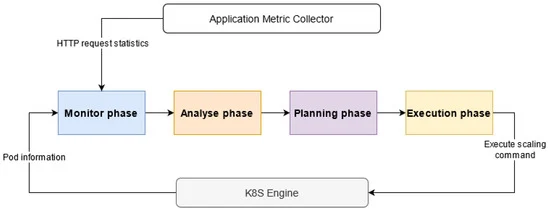
**1 Cấu trúc các phase của hệ thống**



**1.1 Monitor phase**

Giám sát K8s

**1.2 Analyse phase**

Trong giai đoạn phân tích, mô hình dự báo Bi-LSTM thu thập dữ liệu số liệu được thu thập từ Monitor phase với kích thước cửa sổ w. Sau đó, mô hình được huấn luyện sẽ sử dụng những dữ liệu từ các bước thời gian 𝑑𝑡−𝑤,...,𝑑𝑡−2,𝑑𝑡−1,𝑑𝑡 để dự đoán dữ liệu ở bước thời gian 𝑑𝑡+1. Mô hình Bi-LSTM đã được sử dụng để dự đoán workload là số lượng HTTP requests trong tương lai.

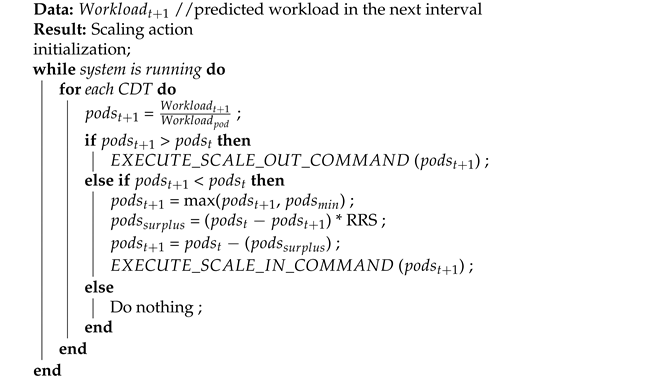
Đầu tiên, chúng ta thảo luận về mô hình LSTM thuần thuý. LSTM là một mô hình chuỗi thời gian, được sử dụng trong nhiều lĩnh vực, bao gồm phân tích cảm xúc, nhận dạng giọng nói và phân tích chuỗi thời gian. Nó là một phiên bản nâng cấp của kiến ​​trúc RNN, được Hochreiter và Schmidhuber giới thiệu lần đầu tiên vào năm 1996 và được phổ biến bởi nhiều nghiên cứu sau đó. Không giống như mạng nơ ron chuyển tiếp, mạng RNN lấy dữ liệu theo bước thời gian hiện tại và trước đó làm đầu vào để xây dựng mạng. Các mạng này có vòng lặp cho phép duy trì dữ liệu lịch sử. Tuy nhiên, RNN bị ảnh hưởng sự phụ thuộc lâu dài do vấn đề triệt tiêu đạo hàm (vanishing gradient) do trong RNN gradient tăng theo cấp số nhân theo thời gian. Để giải quyết vấn đề này, LSTM, một loại RNN đặc biệt, đã được đề xuất. Tuy nhiên, LSTM chỉ có thể xử lý thông tin theo một hướng. Đây là một hạn chế vì nó bỏ qua những thay đổi dữ liệu liên tục, khiến cho mạng chỉ có thể nắm bắt được một phần thông tin.

RNN hoặc LSTM thông thường xử lý chuỗi đầu vào theo chiều thuận của chuỗi thời gian. Bi-LSTM là mô hình mở rộng của LSTM, với hai LSTM được áp dụng cho dữ liệu đầu vào. LSTM đầu tiên được huấn luyện với chuỗi đầu vào chiều chuyển tiếp, LSTM thứ hai được thiết lập với dạng đảo ngược của chuỗi đầu vào. Như vậy, kiến trúc Bi-LSTM là hai mạng LSTM xếp chồng lên nhau. Nó chạy hai chiều đầu vào từ hai hướng, hướng thứ nhất từ ​​quá khứ đến tương lai, hướng thứ hai từ tương lai đến quá khứ. Sự khác biệt so với LSTM thuần thuý là việc chạy lùi giúp bảo toàn thông tin từ tương lai. Bằng cách kết hợp hai trạng thái ẩn từ hai chiều, ta có thể lưu giữ thông tin từ quá khứ và tương lai. Áp dụng LSTM hai lần sẽ cải thiện sự phụ thuộc lâu dài trong học tập và do đó sẽ cải thiện độ chính xác của mô hình.

