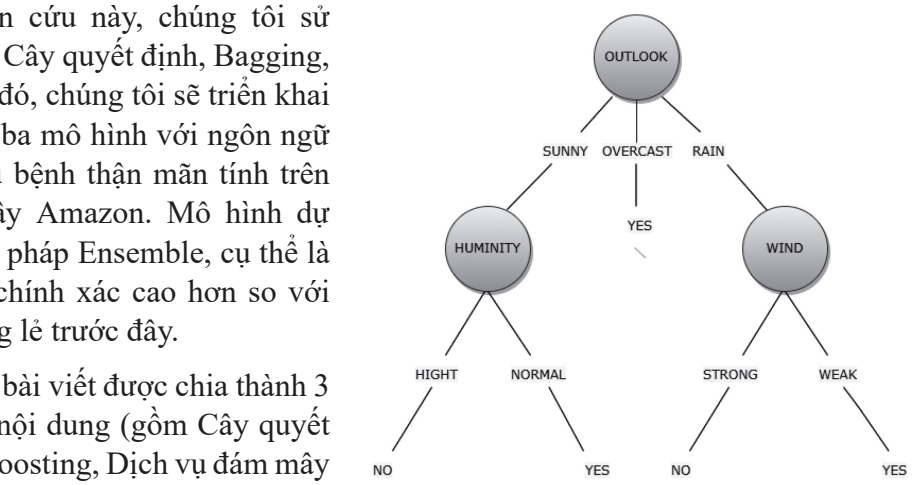
## 2. NỘI DUNG

### 2.1. Cây quyết định

Cây quyết định [12] là một thuật toán học có giám sát được sử dụng tương đối phổ biến và hiệu quả bao gồm cả hai thuật toán phân lớp phân loại và dự báo hồi quy. Cây quyết định mô phỏng tương đối chính xác việc suy nghĩ của con người và để xử lý các vấn đề. Không giống như những thuật toán khác trong học có giám sát, mô hình cây quyết định không tồn tại phương trình dự đoán.

Mục tiêu của chúng ta trong việc sử dụng cây quyết định là tạo ra mô hình đào tạo bằng việc học các quy tắc đơn giản từ dữ liệu đào tạo hay dữ liệu trước đó rồi suy ra kết quả của mô hình. Cho nên việc chúng ta cần làm đó là tìm ra một cây quyết định dự báo thật tốt trên tập dữ liệu huấn luyện và sử dụng cây quyết định này dự báo trên tập dữ liệu kiểm tra.

Sau đây là ví dụ về cây quyết định chơi môn Bóng Đá Bãi Biển với dữ liệu ban đầu là các thuộc tính sau: Thời Tiết, Nhiệt Độ, Độ Ẩm và Gió.




***Hình 1. Cây quyết định***

Dựa vào mô hình trên ta có kết quả: Nếu trời nắng, độ ẩm bình thường thì khả năng chơi. Còn nếu trời nắng, độ ẩm cao thì khả năng không đi chơi. Nếu trời âm u sẽ chơi. Nếu trời mưa gió nhẹ sẽ chơi, còn trời mưa gió mạnh sẽ không chơi.

Có rất nhiều thuật toán để xây dựng cây quyết định như ID3, C4.5, CART, CHAID, MARS…Trong đó thuật toán ID3 là thuật toán phổ biến và được sử dụng khá nhiều.

Iterative Dichotomiser 3(ID3) là một thuật toán tham lam được đề xuất bởi Ross Quilan [18] vào 1986.

Tại mỗi nút N chọn một thuộc tính A (thuộc tính A này có thể giúp chúng ta phân loại dữ liệu N tốt nhất).

Tạo các nhành con cho A và sau đó phân phối dữ liệu vào các nhánh con tương ứng cho phù hợp.

Tương tự như thế phát triển cây cho đến khi nào phân loại hoàn toàn chính xác tất cả dữ liệu huấn luyện của chúng ta hoặc tất cả thuộc tính được sử dụng hết.

Lưu ý là mỗi thuộc tính chỉ sử dụng một lần trên suốt đường đi của cây từ nút gốc đến nút lá. 2 yếu tố để chọn nút gốc cho cây quyết định là Entropy và Gain

**Information gain:**

Entropy: là đặc trưng cho độ tinh khiết của một tập hợp.

