Xây dựng hệ thống dự đoán mức tiêu thụ năng lượng Real-time cho thành phố

1st Nguyễn Ngọc Lương
UIT 21522311
KHDL2021
21522311@gm.uit.edu.vn

2nd Nguyễn Phú An
UIT 21521807
KHDL2021
21521807@gm.uit.edu.vn

***Tóm tắt nội dung*—Ngày nay, việc sử dụng năng lượng là một vấn đề quan trọng đối với kinh tế và môi trường. Việc sử dụng năng lượng không hiệu quả có thể dẫn đến tăng giá năng lượng và ô nhiễm môi trường. Do đó, cần phải sử dụng năng lượng một cách hiệu quả và tiết kiệm nhằm bảo vệ môi trường và giảm chi phí sử dụng năng lượng. Trong báo cáo này, chúng tôi thực hiện xây dựng một hệ thống dự báo mức tiêu thụ năng lượng real-time trên bộ dữ liệu Electric Power Consumption. Chúng tôi đã tiến hành xử lí dữ liệu, sau đó áp dụng các phương pháp dự báo chuỗi thời gian đơn biến và đa biến với các mô hình: CNN, GRU, LSTM, BiLSTM. Sau khi lựa chọn mô hình với kết quả tốt nhất, chúng tôi tiến hành xây dựng hệ thống dự đoán năng lượng tiêu thụ theo thời gian thực.**

***Index Terms*—Preprocessing, Machine Learning, Data Analysis, Electric Power Consumption rediction, Apache Spark, Apache Kafka, Kafka Streaming**

I. GIỚI THIỆU

A. Giới thiệu đề tài

Năng lượng đóng vai trò quan trọng trong việc tăng trưởng kinh tế và bảo vệ môi trường. Do sự phát triển của công nghiệp cùng với sự phát triển của kinh tế xã hôi làm cho nhu cầu sử dụng năng lượng tăng lên nhanh chóng. Việc này có thể gây ra những tác động xấu đến môi trường, bao gồm tăng nhiệt độ và ô nhiễm không khí. Nhiều nước đang chuyển đổi sang nguồn năng lượng tái tạo và tự nhiên, xây dựng các hệ thông dự báo, thay đổi quy trình sản xuất của mình và giảm thiểu tác động đến môi trường.

Dự đoán tiêu thụ năng lượng đã trở thành một phần quan trọng của môi trường thông minh và bền vững. Với dự báo nhu cầu trong tương lai, sản xuất và phân phối năng lượng có thể được tối ưu hóa để đáp ứng nhu cầu của dân số ngày càng tăng.

Trong báo cáo này, trước tiên chúng tôi trình bày bộ dữ liệu "Electric Power Consumption" trong Phần 2. Hướng tiếp cận của hệ thống sẽ được mô tả chi tiết ở Phần 3. Ở Phần 4, chúng tôi sẽ tiến hành thực nghiệm với các mô hình và phương pháp khác nhau. Sau đó, so sánh kết quả đạt được để chọn ra mô hình tốt nhất để xây dựng hệ thống. Tiếp theo, chúng tôi sẽ tiến hành xây dựng hệ thống ở Phần 5. Cuối cùng, chúng tôi đưa ra kết luận và hướng phát triển ở Phần 6.

B. Vấn đề nghiên cứu

Vấn đề nghiên cứu về dự đoán Năng lượng tiêu thụ là một chủ đề quan trọng trong lĩnh vực năng lượng và quản lý tài nguyên. Các nghiên cứu này thường tập trung vào phát triển mô hình dự đoán có khả năng ước lượng lượng điện năng mà một hệ thống hay một khu vực cụ thể sẽ tiêu thụ trong tương lai, giúp người quản lý năng lượng lập kế hoạch để tăng cường hiệu quả và giảm thiểu chi phí, các hệ thống quản lý tải có thể được thiết kế để phản ứng tự động, giảm nguy cơ quá tải mạng lưới và cải thiện ổn định hệ thống.

C. Mục tiêu của bài báo cáo

Mục tiêu của đề tài này là xây dựng và triển khai một mô hình dự đoán năng lượng tiêu thụ ở khu vực. Cụ thể, đề tài tập trung vào các mục tiêu sau:

- **Xây Dựng Mô Hình Dự Đoán**: Phát triển một mô hình dự đoán chính xác về lượng năng lượng sẽ được tiêu thụ trong khu vực cụ thể. Điều này bao gồm việc lựa chọn và thử nghiệm các phương pháp máy học hoặc học sâu phù hợp để dự đoán nhu cầu năng lượng.
- **Dự Báo Theo Thời Gian**: Nghiên cứu sẽ cố gắng dự đoán tiêu thụ năng lượng theo từng khoảng thời gian cụ thể, chẳng hạn như ngày, tuần, hoặc theo giờ, để cung cấp thông tin chi tiết và hữu ích cho quản lý năng lượng.
- **Triển Khai và Ứng Dụng Thực Tế**: Mục tiêu cuối cùng là triển khai mô hình vào môi trường thực tế, đảm bảo tính khả thi và ứng dụng trong quản lý năng lượng thực tế của khu vực.