**1.3 Planning phase**

Giai đoạn này cần tính toán số lượng pods cần thiết để cung cấp hoặc hủy cung cấp dựa trên khối lượng HTTP requests từ các bước trước đó. Nó được thiết kế để mở rộng hoặc thu hẹp quy mô hệ thống nhằm đáp ứng khối lượng công việc trong tương lai. Thuật toán sau đây sẽ giải thích chiến lược mở rộng quy mô sau khi mô hình Bi-LSTM cung cấp số lượng HTTP requests tại bước thời gian tiếp theo, 𝑊𝑜𝑟𝑘𝑙𝑜𝑎𝑑𝑡+1. Do vấn đề dao động, bộ chia tỷ lệ có thể thực hiện các hành động mở rộng hoặc thu hẹp trong một khoảng thời gian ngắn, gây lãng phí tài nguyên và chi phí. Để giải quyết vấn đề này, ta đặt thời gian hạ nhiệt (cooling down time) thành 1 phút sau mỗi quyết định chia tỷ lệ. Giả sử 𝑊𝑜𝑟𝑘𝑙𝑜𝑎𝑑𝑝𝑜𝑑 là khối lượng công việc tối đa mà một pod có thể xử lý trong một phút và 𝑝𝑜𝑑𝑠𝑚𝑖𝑛 là số lượng pod tối thiểu phải được duy trì để vận hành hệ thống và không thể giảm quy mô số pod thấp hơn giá trị này. Trong quá trình vận hành, cứ mỗi phút tính toán số lượng pod cần thiết, 𝑝𝑜𝑑𝑠𝑡+1 , trong bước thời gian tiếp theo và so sánh nó với nhóm hiện tại, 𝑝𝑜𝑑𝑠𝑡. Nếu số lượng pods yêu cầu vượt quá số lượng pods hiện tại 𝑝𝑜𝑑𝑠𝑡, lệnh mở rộng quy mô sẽ được kích hoạt và bộ chia tỷ lệ sẽ tăng số lượng pods để đáp ứng nhu cầu tài nguyên trong tương lai gần. Mặt khác, nếu số lượng nhóm được yêu cầu thấp hơn số nhóm hiện tại, bộ chia tỷ lệ sẽ loại bỏ một lượng pods dư thừa, 𝑝𝑜𝑑𝑠𝑠𝑢𝑟𝑝𝑙𝑢𝑠 , tuân theo chiến lược loại bỏ tài nguyên (RRS), để ổn định hệ thống đồng thời xử lý hệ thống nhanh hơn nếu khối lượng công việc tăng đột biến trong khoảng thời gian tiếp theo. Đầu tiên, chúng ta sử dụng hàm max để chọn giá trị cao hơn giữa 𝑝𝑜𝑑𝑠𝑡+1 và 𝑝𝑜𝑑𝑠𝑚𝑖𝑛 để cập nhật giá trị 𝑝𝑜𝑑𝑠𝑡+1. Điều này phải được thực hiện vì hệ thống không thể giảm quy mô xuống số thấp hơn giá trị 𝑝𝑜𝑑𝑠𝑚𝑖𝑛. Thứ hai, số lượng nhóm dư thừa, tuân theo RRS, được tính theo Công thức:

𝑝𝑜𝑑𝑠𝑠𝑢𝑟𝑝𝑙𝑢𝑠=(𝑝𝑜𝑑𝑠𝑡−𝑝𝑜𝑑𝑠𝑡+1)\*𝑅𝑅𝑆;



