 **NHẬN XÉT CỦA GIẢNG VIÊN**

Người hướng dẫn:

Thái Bảo Trân

Nhóm 4 :

|  |  |
| --- | --- |
| Trần Phúc Thịnh  Nguyễn Thị Vân | 21521475  20522144 |
| Nguyễn Thị Mỹ Huyền | 20520559 |

**Ngày 04 tháng 06 năm 2024, TP. Hồ Chí Minh**

**BÁO CÁO CUỐI KỲ**

**MẠNG XÃ HỘI**

**ĐỀ TÀI: PHÁT HIỆN TIN TỨC GIẢ TRÊN MẠNG XÃ HỘI BẰNG MÔ HÌNH LSTM**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

# MỤC LỤC

[MỤC LỤC 3](#_Toc168678646)

[MỤC LỤC HÌNH ẢNH 5](#_Toc168678647)

[BẢNG PHÂN CÔNG 7](#_Toc168678648)

[I. GIỚI THIỆU 8](#_Toc168678649)

[1. Lý do chọn đề tài 8](#_Toc168678650)

[2. Tính cấp thiết của đề tài 8](#_Toc168678651)

[II. LÝ DO CHỌN MÔ HÌNH LSTM 9](#_Toc168678652)

[III. MÔ HÌNH TOÁN HỌC LSTM PHÂN LOẠI ĐỂ PHÁT HIỆN TIN GIẢ 10](#_Toc168678653)

[1. Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) 10](#_Toc168678654)

[1.1 Đặc điểm chính 10](#_Toc168678655)

[1.2 Kiến trúc cơ bản 11](#_Toc168678656)

[1.3 Các biến thể của RNN 11](#_Toc168678657)

[1.4 Thách thức của RNN 12](#_Toc168678658)

[2. Lý thuyết mô hình LSTM 12](#_Toc168678659)

[2.1 Giới thiệu 12](#_Toc168678660)

[2.2 Cấu trúc mô hình LSTM 12](#_Toc168678661)

[2.3 Ưu Điểm của LSTM 15](#_Toc168678662)

[IV. TỔNG QUAN VỀ DỮ LIỆU ĐÀO TẠO 16](#_Toc168678663)

[1. Dataset 16](#_Toc168678664)

[1.1 Đánh giá theo ngày tạo tài khoản 18](#_Toc168678665)

[1.2 Đánh giá theo tỉ lệ người dùng thật hay bot 20](#_Toc168678666)

[1.3 Đánh giá theo tỉ lệ cảm xúc của các phản hồi 20](#_Toc168678667)

[1.4 Đánh giá theo tỉ lệ lượt thích, retweet, phản hồi của bài đăng 21](#_Toc168678668)

[1.5 Đánh giá theo sự phân bổ số lượng người theo dõi tài khoản đăng tin 22](#_Toc168678669)

[1.6 Đánh giá theo tỉ lệ tiếp cận tin tức thật so với giả 23](#_Toc168678670)

[1.7 Đánh giá theo phân bố không gian của người đăng tin giả và tin thật 24](#_Toc168678671)

[V. CHI TIẾT THỰC HIỆN 25](#_Toc168678672)

[1. Data 25](#_Toc168678673)

[2. Phân loại bằng LSTM 30](#_Toc168678674)

[VI. KẾT QUẢ NHẬN ĐƯỢC 49](#_Toc168678675)

[VII. PHẦN KẾT LUẬN 51](#_Toc168678676)

[VIII. TÀI LIỆU THAM KHẢO 52](#_Toc168678677)

# MỤC LỤC HÌNH ẢNH

[Hình 1. Độ chính xác của các thuật dùng trong phát hiện tin tức giả mạo 10](#_Toc168678786)

[Hình 2. Công thức cơ bản của RNN 11](#_Toc168678787)

[Hình 3. Cổng quên 13](#_Toc168678788)

[Hình 4. Cổng đầu vào 13](#_Toc168678789)

[Hình 5. Cổng đầu ra 14](#_Toc168678790)

[Hình 6. Model của LSTM 14](#_Toc168678791)

[Hình 7. Bi-LSTM visualization 15](#_Toc168678792)