II. BỘ DỮ LIỆU

A. Bối cảnh hình thành của bộ dự liệu

Tétouan là một thành phố nằm ở phía bắc của Morocco, chiếm diện tích khoảng 10.375 km² và có dân số khoảng 550.374 người. Với việc tiêu thụ điện năng trở nên quan trọng đối với đất nước, ý tưởng là nghiên cứu các tác động ảnh hưởng đến mức tiêu thụ năng lượng. Bộ dữ liệu này thể hiện mức tiêu thụ năng lượng của thành phố Tétouan ở Morocco. Mạng lưới phân phối được cấp điện bởi 3 trạm khu vực, gọi là: Quads, Smir và Boussafou.

B. Mô tả dữ liệu và các thuộc tính

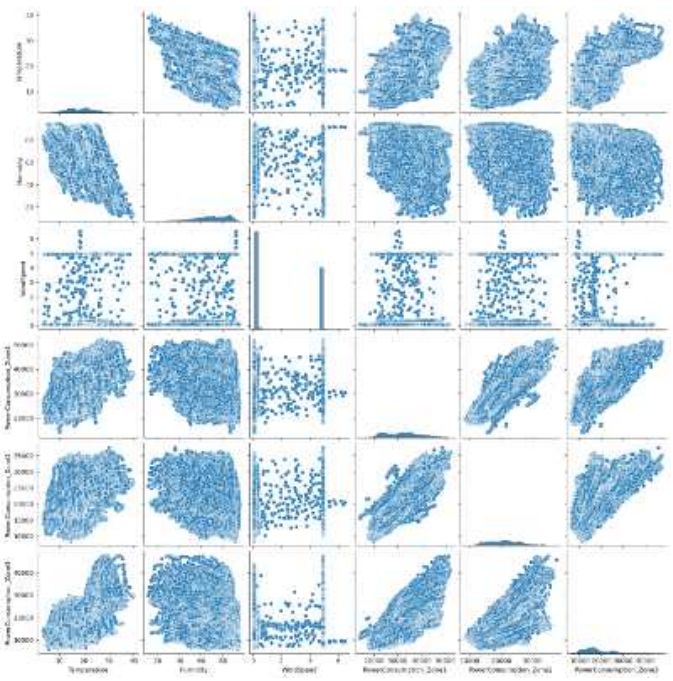
Bộ dữ liệu được sử dụng trong hệ thống dự đoán năng lượng này là **Electric Power Consumption** được thu thập từ Kaggle- được biết đến một nền tảng trực tuyến cho cộng đồng Khoa học dữ liệu, cho phép người dùng chia sẻ, tìm kiếm các bộ dữ liệu.

Bộ dữ liệu bao gồm 52,416 điểm dữ liệu với khoảng thời gian cập nhật 10 phút/lần về năng lượng được sử dụng. Mỗi quan sát được mô tả bởi 9 cột đặc trưng:

- **Date Time**: Thời gian.
- **Temperature**: Nhiệt độ thời tiết.
- **Humidity**: Độ ẩm thời tiết.
- **Wind Speed**: Tốc độ gió.
- **General Diffuse Flows**: 'Dòng phân tán' là một thuật ngữ tổng quát để mô tả các chất lỏng nhiệt độ thấp (< 0,2° đến 100°C) chảy với tốc độ chậm.
- **Diffuse Flows**: Thông tin về dòng phân tán.
- **Zone 1 Power Consumption**: Tiêu Thụ Điện Khu Vực 1.
- **Zone 2 Power Consumption**: Tiêu Thụ Điện Khu Vực 2.
- **Zone 3 Power Consumption**: Tiêu Thụ Điện Khu Vực 3.

C. Khám phá dữ liệu và phân tích đặc trưng

Sau khi vẽ biểu đồ và tiến hành tính toán, quan sát các giá trị như giá trị nhỏ nhất, lớn nhất, trung bình và phân vị cho biến liên tục và số lượng, phân bố, đặc điểm của các nhãn, nhóm rút ra được một số phân tích quan trọng sau về bộ dữ liệu:

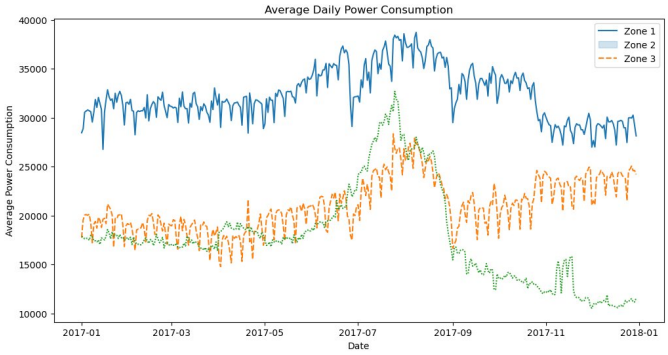


Hình 1. Mối quan hệ giữa các thuộc tính

Thời gian và Năng Lượng: Kiểm tra liệu có sự biến động đáng kể trong việc sử dụng năng lượng theo thời gian không. Điều này có thể giúp định rõ xu hướng hoặc sự biến động trong quá trình sử dụng điện.

Tương Quan giữa Nhiệt Độ và Độ Ẩm: Mối quan hệ dương: Nếu nhiệt độ tăng, độ ẩm có thể giảm và ngược lại. Điều này có thể phản ánh mối quan hệ tưởng chung trong thời tiết nhiệt đới.