Sau đó, chúng ta cập nhật giá trị của 𝑝𝑜𝑑𝑠𝑡+1 bằng cách sử dụng giá trị của nhóm hiện tại 𝑝𝑜𝑑𝑠𝑡 trừ đi giá trị của các pod dư thừa 𝑝𝑜𝑑𝑠𝑠𝑢𝑟𝑝𝑙𝑢𝑠 đạt được ở bước trước. Bằng cách này, chúng ta chỉ cần loại bỏ một phần tài nguyên dư thừa (nhóm) chứ không phải tất cả. Cuối cùng, bộ chia tỷ lệ cập nhật 𝑝𝑜𝑑𝑠𝑡+1. Sử dụng chiến lược RRS không chỉ giúp hệ thống ổn định với khối lượng công việc thấp mà còn thích ứng nhanh hơn với khối lượng công việc tăng đột biến vì mất ít thời gian hơn để tạo và cập nhật số lượng nhóm. Trong các thử nghiệm, ta đặt RRS thành 0,60 (60%) và 𝑝𝑜𝑑𝑠𝑚𝑖𝑛 là 10 và giả sử 𝑊𝑜𝑟𝑘𝑙𝑜𝑎𝑑𝑝𝑜𝑑 là 300 (ứng với 5 requests trên giây).

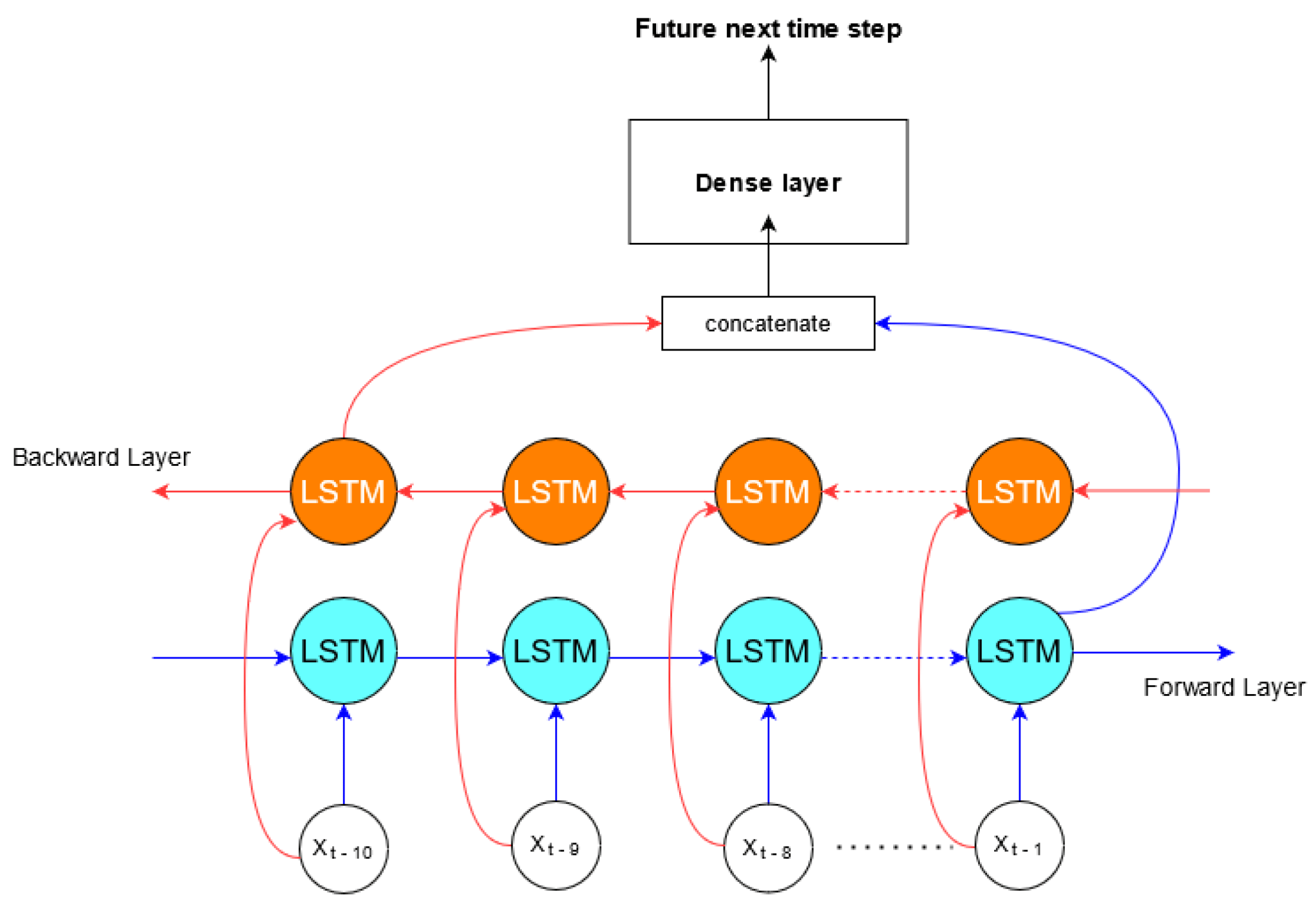
**1.4 Execution phase**

Kubernetes (kube-apiserver) nhận lệnh từ Planning phase và thay đổi số lượng replica pods.