Cho một tập S với c lớp có thể định nghĩa như sau

$$Entropy(S) = \sum_{i=1} -p_i \log_2 p_i \quad (1)$$

Information Gain giúp chúng ta đo đạc mức độ giảm entropy nếu chúng ta chia tập S thành các tập con theo thuộc tính, được tính theo công thức sau:

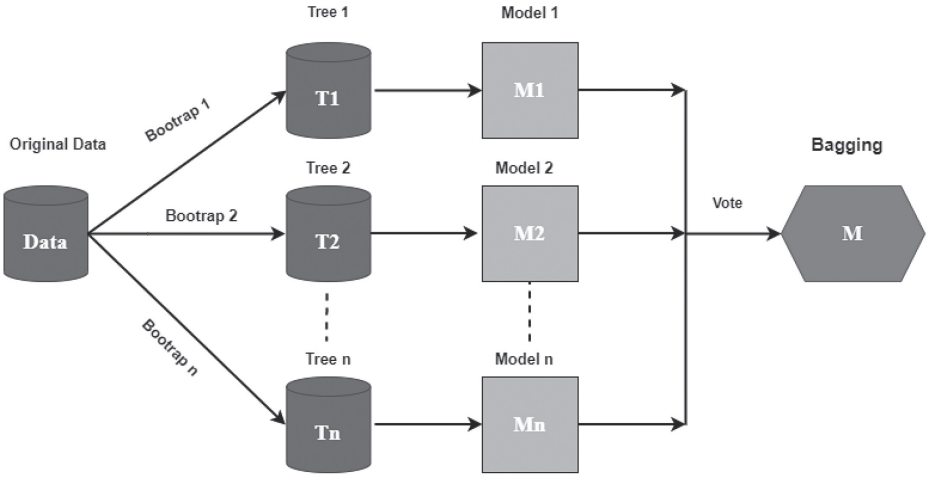
$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

### 2.2. Bagging

Bagging được đề xuất bởi Breiman [17] nhằm mục đích cải thiện hiệu quả với việc mất cân bằng dữ liệu của thuật toán đơn lẻ như là cây quyết định. Từ tập dữ liệu ban đầu chúng ta sử dụng phương pháp lấy mẫu boostrap để chia thành nhiều tập dữ liệu con huấn luyện song song nhau cùng một thuật toán. Kết quả của mô hình thu được là giá trị trung bình của các mô hình hoặc chọn theo kiểu đa số phiếu bầu.

**Random Forest**

Random Forest [9] tập hợp rất nhiều cây thì chúng ta gọi là rừng cho nên rừng quyết định là tập hợp nhiều cây quyết định theo phương pháp bagging nhằm xây dựng một bộ sưu tập lớn các cây không tương quan để cải thiện hơn nữa hiệu suất dự đoán.