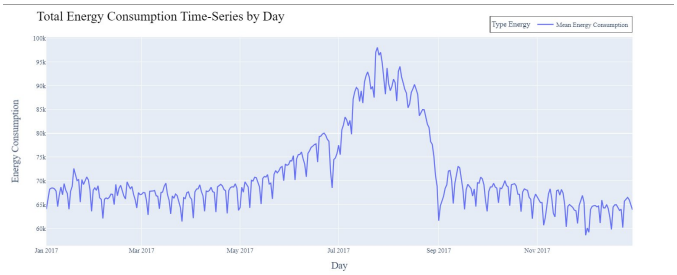
Tương Quan giữa Nhiệt Độ và Tốc Độ Gió: Mối quan hệ có thể là dương hoặc tiêu cực tùy thuộc vào địa hình. Ví dụ, nếu có một ngày nắng, nhiệt độ tăng có thể đi kèm với tăng tốc độ gió.



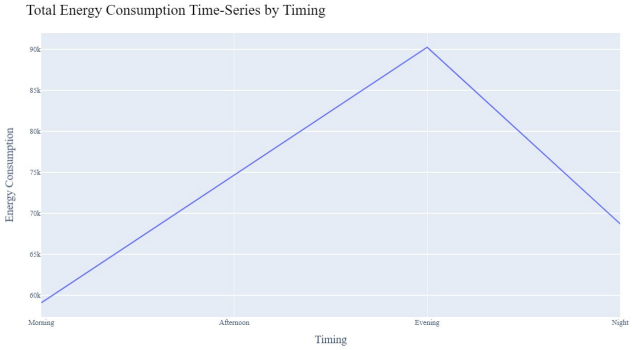
Hình 2. Năng lượng tiêu thụ trung bình mỗi ngày trong năm

Hình 2 cho chúng ta thấy năng lượng tiêu thụ trung bình mỗi ngày của 3 khu vực trong năm, có thể dễ dàng nhận thấy khu vực 1 có lượng năng lượng tiêu thụ lớn nhất, điều này có thể được giải thích vì trên thực tế khu vực 1 là khu có diện tích lớn nhất ở thành phố này.

Nhóm tiến hành gộp năng lượng tiêu thụ của cả 3 khu vực lại thành 1 cột tổng năng lượng tiêu thụ. Hình 3 cho chúng ta thấy rằng thành phố tiêu thụ điện nhiều nhất vào mùa hè (tháng 8) và ít nhất vào những tháng mùa đông. Ngoài ra, chúng ta có thể nhận thấy ở hình 4 rằng lượng tiêu thụ điện trong ngày đạt cực tiêu vào buổi sáng sớm và lên đến cực đại vào buổi chiều tối.

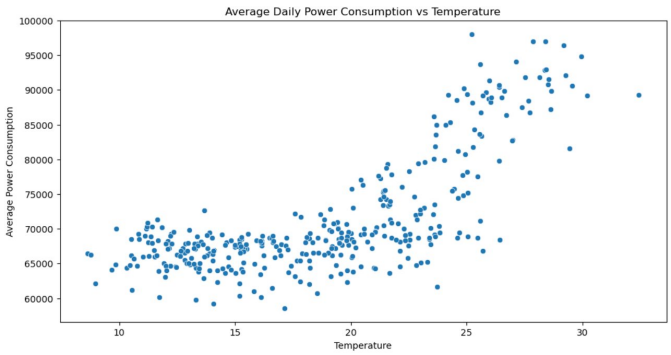


Hình 3. Năng lượng tiêu thụ của thành phố trong năm



Hình 4. Năng lượng tiêu thụ của thành phố trong ngày

Ngoài ra, khi nhiệt độ càng cao thì lượng tiêu thụ điện năng của thành phố càng lớn. Điều này cũng dễ hiểu bởi khi nhiệt độ lên cao chúng ta thường phải sử dụng thêm các thiết bị như điều hòa, quạt, ...



Hình 5. Mối quan hệ giữa nhiệt độ và lượng năng lượng tiêu thụ

III. Hướng Tiếp Cận

A. Univariate Time Series Forecasting

Univariate Time Series Forecasting (Dự báo chuỗi thời gian đơn biến – UTSF) là một bài toán dự báo dữ liệu thời gian mà chỉ có một chỉ số quan trọng được xem xét. Điều này có nghĩa là chỉ có một biến độc lập được sử dụng để dự báo giá trị tiếp theo. Ví dụ, trong bài toán dự báo năng lượng tiêu thụ, chỉ có năng lượng là một chỉ số quan trọng được xem xét, trong khi các chỉ số khác về thời tiết như: nhiệt độ, độ ẩm, tốc độ gió, v.v. đều được bỏ qua.

Có nhiều model được sử dụng cho UTSF, bao gồm: Prophet (Facebook's Time Series Forecasting Library)[1], Neural Network Models (CNN, LSTM, GRU, Feedforward, . . .), ...Tùy vào tính toán và mục tiêu, chúng ta có thể chọn một trong số các model trên để phù hợp với nhu cầu của mình.