[Hình 8. Đánh giá theo ngày tạo tài khoản của PolitiFact 18](#_Toc168678793)

[Hình 9. Đánh giá theo ngày tạo tài khoản của GossipCop 19](#_Toc168678794)

[Hình 10. Đánh giá theo tỉ lệ người dùng thật hay bot 20](#_Toc168678795)

[Hình 11. Đánh giá theo tỉ lệ cảm xúc của các phản hồi 20](#_Toc168678796)

[Hình 12. Đánh giá theo tỉ lệ lượt thích, retweet, phản hồi của bài đăng 21](#_Toc168678797)

[Hình 13. Đánh giá theo sự phân bổ số lượng người theo dõi tài khoản đăng tin 22](#_Toc168678798)

[Hình 14. Đánh giá theo tỉ lệ tiếp cận tin tức thật so với giả 23](#_Toc168678799)

[Hình 15. Đánh giá theo phân bố không gian của người đăng tin giả và tin thật 24](#_Toc168678800)

[Hình 16. Data 25](#_Toc168678801)

[Hình 17. Data bước 1 25](#_Toc168678802)

[Hình 18. Data bước 2 26](#_Toc168678803)

[Hình 19. Data bước 2.1 26](#_Toc168678804)

[Hình 20. Data bước 2.2 26](#_Toc168678805)

[Hình 21. Data bước 2.3 27](#_Toc168678806)

[Hình 22. Data bước 3 27](#_Toc168678807)

[Hình 23. Data bước 3.1 28](#_Toc168678808)

[Hình 24. Data bước 3.2 28](#_Toc168678809)

[Hình 25. Data bước 3.3 28](#_Toc168678810)

[Hình 26. Data bước 4 29](#_Toc168678811)

[Hình 27. Data bước 4.1 29](#_Toc168678812)

[Hình 28. Data bước 5 29](#_Toc168678813)

[Hình 29. Data bước 6 30](#_Toc168678814)

[Hình 30. Phân loại bằng LSTM bước 1 31](#_Toc168678815)

[Hình 31. Phân loại bằng LSTM bước 2 31](#_Toc168678816)

[Hình 32. Phân loại bằng LSTM bước 3 32](#_Toc168678817)

[Hình 33. Phân loại bằng LSTM bước 3.1 33](#_Toc168678818)

[Hình 34. Phân loại bằng LSTM bước 3.2 34](#_Toc168678819)

[Hình 35. Phân loại bằng LSTM bước 3.3 35](#_Toc168678820)

[Hình 36. Phân loại bằng LSTM bước 3.4 36](#_Toc168678821)

[Hình 37. Phân loại bằng LSTM bước 3.5 37](#_Toc168678822)

[Hình 38. Phân loại bằng LSTM bước 3.6 38](#_Toc168678823)

[Hình 39. Phân loại bằng LSTM bước 4 39](#_Toc168678824)

[Hình 40. Phân loại bằng LSTM bước 4.1 40](#_Toc168678825)

[Hình 41. Phân loại bằng LSTM bước 4.2 40](#_Toc168678826)

[Hình 42. Phân loại bằng LSTM bước 5 41](#_Toc168678827)

[Hình 43. Phân loại bằng LSTM bước 6 42](#_Toc168678828)

[Hình 44. Phân loại bằng LSTM bước 6.1 43](#_Toc168678829)

[Hình 45. Phân loại bằng LSTM bước 7 44](#_Toc168678830)

[Hình 46. Phân loại bằng LSTM bước 7.1 45](#_Toc168678831)

[Hình 47. Phân loại bằng LSTM bước 8 46](#_Toc168678832)

[Hình 48. Phân loại bằng LSTM bước 8.1 47](#_Toc168678833)

[Hình 49. Phân loại bằng LSTM bước 8.2 47](#_Toc168678834)

[Hình 50. Phân loại bằng LSTM bước 8.3 48](#_Toc168678835)

[Hình 51. Phân loại bằng LSTM bước 8.4 49](#_Toc168678836)

[Hình 52. Độ chính xác 50](#_Toc168678837)

[Hình 53. Độ chính xác 1 50](#_Toc168678838)