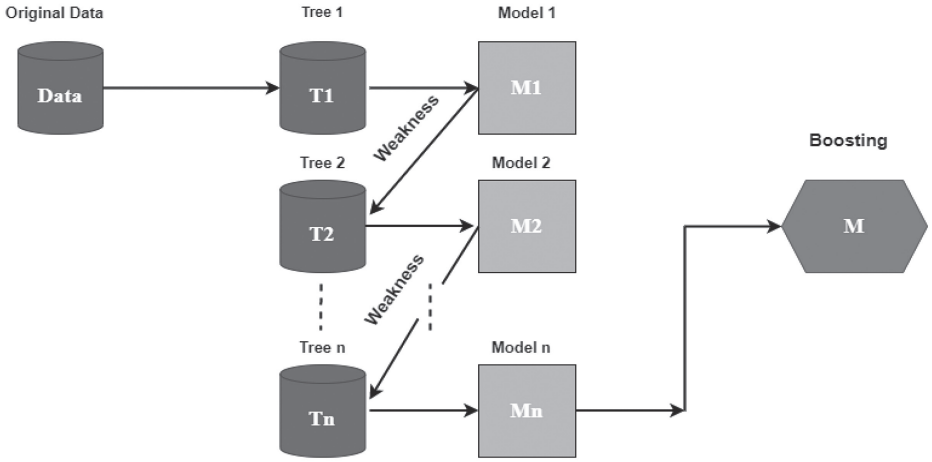

*Hình 2. Bagging(Random Forest)*

### 2.3. Boosting

Boosting [6] xây dựng với mong muốn cải thiện hạn chế của Bagging( do các mô hình trong Bagging học riêng lẻ nhau không ảnh hưởng và liên quan gì nhau dẫn đến kết quả không tốt khi các mô hình trùng kết quả với nhau). Vì vậy trong boosting các model yếu sẽ học bổ sung lẫn nhau để hạn chế lỗi của model trước đó, quá trình huấn luyện trong phương pháp này diễn ra tuần tự theo chuỗi.

**Gradient Boosting**

Gradient Boosting [14] tạo ra người học bằng cách sử dụng cùng một quy trình học tập tăng cường chung. Trước tiên, mô hình xây dựng cho người học để dự đoán các giá trị, nhãn của các mẫu và tính toán sự mất mát (sự khác biệt giữa kết quả của người học đầu tiên và giá trị thực). Mô hình sẽ xây dựng một người học thứ hai để dự đoán sự mất mát sau bước đầu tiên. Bước tiếp tục tìm hiểu bước thứ ba, thứ tư… cho đến ngưỡng nhất định được tính trong các lần lặp trước.




***Hình 3. Boosting***

Như hình trên boosting gồm các bước cơ bản bước 1 tạo nhiều tập dữ liệu thông qua lấy mẫu ngẫu nhiên bằng cách thay thế dữ liệu có trọng số, bước 2 xây dựng học viên theo trình tự, bước 3 kết hợp tất cả học viên bằng cách sử dụng chiến lược lấy trung bình có trọng số.

**2.4. Dịch vụ đám mây Amazon**

Ngày nay các dịch vụ điện toán đám mây có thể giúp chúng ta giảm chi phí về cơ sở vật chất hạ tầng, thiết bị lưu trữ cũng như xử lý dữ liệu, đồng thời giúp cho người sử dụng có thể truy cập mọi nơi.…

Có rất nhiều cách để sử dụng công nghệ đám mây để chạy R như IBM, Microsoft, Amazon, Google, Vmware…Ở đây chúng tôi sử dụng đám mây Amazon (AWS).

Bước 1: Vào https://aws.amazon.com/ để tạo tài khoản

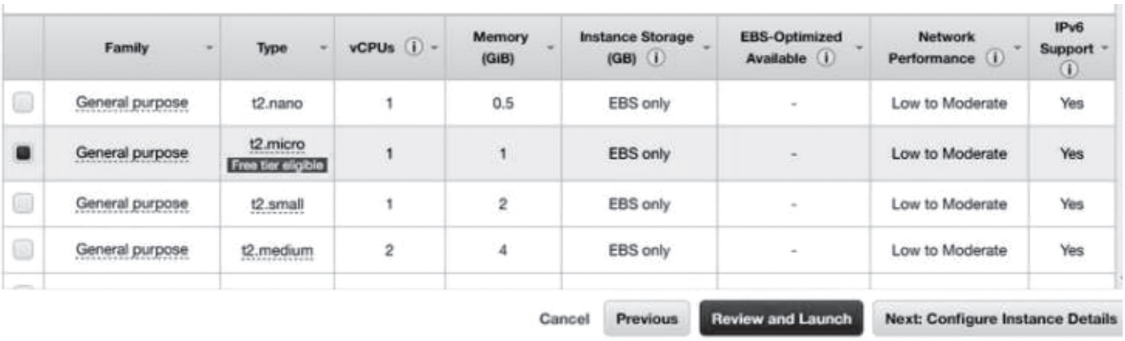
Bước 2: Sử dụng Amazon Machine Image(AMI)