B. Multivariate Time Series Forecasting

Multivariate Time Series Forecasting (Dự báo chuỗi thời gian đa biến – MTSF) là một bài toán dự báo dữ liệu thời gian mà nhiều chỉ số quan trọng được xem xét. Điều này có nghĩa là nhiều biến độc lập được sử dụng để dự báo giá trị tiếp theo. Ví dụ, trong bài toán dự báo năng lượng tiêu thụ, năng lượng cùng với các chỉ số về thời tiết như: nhiệt độ, độ ẩm, tốc độ gió, v.v. đều được xem xét là các chỉ số quan trọng để dự báo năng lượng tiêu thụ tiếp theo.

Bài toán dự báo chuỗi thời gian đã biến đòi hỏi sự tương tác giữa các biến được xem xét và yêu cầu sử dụng các mô hình phức tạp hơn so với bài toán dự báo đơn biến. Từ đó có thể giúp tăng độ chính xác của dự báo.

Có nhiều model có thể sử dụng cho MTSF, bao gồm: VAR (Vector Autoregression) [], Prophet (Facebook's Time Series Forecasting Library) [1], Neural Network Models (CNN, LSTM, GRU, Feedforward, . . .), ... Tùy vào tính toán và mục tiêu, chúng ta có thể chọn một trong số các model trên để phù hợp với nhu cầu của mình.

C. CNN

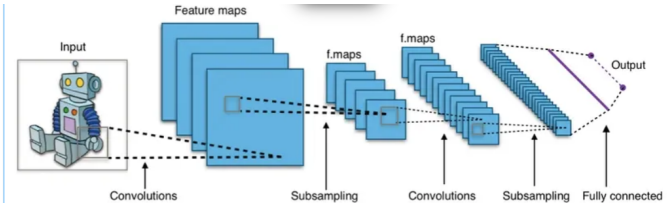
CNN được viết tắt của Convolutional Neural Network [3] hay còn được gọi là CNNS mang nơ-ron tích chập, là một trong những mô hình Deep Learning cực kỳ tiên tiến, bởi chúng cho phép bạn xây dựng những hệ thống có độ chính xác cao và thông minh. Nhờ khả năng đó, CNN có rất nhiều ứng dụng, đặc biệt là những bài toán cần nhận dạng vật thể (object) trong ảnh.

CNN vô cùng quan trọng để tạo nên những hệ thống nhận diện thông minh với độ chính xác cao trong thời đại công nghệ ngày nay. Lý do cụ thể vì sao CNN đặc biệt phát huy hiệu quả trong việc nhận dạng (detection). Mạng CNN là một trong những tập hợp của lớp Convolution được chồng lên nhau. Mạng CNN còn sử dụng các hàm nonlinear activation (như ReLU và tanh) nhằm kích hoạt trọng số trong node. Khi đã thông qua hàm, lớp này sẽ thu được trọng số trong các node và tạo ra nhiều thông tin trừu tượng hơn cho các lớp kế cận.

Đặc điểm mô hình CNN có 2 khía cạnh cần phải đặc biệt lưu ý là tính bất biến và tính kết hợp, do đó độ chính xác hoàn toàn có thể bị ảnh hưởng nếu có cùng một đối tượng được chiếu theo nhiều phương diện khác biệt. Với các loại chuyển dịch, co giãn và quay, người ta sẽ sử dụng pooli layer và làm bất biến những tính chất này. Từ đó, CNN sẽ cho ra kết quả có độ chính xác ứng với từng loại mô hình.

Pooling layer giúp tạo nên tính bất biến đối với phép dịch chuyển, phép co giãn và phép quay. Trong khi đó, tính kết hợp cục bộ sẽ thể hiện các cấp độ biểu diễn, thông tin từ mức độ thấp đến cao, cùng độ trừu tượng thông qua convolution từ các filter. Dựa trên cơ chế convolution, một mô hình sẽ liên kết được các layer với nhau.

Với cơ chế này, layer tiếp theo sẽ là kết quả được tạo ra từ convolution thuộc layer kế trước. Điều này đảm bảo bạn có được kết nối cục bộ hiệu quả nhất. Mỗi nơ-ron sinh ra ở lớp tiếp theo từ kết quả filter sẽ áp đặt lên vùng ảnh cục bộ của nơ-ron tương ứng trước đó. Cũng có một số layer khác như pooling/subsampling layer được dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).



Hình 6. Cách thức hoạt động của CNN

D. Gated Recurrent Unit

Recurrent Neural Network (RNN) [4] là một loại neural network hồi quy dựa trên việc sử dụng cùng một hàm tính

toán để xử lý các đầu vào liên tục. Trong RNN, trạng thái hiện tại được sử dụng để dự báo trạng thái tiếp theo và quá trình này được lặp đi lặp lại cho mỗi đầu vào trong chuỗi. Gated Recurrent Unit (GRU) [5] là một phiên bản cải tiến của RNN, được sử dụng trong học sâu cho các bài toán như phân loại văn bản, dự đoán chuỗi thời gian và giải quyết các bài toán Natural Language Processing (NLP) khác. GRU có cấu trúc gồm 2 cổng (gate): cổng cập nhật trạng thái (update gate) và cổng xóa trạng thái (reset gate). Các cổng này dùng để quản lý và giữ nguyên thông tin trong một chuỗi và cho phép model tự động chọn việc cập nhật hay loại bỏ thông tin cũ. Cụ thể, update gate sẽ xác định mức độ cập nhật thông tin mới vào hidden state, trong khi reset gate sẽ xác định mức độ loại bỏ thông tin cũ để chuẩn bị cho việc nhập thông tin mới. Sau đó, một giá trị mới của hidden state được tính toán dựa trên giá trị cũ và thông tin mới. Kết quả cuối cùng của GRU là giá trị của hidden state đó, sẽ được dùng để dự đoán kết quả. GRU có thể hoạt động tốt với dữ liệu time-series bằng cách giữ nguyên những thông tin quan trọng và loại bỏ những thông tin không cần thiết.