**2 Mô hình dự báo Bi-LSTM**

**2.1 Kiến trúc mạng**

Mô hình Bi-LSTM được đề xuất như hình sau, có 2 layer tiến và lùi. Mỗi layer ẩn chứa một lớp đầu vào có 10 ô thần kinh cho 10 bước thời gian, với 30 đơn vị ẩn cho mỗi nơ-ron. Đầu ra cuối cùng của các lớp ẩn theo cả hướng tiến và lùi được nối với nhau và được sử dụng làm đầu vào cho lớp Dense. Lớp Dense là một lớp được kết nối đầy đủ, có nghĩa là mỗi nơ-ron trong Dense nhận được đầu vào từ tất cả các nơ-ron của lớp trước đó. Hình dạng đầu ra của Dense sẽ bị ảnh hưởng bởi số lượng nơ-ron được chỉ định trong đó. Trong đồ án này, Dense được sử dụng để đưa ra dự đoán. Vì mô hình là mô hình hồi quy phi tuyến tính nên sử dụng hàm kích hoạt ReLU ở lớp ẩn.

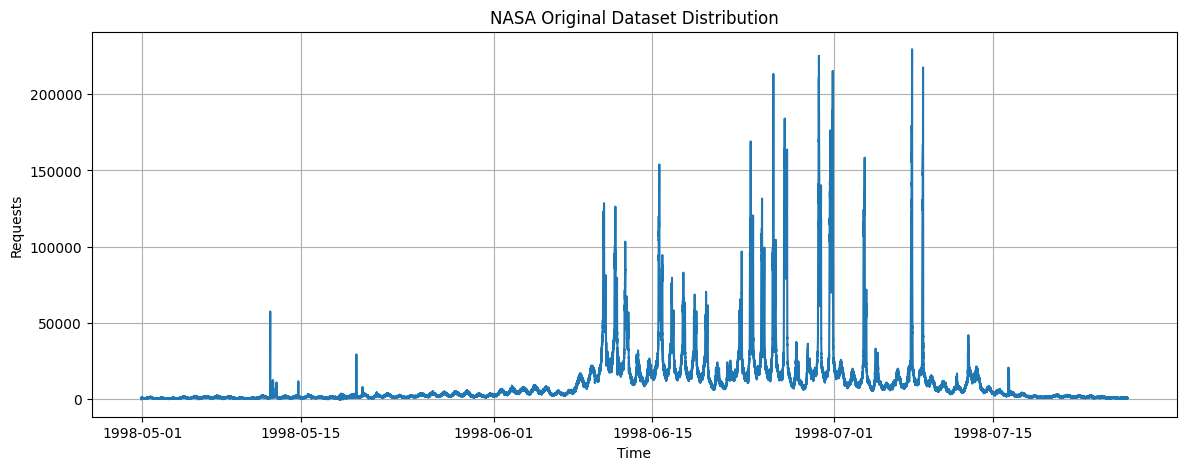
****

**2.2 Thử nghiệm và đánh giá**

Trong phần này, ta đánh giá thiết kế hệ thống được đề xuất được trình bày trong Phần 1. Đầu tiên, mô hình Bi-LSTM được đề xuất được đánh giá và độ chính xác so sánh với mô hình RNN và LSTM thuần thuý bằng cách sử dụng bộ dữ liệu FIFA World Cup 98 web servers.

**2.2.1 Chuẩn bị dữ liệu và cấu hình tham số**

Bộ dữ liệu FIFA World Cup chứa 1.352.804.107 HTTP requets, được ghi lại từ ngày 30 tháng 4 năm 1998 đến ngày 26 tháng 7 năm 1998. Bộ dữ liệu này đã được sử dụng rộng rãi để nghiên cứu các bộ chia tỷ lệ tài nguyên tự động. Tập dữ liệu này cũng được tiền xử lý trước bằng cách tổng hợp tất cả các yêu cầu HTTP trong cùng một phút thành một bản ghi tích lũy.