| Release | | EU West Ireland | EU West London | EU West Paris | EU Central Frankfurt | EU North Stockholm | Canada Central |
|---|---|---|---|---|---|---|---|
| RStudio 1.3.1073<br>R 4.0.2<br>CUDA 10.1/cuDNN 7.6.5<br>**Available** | 64-bit HVM | ami-05bf201d51b1db642 | ami-0b4be5cd9e848fabb | ami-005af3b164a016fac | ami-076abd591c4335092 | ami-0fa80e7cbbc94e3b7 | ami-0bdd24fd36f07b638 |
| RStudio 1.2.1335<br>R 3.6.0<br>CUDA 10.0/cuDNN 7.5.1<br>**Available** | 64-bit HVM | ami-0754449f54adcc62d | ami-0e9e5245fffe34a3e | ami-08cd0f9ecf5f0a4c2 | ami-059a2456bd2027e31 | ami-e2a32b9c | ami-09b8f2f441fc21b6f |
| RStudio 1.1.456<br>R 3.5.1<br>CUDA 9/cuDNN 7.2.1<br>**Available** | 64-bit HVM | ami-003a0987ccad642ec | ami-0f530f457cea9b2ce | ami-0a4e5ca7ecce53941 | ami-0cccb5af1db2d3ec6 | N/A* | ami-0ccb91fcd3d2460f3 |
| RStudio 1.1.383<br>R 3.4.2<br>Julia 0.6.0<br>CUDA 8/cuDNN 6<br>**Available** | 64-bit HVM | ami-93805fea | ami-bf6b76db | N/A* | ami-a80db3c7 | N/A* | ami-75c17911 |
| RStudio 1.0.153<br>R 3.4.1<br>Julia 0.6.0<br>**Retired** | 64-bit HVM | ami-fodf2489 | ami-dfdbcbbb | N/A* | ami-0e5cf661 | N/A* | ami-7245fb16 |

***Hình 4. AMI***

Bước 3: Chọn server phù hợp

Bước 4: Chọn cấu hình máy mong muốn sử dụng



| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|---|---|---|---|---|---|---|---|---|
| ☐ | General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| ■ | General purpose | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| ☐ | General purpose | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| ☐ | General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |



***Hình 5. Chọn một phiên bản***

Bước 5: Chọn Configure Sercurity Group

Bước 6: Tìm địa chỉ IP Public của máy tính vừa tạo và truy cập

Bước 7: Truy cập và sử dụng Rstudio trên AWS.

**2.5. Thực nghiệm**

***2.5.1. Dữ liệu***

Trong nghiên cứu này, chúng tôi đã sử dụng bộ dữ liệu Chronic KIdney Disease L.Jerlin Rubini [19] với các thông tin: Tuổi, Huyết áp, Trọng lượng riêng, Albumin, Đường, Tế bào hồng cầu, Tế bào mủ, Khối tế bào mủ, Vi khuẩn, Glucose máu ngẫu nhiên, Huyết thanh Urê, Creatinine, Natri, Kali, Hemoglobin, Thể tích tế bào, Số lượng tế bào bạch cầu, Số lượng hồng cầu, Tăng huyết áp, Bệnh tiểu đường, Mellitus, Bệnh động mạch vành, Thèm ăn, Bàn đạp, Phù Thiếu máu, Phân loại.

***2.5.2. Công cụ***

Chúng tôi đã lập trình bằng ngôn ngữ R cho mô hình tiên đoán, các mô hình được sử dụng là: Mô hình Cây quyết định, mô hình Bagging và mô hình Boosting. Ngoài ra, chúng tôi cũng sử dụng các thư viện: thư viện (dplyr), thư viện (ggplot2), thư viện (e1071), thư viện (caret), thư viện (rpart), thư viện (ipred), thư viện (tidyverse) và gói h2o, thư viện (h2o) để tích hợp các mô hình.

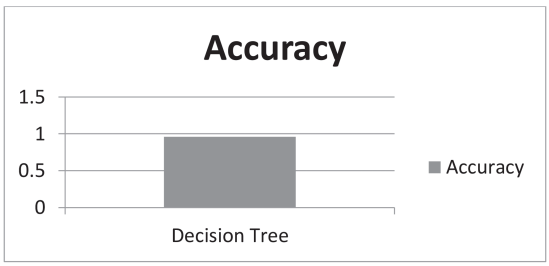
***2.5.3. Kịch bản 1:*** Cây quyết định

Dựa vào biến "Appetite"(Thèm ăn): good (thèm ăn nhiều) và poor (ít thèm ăn) để đưa ra dự đoán về hiệu suất good, poor chúng tôi đã chọn thuật toán Cây quyết định.