E. Long Short-Term Memory

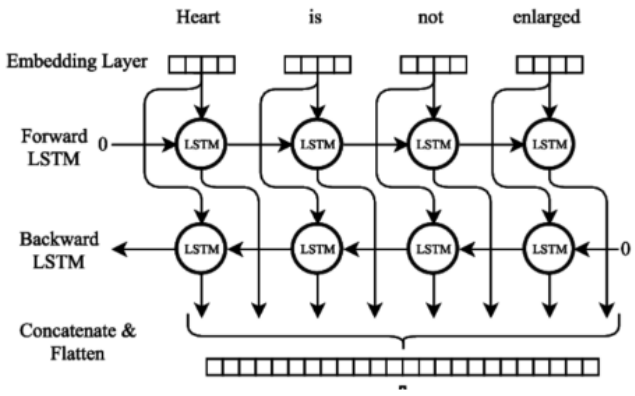
LSTM (Long Short-Term Memory) [6] là một loại mô hình RNN được thiết kế để xử lý dữ liệu chuỗi thời gian, trong đó một số thông tin của trạng thái trước được giữ lại để dùng cho các bước tiếp theo. So với RNN, LSTM có khả năng xử lý tốt hơn các bài toán với dữ liệu chuỗi thời gian dài, vì nó có thể giữ lại những thông tin quan trọng và loại bỏ những thông tin không cần thiết. So với GRU, LSTM có cấu trúc phức tạp hơn và yêu cầu nhiều tài nguyên hơn để huấn luyện, nhưng nó có khả năng xử lý tốt hơn một số bài toán NLP với dữ liệu chuỗi thời gian. LSTM có cấu trúc gồm:

- **Memory cell**: là phần chứa thông tin quan trọng trong thời gian dài.
- **Input gate**: điều khiển việc cập nhật thông tin vào memory cell.
- **Forget gate**: điều khiển việc xóa thông tin không cần thiết từ memory cell.
- **Output gate**: điều khiển việc truyền thông tin từ memory cell ra ngoài mô hình.

Các phần trên hoạt động đồng thời và những giá trị tính toán từ các neural network đều được sử dụng để cập nhật, xóa hoặc truyền thông tin từ memory cell ra ngoài mô hình. Điều này cho phép mô hình LSTM giữ lại những thông tin quan trọng trong thời gian dài và loại bỏ những thông tin không cần thiết.

F. Bidirectional LSTM

Bidirectional LSTM hoặc BiLSTM [7] là một thuật ngữ được sử dụng để mô tả một mô hình chuỗi có chứa hai lớp LSTM, một để xử lý đầu vào theo hướng chuyển tiếp và một để xử lý theo hướng ngược lại. Thông thường, nó được sử dụng trong các nhiệm vụ liên quan đến xử lý ngôn ngữ tự nhiên (NLP). Ý tưởng đằng sau phương pháp này là bằng cách xử lý dữ liệu theo cả hai hướng, mô hình có khả năng hiểu quan hệ giữa các chuỗi tốt hơn.



Hình 7. Mô hình hoạt động của BiLSTM

G. Apache Kafka

Apache Kafka [8] là một hệ thống mã nguồn mở để xử lý luồng dữ liệu từ các nguồn đầu vào và gửi đến các nguồn đầu ra, cung cấp một cơ chế tự động hóa, mã hóa và gửi lại dữ liệu. Nó cung cấp các tính năng như lưu trữ lịch sử, bảo mật, quản lý bản ghi, tìm kiếm và phân tích. Đặc biệt, Apache Kafka còn cung cấp tính năng mức độ đồng bộ cao, có thể mở rộng dễ dàng, và cho phép phân tán dữ liệu giữa nhiều máy chủ. Apache Kafka có 4 thành phần chính:

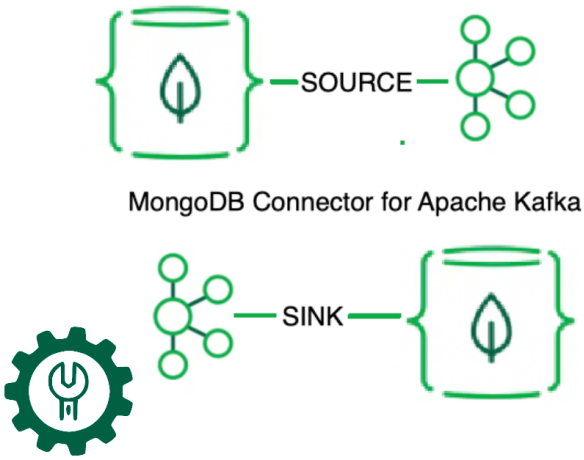
- Topics: Là một tập hợp các bản ghi sự kiện (records) được lưu trữ trong một topic. Mỗi topic là một đối tượng riêng biệt để chứa dữ liệu và cho phép cho các hệ thống khác đồng bộ với nó.
- Producers: Là các đối tượng gửi dữ liệu đến các topic. Producers có thể gửi dữ liệu đến nhiều topic cùng một lúc và đảm bảo tính toàn vẹn của dữ liệu truyền tải.
- Brokers: Là các nút trung gian trong một cluster Kafka, chịu trách nhiệm lưu trữ dữ liệu và trao đổi dữ liệu giữa các producers và consumers.
- Consumers: Là các đối tượng nhận dữ liệu từ các topic. Consumers có thể đăng ký với nhiều topic và đồng bộ với chúng để nhận dữ liệu.