[Hình 54. Confusion Matrix 51](#_Toc168678839)

# BẢNG PHÂN CÔNG

|  |  |  |  |
| --- | --- | --- | --- |
| NHIỆM VỤ | THỊNH | VÂN | HUYỀN |
| Thu thập và tiền xử lý dữ liệu |  |  | x |
| Nghiên cứu mô hình LSTM |  | x |  |
| Thực hiện thử nghiệm và đánh giá mô hình LSTM | x |  |  |
| Tổng hợp báo cáo | x | x | x |
| Tổng hợp slide thuyết trình | x | x | x |

# GIỚI THIỆU

## Lý do chọn đề tài

Trong thập kỷ qua, mạng xã hội đã trở thành một phần không thể thiếu của cuộc sống hàng ngày, với hàng tỷ người dùng trên khắp thế giới sử dụng các nền tảng như Facebook, Twitter, và Instagram để cập nhật thông tin và giao tiếp. Tuy nhiên, sự phát triển nhanh chóng của mạng xã hội cũng kéo theo một vấn đề nghiêm trọng: sự lan truyền của tin tức giả (fake news). Tin tức giả không chỉ làm sai lệch thông tin mà còn gây ra những hậu quả tiêu cực về mặt xã hội, chính trị và kinh tế. Chính vì vậy, việc phát hiện và ngăn chặn tin tức giả trên mạng xã hội đã trở thành một đề tài nghiên cứu cấp thiết và quan trọng.

Mạng xã hội cung cấp một lượng dữ liệu khổng lồ, bao gồm các bài viết, bình luận, hình ảnh và video. Đây là nguồn dữ liệu vô giá cho việc nghiên cứu và phát triển các mô hình phát hiện tin giả. Bằng cách phân tích các mẫu dữ liệu này, chúng ta có thể nhận diện các đặc điểm của tin giả và phát triển các công cụ tự động để phát hiện chúng.

Tính tương tác của mạng xã hội cho phép chúng ta theo dõi và phân tích cách mà tin tức lan truyền giữa các người dùng. Điều này giúp chúng ta hiểu rõ hơn về các cơ chế lan truyền tin giả và từ đó xây dựng các biện pháp ngăn chặn hiệu quả.

Chọn đề tài phát hiện tin tức giả trên mạng xã hội là một lựa chọn hợp lý và cần thiết trong bối cảnh hiện nay. Đề tài không chỉ mang tính cấp thiết cao mà còn có tiềm năng đóng góp lớn vào khoa học và thực tiễn. Việc phát hiện tin giả không chỉ giúp cải thiện chất lượng thông tin trên mạng xã hội mà còn bảo vệ an ninh xã hội và hỗ trợ các cơ quan chức năng.

## Tính cấp thiết của đề tài

Một trong những đặc điểm nổi bật của mạng xã hội là khả năng lan truyền thông tin một cách nhanh chóng và rộng rãi. Tin tức giả có thể dễ dàng tiếp cận hàng triệu người trong một thời gian ngắn, gây ra sự hoang mang và hiểu lầm. Ví dụ, trong các cuộc bầu cử, tin giả có thể ảnh hưởng đến quyết định của cử tri và làm sai lệch kết quả bầu cử. Tương tự, trong các tình huống khẩn cấp như dịch bệnh, tin giả có thể làm tăng sự lo lắng và gây ra các hành động nguy hiểm.

Tin giả có thể gây ra những hậu quả nghiêm trọng, từ việc làm mất lòng tin của công chúng vào các nguồn tin chính thống đến việc kích động bạo lực và xung đột. Ví dụ, những tin tức giả về các vấn đề sức khỏe có thể dẫn đến việc sử dụng các biện pháp điều trị không đúng, gây hại cho sức khỏe của người dân. Trong lĩnh vực kinh tế, tin giả có thể ảnh hưởng đến thị trường chứng khoán và gây ra những tổn thất tài chính lớn.

Bằng cách phát hiện và loại bỏ tin giả, chúng ta có thể cải thiện chất lượng thông tin trên mạng xã hội. Điều này không chỉ giúp người dùng có được thông tin chính xác mà còn giúp xây dựng lòng tin của công chúng vào các nền tảng mạng xã hội.