*Bảng 1. Độ chính xác của mô hình Cây quyết định*

| Model | Decision Tree |
|---|---|
| Accuracy | 0.9592 |

Kết quả độ chính xác trung bình của mô hình Cây quyết định là 0.9592





***Hình 6. Biểu đồ độ chính xác mô hình Cây quyết định***

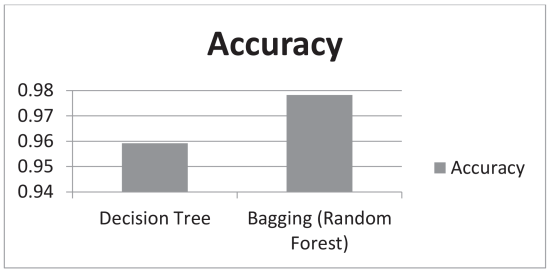
***2.5.4. Kịch bản 2: Baging (Random Forest)***

Chúng tôi sử dụng phương pháp Bagging (Random Forest) với library (tidyverse) library(h2o). Chia tập dữ liệu thành tập dữ liệu Train (80%) và Test (20%).

*Bảng 2. Độ chính xác của mô hình Bagging*

| Model | Decision Tree | Bagging |
|---|---|---|
| Accuracy | 0.9592 | 0.9782 |

Kết quả độ chính xác trung bình của mô hình Bagging(Random Forest) là 0.9782 so với Cây quyết định là 0.9592




***Hình 7. Biểu đồ độ chính xác mô hình Bagging so với Cây quyết định***

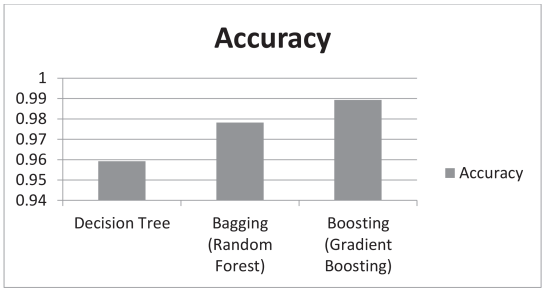
### *2.5.5. Kịch bản 3: Boosting (Gradient Boosting)*

Chúng tôi sử dụng phương pháp Gradient boosting với library (tidyverse) library (h2o). Chia tập dữ liệu thành tập dữ liệu Train (80%) và Test (20%).

*Bảng 3. Độ chính xác của mô hình Bossting*

| Model | Decision Tree | Bagging (Random Forest) | Boosting (Gradient Boosting) |
|---|---|---|---|
| Accuracy | 0.9592 | 0.9782 | 0.9893 |

Kết quả độ chính xác trung bình của mô hình Boosting( Gradient Boosting) là Bagging(Random Forest) là 0.9782 so với Cây quyết định là 0.9592




***Hình 8. Biểu đồ độ chính xác mô hình Boosting so với Bagging và Cây quyết định***

## 3. THẢO LUẬN VÀ KẾT LUẬN

Ngày nay các dịch vụ điện toán đám mây có thể giúp chúng ta giảm chi phí về cơ sở vật chất hạ tầng, thiết bị lưu trữ cũng như xử lý dữ liệu, đồng thời giúp cho người sử dụng có thể truy cập mọi nơi.… với thí nghiệm và phân tích này chúng tôi chạy thực nghiệm trên nền tảng dịch vụ đám mây Amazon và có thể đưa ra một dự đoán về mức độ khả năng gây bệnh thận mãn tính nhằm mục đích giảm thiểu thấp nhất khả năng bệnh cho nhóm người có nguy cơ cao, để những người này có biện pháp an toàn cho sức khỏe của họ và có thể tránh được bệnh thận mãn tính nói riêng và có thể áp dụng cho các bệnh khác nói chung.Việc kết hợp chạy phương pháp Ensemble trên nền tảng đám mây sẽ giảm được chi phí về cơ sở vật chất, thiết bị lưu trữ … và phương pháp Ensemble Boosting so với Cây quyết định và Bagging sẽ cho kết quả có hiệu suất cao hơn các phương pháp khác. Trong nghiên cứu này, chúng tôi chỉ sử dụng một số mô hình dự đoán trên ngôn ngữ R, một số phương pháp và trên bộ dữ liệu thực nghiệm. Nhưng trên thực tế, có rất nhiều loại mô hình dự đoán bằng các ngôn ngữ khác và thực nghiệm trên các bộ dữ liệu khác nhau, vì vậy để tìm ra phương pháp dự đoán tốt hơn, chúng ta có thể áp dụng cách tiếp cận này cho các loại mô hình ngôn ngữ và bộ dữ liệu khác trong tương lai.