Ngoài ra, Apache Kafka còn có một số thành phần khác như: Partitions, Replicas, và Zookeeper, nhưng 4 thành phần chính trên là các thành phần quantrọng nhất để hiểu sơ bộ về Apache Kafka. Kiến trúc của Apache Kafka đượcthể hiện trong Hình 2 bao gồm các nút broker tách biệt và liên kết với nhau,Producers gửi dữ liệu đến Topics, Consumers đọc dữ liệu từ Topics, và Zookeeperquản lý toàn bộ hệ thống.

H. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL, cho phép lưu trữ và truy xuất dữ liệu dạng Document. Nó hỗ trợ tính năng phân tán và tự động sharding, giúp tăng hiệu suất và khả năng mở rộng dữ liệu. MongoDB là một trong những hệ quản trị cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong các ứng dụng web và mobile. Có rất nhiều ưu điểm của MongoDB như: Dữ liệu dạng Document: MongoDB lưu trữ dữ liệu dạng document,

giúp tăng tính linh hoạt và dễ dàng mở rộng hơn so với các hệ quản trị cơ sở dữ liệu dạng bảng; Phân tán và Sharding tự động: MongoDB có tính năng phân tán và sharding tự động, giúp tăng hiệu suất và khả năng mở rộng dữ liệu một cách dễ dàng; Tính năng Map-Reduce: MongoDB cung cấp tính năng Map-Reduce, giúp cho việc thống kê và phân tích dữ liệu trở nên dễ dàng hơn; Tính năng Replication và failover : MongoDB có hệ thống quản lý bản sao (replication) và tự động failover, giúp tăng tin cậy hệ thống và khả năng dự phòng; Tương thích với nhiều ngôn ngữ lập trình: MongoDB tương thích với nhiều ngôn ngữ lập trình như Java, Python, Ruby, v.v., giúp cho việc sử dụng và mở rộng hệ thống trở nên dễ dàng hơn.



Hình 8. Mô hình MongoDB kết nối với Kafka

Vì data có khối lượng rất lớn nên việc lưu trữ và quản lý dữ liệu trở nên khó khăn, nhờ vào các tính năng vượt trội của MongoDB chúng tôi sử dụng nó để quản lý dữ liệu lớn này. Mục đích của chúng tôi sử dụng MongoDB trong hệ thống này còn vì muốn tận dụng khả năng multi node, khi hai hay nhiều máy đều chứa dữ liệu cần thì có thể truy cập P2P (peer-to-peer) để truy vấn dữ liệu qua lại, từ đó có thể lấy dữ liệu từ nhiều node 1 khác nhau, đó là bản chất của dữ liệu lớn vì nó có khối lượng rất lớn nên khả năng lưu trữ phân tán cao, dữ liệu khó mà tập trung tại 1 node.

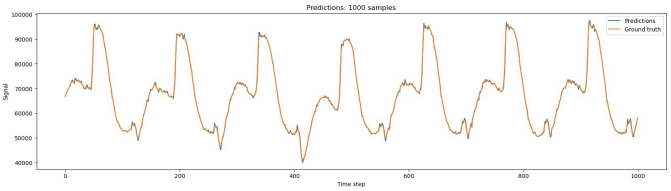
IV. Thực Nghiệm và Kết Quả

Chúng tôi tiến hành xây các model: CNN, GRU, LSTM, BiLSTM. Với các model, chúng tôi tiến hành thực nghiệm với các phương pháp UTSF và MTSF. Trong MTSF gồm: MTSF sử dụng biến nhiệt độ (temperature); MTSF sử dụng biến nhiệt độ và độ ẩm (temperature + humidity); MTSF sử dụng biến nhiệt độ, độ ẩm và tốc độ gió (temperature + humidity + windSpeed). Để tránh tính ngẫu nhiên của kết quả thực nghiệm, chúng tôi thực hiện 5 lần chạy và ghi lại kết quả, sau đó tính kết quả trung bình. Bảng 1 trình bày kết quả so sánh hiệu suất RMSE trung bình của các model. Với phương pháp UTSF thì mô hình GRU đạt kết quả tốt nhất theo độ đo RMSE với kết quả RMSE = 0.6010. Với phương pháp MTSF sử dụng biến nhiệt độ, mô hình BiLSTM cho kết quả tốt nhất với RMSE = 0.5623; Khi thêm biến nhiệt độ, nhìn chung các mô hình có kết quả tốt hơn so với phương pháp UTSF, đặc biệt là mô hình BiLSTM và LSTM. Với phương pháp MTSF sử dụng biến nhiệt độ và độ ẩm, mô hình LSTM cho kết quả tốt nhất với RMSE = 0.5747, tuy nhiên nhìn chung các mô hình cho kết quả thấp hơn. Cuối cùng, chúng tôi sử dụng cả 3 biến nhiệt độ, độ ẩm và tốc độ gió, mô hình GRU tiếp tục cho kết quả tốt nhất với RMSE = 0.5619, đây cũng là kết quả tốt nhất mà chúng tôi đạt được.