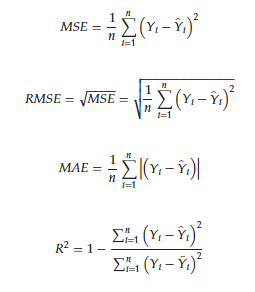


Chia tập dữ liệu thành 70% cho tập huấn luyện và 30% cho tập kiểm thử. Mô hình dự đoán Bi-LSTM được so sánh với các mô hình dự đoán RNN và LSTM. Cấu hình của các mô hình đều thiết lập tương tự nhau như bản sau. Các mô hình đều được triển khai bằng thư viện Keras từ framework TensorFlow, ngôn ngữ lập trình Python và được huấn luyện trên GPU Tesla T4 x2 của Kaggle.

| Số lượng input | 10 |
| --- | --- |
| Số lượng đơn vị ẩn trên mỗi nơ-ron | 30 |
| Hàm mất mát | MSE |
| Batch-size | 64 |
| Epochs | 20 |
| Hàm kích hoạt | ReLU |

**2.2.2 Phương pháp đánh giá**

Để đánh giá độ chính xác của mô hình được đề xuất, ta đã sử dụng bốn số liệu. Các số liệu này được sử dụng để so sánh độ chính xác dự đoán của các mô hình. Định nghĩa 𝐘𝐢 là giá trị thực tế, 𝐘̂ 𝐢 là giá trị dự báo và 𝐘¯𝐢 là giá trị trung bình của 𝐘. Ngoài ra, chúng tôi cũng so sánh tốc độ dự đoán trung bình của 30 lần thử mà mỗi mô hình thực hiện để dự đoán. Sai số bình phương trung bình (MSE) là tính toán chênh lệch bình phương trung bình giữa giá trị dự báo và giá trị quan sát được. Sai số bình phương trung bình gốc (RMSE), biểu thị căn bậc hai của hai trong số những khác biệt giữa giá trị dự báo và giá trị quan sát được. Sai số tuyệt đối trung bình (MAE) biểu thị mức trung bình trên mẫu thử nghiệm của những khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả những khác biệt riêng lẻ đều có trọng số bằng nhau. Các số liệu MSE, RMSE, MAE được sử dụng để đánh giá lỗi dự đoán của các mô hình, trong đó giá trị nhỏ hơn biểu thị độ chính xác dự đoán cao hơn và ngược lại. Hệ số xác định, ký hiệu là 𝑅2, được thể hiện như một thước đo mức độ phù hợp cho các mô hình hồi quy tuyến tính. Nó là bình phương của các hệ số tương quan giữa các kết quả được quan sát và các giá trị dự đoán được quan sát. 𝑅2 càng cao, mô hình càng phù hợp với dữ liệu.