Trong bài viết này, chúng tôi tìm hiểu 3 phương pháp học tập thể tiêu chuẩn cho máy học và áp dụng chúng vào bài toán dự đoán về khả năng gây bệnh thận mãn tính từ biến "Appetite"(Thèm ăn). Kết quả thực nghiệm cho thấy các phương pháp Ensemble tốt hơn các phương pháp khác với Boosting cho kết quả tốt nhất trong thử nghiệm của chúng tôi. Những bộ phân loại tổng hợp này thực sự phù hợp để phân loại tập dữ liệu nhưng cần có một cách tiếp cận khác để tìm ra bộ phân loại chính xác nhất nhằm cải thiện độ chính xác của tập dữ liệu.

## TÀI LIỆU THAM KHẢO

[1] Dibaba Adeba Debal, Tilahun Melak Sitote, 2022; "Chronic kidney disease prediction using machine learning techniques", Journal of Big Data.

[2] Elias Dritsas, Maria Trigka, 2022; "Machine Learning Techniques for Chronic Kidney Disease Risk Prediction", MDBI.

[3] Imesh Udara Ekanayake, Damayanthi Herath, 2020; "Chronic Kidney Disease Prediction Using Machine Learning Methods", IEEE

[4] Cha Zang, Yunqian Ma, 2012; "Ensemble Machine Learning Methods and Applications", Springer Science+Business Media. LLC

[5] Rosaida Rosly, Mokhairi Makhtar, Mohd Khalid Awang, Nordin Abdul Rahman, Mustafa Mat Deris, 2015; "Comparison of Ensemble Classifiersfor Water Quality Dataset". Proceedings of the UniSZA Research Conference .

[6] Harris Drucker, Corinna Cortes, Larry Jackel, Yann LeCun, 1994; "Boosting and Other Ensemble Methods" Neural Computation 6.

[7] Ljupco Todorovski, Saso Dzeroski, 2003; "Combining classifiers with meta decision trees", Machine learning, 50(3).

[8] David H.Wolpert, 1992; "Stacked generalization", Neural networks

[9] Adele, C., David, R. C., John, R. S, 2011; "Random Forests", Machine Learning.

[10] Panagiotis Pintelas, Ioannis E. Livieris, 2020; "Ensemble Algorithms and Their Applications", Mdpi AG.

[11] Theyazn H. H. Aldhyani, Mohammed AI-Yaari, Hasan Alkahtanni, Mashael Maashi. 2020; "Water Quality Prediction Using Artificial Intelligence Algorithms", Applied Bionics and Biomechanics.

[12] Lior Rokach, Oded Maimon, 2005; "Decision Tree", researchGate.

[13] SOCIAL-SCIENCES https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/bootstrap-method, (2022).

[14] https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost, (2022).

[15] https://www.geeksforgeeks.org/ml-xgboost-extreme-gradient-boosting/?ref=lbp, (2022).

[16] Felipe Kenji Nakano, Saulo M. Mastelini, Sylvio Barbon, Ricardo Cerri, (2017); "Stacking Methods for Hierarchical Classification", IEEE International Conference on Machine Learning and Applications

[17] Leo Breiman,1996; "Bagging predictors", Machine learning.

[18] J.R.Quinlan, 1986; "Induction of Decision Trees", Mach. Learn.

[19] https://matthew-brett.github.io/cfd2019/data/chronic_kidney_disease