	CNN	GRU	LSTM	BiLSTM
UTSF	0.6319	0.6010	0.8702	0.6493
MTSF(T)	0.6634	0.5713	0.5816	0.5623
MTSF(T+H)	0.6746	0.6088	0.5747	0.6788
MTSF(T+H+W)	0.6479	0.5619	0.6189	0.6749

Bảng I
Kết quả của các mô hình

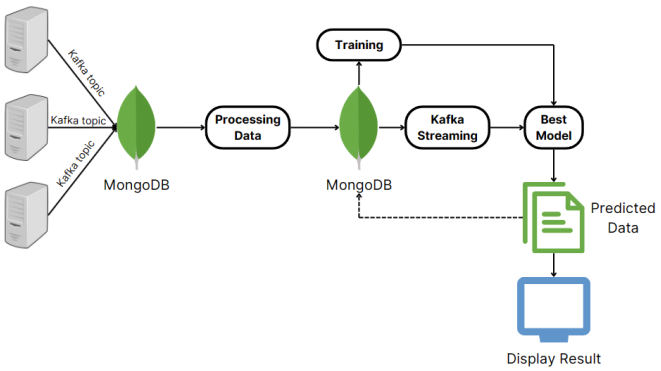
Hình 9 thể hiện kết quả thực nghiệm của các mô hình tốt nhất trên 1000 điểm dữ liệu của tập test. Có thể thấy rằng kết quả dự đoán của các mô hình khá tốt, giá trị năng lượng tiêu thụ được dự đoán gần đúng với giá trị năng lượng tiêu thụ thực tế. Bên cạnh đó, kết quả dự đoán của các mô hình khá tương đồng nhau. Dựa vào các kết quả trên, chúng tôi áp dụng mô hình GRUGRU với phương pháp MTSF sử dụng biến nhiệt độ, độ ẩm và tốc độ gió để xây dựng hệ thống trong phần tiếp theo.



Hình 9. Kết quả của mô hình CNN trên tập test

V. Kiến trúc hệ thống

Kiến trúc hệ thống được chia làm ba giai đoạn chính, được thể hiện trong Hình 1010:

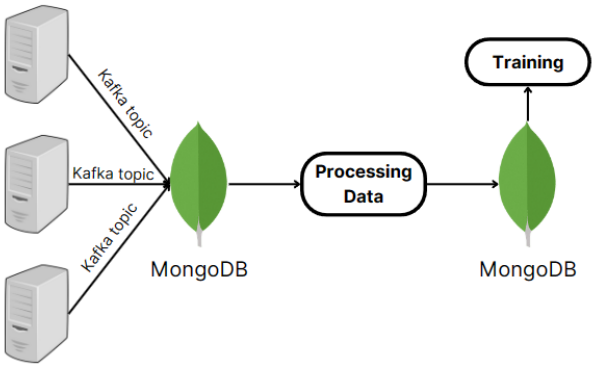


Hình 10. General architecture of system

A. Giai Đoạn Xây Dựng và Lựa Chọn Mô Hình

Dữ liệu được tổng hợp từ nhiều nguồn khác nhau sẽ được Producer Kafka gửi vào Topic Kafka. Sau đó Consumer Kafka

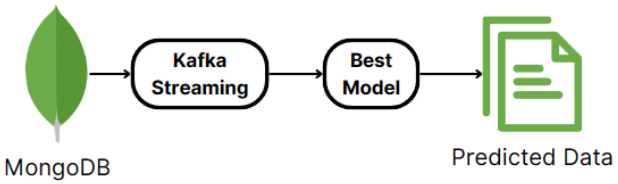
có vai trò chuyển nhận dữ liệu và truyền dữ liệu trực tiếp vào cơ sở dữ liệu MongoDB để lưu trữ. Dữ liệu thô sau khi được lưu trữ vào MongoDB sẽ được load ra để tiến hành các bước tiền xử lý dữ liệu để thu được một bộ dữ liệu sạch. Sau đó, dữ liệu sẽ được lưu lại thành hai bộ dữ liệu gồm: Traing (dữ liệu dùng để huấn luyện các mô hình) và Streaming (dữ liệu dùng để giả lập hệ thống real-time) vào MongoDB. Tiếp theo, dữ liệu Training được load từ MongoDB để tiến hành huấn luyện các mô hình và chọ ra mô hình tốt nhất để sử dụng làm mô hình chính của hệ thống này. Hình 11 thể hiện đầy đủ chi tiết của giai đoạn này. Các bước xử lý dữ liệu, thực nghiệm huấn luyện mô hình ở Phần 4 cho thấy mô hình GRUGRU với phương pháp MTSF sử dụng biến nhiệt độ, độ ẩm và tốc độ gió đạt kết quả tốt nhất, nên chúng tôi sẽ sử dụng mô hình CNN cho hệ thống này.



Hình 11. Giai Đoạn Xây Dựng và Lựa Chọn Mô Hình.