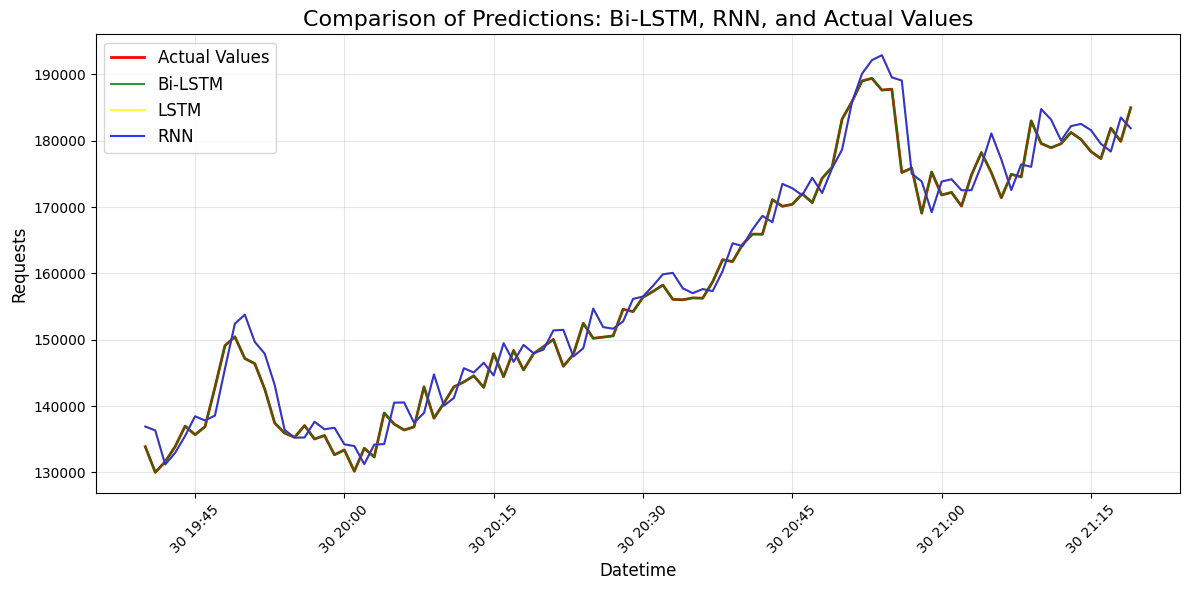


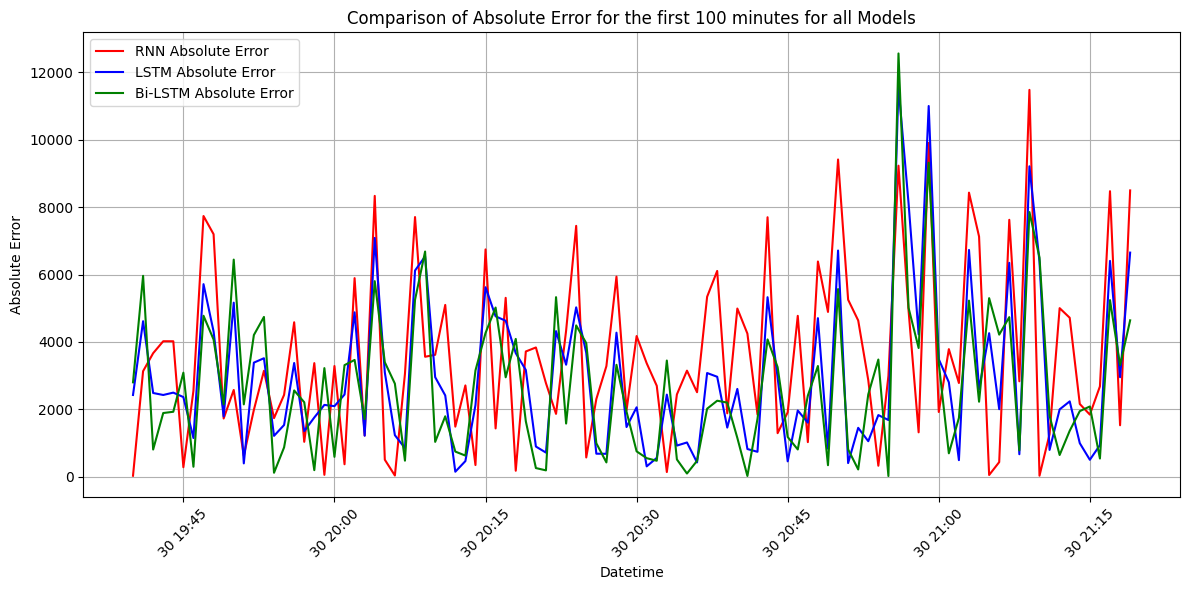
**2.2.3 Kết quả mô hình**

Mô hình Bi-LSTM đạt được các giá trị lỗi dự đoán nhỏ hơn trên các số liệu MSE, RMSE, MAE so với mô hình RNN, LSTM. Hơn nữa, 𝑅2 của Bi-LSTM đều cao hơn so với mô hình RNN, LSTM, điều này cho thấy Bi-LSTM có thể phù hợp với dữ liệu tốt hơn các mô hình còn lại.

|  | RNN | LSTM | Bi-LSTM |
| --- | --- | --- | --- |
| MSE | 762796.61030 | 492469.91672 | **482898.32523** |
| RMSE | 873.38228 | 701.76201 | **694.90886** |
| MAE | 604.79253 | 392.31183 | **361.14352** |
| R2 | 0.99782 | 0.99859 | **0.99862** |

Hình sau trình bày kết quả của giá trị dự đoán so với giá trị thực tế. Chúng ta có thể quan sát thấy mô hình RNN, LSTM thường có giá trị sai số tuyệt đối cao hơn mô hình Bi-LSTM.





**2.2.3 Kết quả scale pods**