B. Giai Đoạn Stream Dữ Liệu

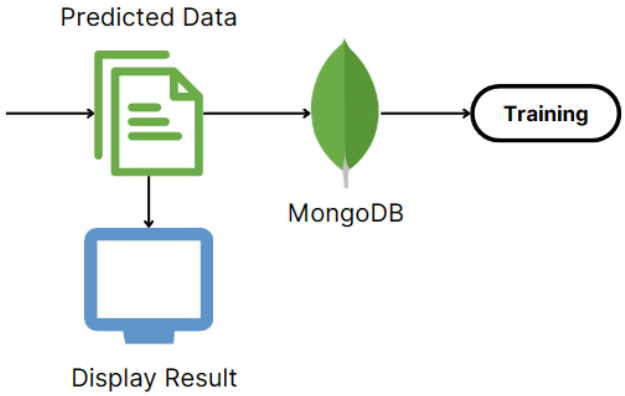
Ở phần này, chúng tôi sử dụng Apache Spark để load dữ liệu Streaming từ MongoDB để giả lập real-time bằng cách dùng Kafka Streaming. Dữ liệu sau khi được load ra sẽ được Producer Kafka gửi vào Topic Kafka, sau đó Consumer Kafka sẽ nhận dữ liệu liên tục từ Producer Kafka để đưa đến mô hình tiến hành quá trình dự đoán. Ở đây chúng tôi sử dụng MongoDB kết hợp với Kafka Streaming để hệ thống cho tốc độ nhận tin và xử lý liên tục. Kafka Streaming giúp xử lý dữ liệu thời gian thực từ nhiều nguồn khác nhau, được sử dụng cho dữ liệu thời gian thực có thể là dữ liệu phi cấu trúc như hình ảnh hoặc văn bản. Hình 12 thể hiện chi tiết giai đoạn giải lập real-time.



Hình 12. Giai Đoạn Stream dữ liệu.

C. Giai Đoạn Cập Nhật Mô Hình

Dữ liệu được dự đoán xong sẽ được lưu lại vào MongoDB để tiến hành cập nhật lại mô hình sau này. Đồng thời dữ liệu dự đoán cũng sẽ được đưa ra màn hình để người dùng có thể nhìn trực quan kết quả. Hình 13 mô phỏng lại kiến trúc ở giai đoạn này.



Hình 13. Giai Đoạn cập nhật mô hình.

VI. Kết luận

Trong báo cáo này, chúng tôi đã sử dụng bộ dữ liệu Electric Power Consumption để xây dựng hệ thống dự đoán năng lượng tiêu thụ. Qua quá trình xử lý dữ liệu, lựa chọn phương pháp, và thực hiện các thử nghiệm, chúng tôi đã tìm ra kết quả tốt nhất. Mục tiêu là xây dựng hệ thống dự đoán năng lượng tiêu thụ real-time cho thành phố, từ đó giúp cho thành phố có thể quản lý năng lượng hiệu, giảm thiểu lãng phí và tối ưu hóa việc sử dụng năng lượng.

Chúng tôi đã áp dụng nhiều phương pháp dự báo chuỗi thời gian đơn biến và đa biến, bao gồm các mô hình như CNN, GRU, lSTM và BiLSTM. Để đánh giá hiệu suất, chúng tôi sử dụng độ đo RMSE. Kết quả tốt nhất mà chúng tôi đạt được là 0.5619, sử dụng mô hình GRU.

Tuy nhiên, hệ thống của chúng tôi vẫn còn một số hạn chế, bao gồm kết quả dự đoán chưa đạt được mức cao mong muốn và hiệu suất vận hành chưa đạt được độ ổn định mong đợi.

Hướng phát triển trong tương lai:

- Áp dụng thêm nhiều mô hình khác nhau như VGG16, VGG19, ResNet50 để có thêm lựa chọn phương pháp tốt nhất cho bài toán dự báo chuỗi thời gian.
- Kết hợp nhiều thuộc tính khác nhau trong bộ dữ liệu để cải thiện độ chính xác của kết quả dự đoán.
- Xây dựng mô hình dự đoán thời tiết Real-time cho thành phố

Tài liệu

[1] Taylor, Sean J., and Benjamin Letham. "Forecasting at scale."The American Statistician 72, no. 1 (2018): 37-45.

[2] Stock, James H., and Mark W. Watson. "Vector autoregressions."Journal of Economic perspectives 15, no. 4 (2001): 101-115.

[3] Zhao, Bendong, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. "Convolutional neural networks for time series classification."Journal of Systems Engineering and Electronics 28, no. 1 (2017): 162-169.

[4] Husken, Michael, and Peter Stagge. "Recurrent neural networks for time series classification."Neurocomputing 50 (2003): 223-235.
[5] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling."arXiv preprint arXiv:1412.3555 (2014).
[6] Hua, Yuxiu, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. "Deep learning with long short-term memory for time series prediction."IEEE Communications Magazine 57, no. 6 (2019): 114-119.
[7] Sima Siami-Namini, Neda Tavakoli and Akbar Siami Namin. "The Performance of LSTM and BiLSTM in Forecasting Time Series."IEEE International Conference on Big Data
[8] Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing."In Proceedings of the NetDB, vol. 11, no. 2011, pp. 1-7. 2011.