Phát hiện và ngăn chặn tin giả giúp bảo vệ an ninh xã hội bằng cách ngăn chặn các thông tin sai lệch có thể gây ra bạo lực hoặc xung đột. Ngoài ra, nó còn giúp ngăn chặn các hành vi lừa đảo và gian lận trên mạng xã hội.

Các công cụ phát hiện tin giả có thể hỗ trợ các cơ quan chức năng trong việc kiểm soát và xử lý các thông tin sai lệch, từ đó giúp duy trì trật tự xã hội và bảo vệ lợi ích công cộng.

# LÝ DO CHỌN MÔ HÌNH LSTM

Các nghiên cứu và thí nghiệm trước đó đã chứng minh rằng LSTM có khả năng dự đoán và phân loại tin tức giả mạo với độ chính xác cao hơn so với các mô hình khác. Điều này được minh chứng qua nhiều bài báo khoa học và các nghiên cứu thực nghiệm.

LSTM vượt trội trong việc xử lý các mối quan hệ phức tạp trong dữ liệu chuỗi, điều này rất quan trọng trong việc phân tích các bài viết trên mạng xã hội.

Với khả năng dự đoán xu hướng và phân tích mẫu phân phối, LSTM không chỉ giúp phát hiện tin tức giả mạo mà còn cung cấp các thông tin hữu ích cho các nhà quản lý mạng xã hội và các cơ quan chức năng.

Việc chọn mô hình LSTM để phát hiện tin tức giả trên mạng xã hội là một quyết định dựa trên sự cân nhắc kỹ lưỡng về hiệu quả và tính khả thi của mô hình. LSTM không chỉ giúp giải quyết các hạn chế của RNN truyền thống mà còn mang lại độ chính xác cao trong việc phân tích và dự đoán tin tức giả. Qua việc sử dụng LSTM, chúng tôi hy vọng có thể đóng góp vào việc cải thiện tính đáng tin cậy và an toàn của các nền tảng mạng xã hội, đồng thời cung cấp các công cụ hữu ích cho việc kiểm soát và ngăn chặn tin tức giả mạo.

A screenshot of a graph

Description automatically generated

Hình 1. Độ chính xác của các thuật dùng trong phát hiện tin tức giả mạo

# MÔ HÌNH TOÁN HỌC LSTM PHÂN LOẠI ĐỂ PHÁT HIỆN TIN GIẢ

## Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN)

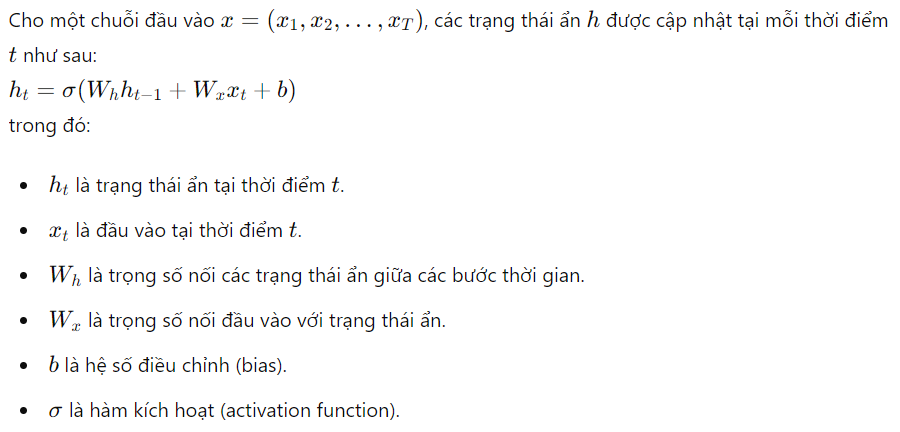
### Đặc điểm chính

* *Trạng thái ẩn (Hidden States)* RNN duy trì các trạng thái ẩn, cho phép nó lưu trữ thông tin từ các bước thời gian trước đó trong dãy dữ liệu.
* *Khả năng tuần hoàn (Recurrent)* Mỗi nơ-ron trong RNN không chỉ nhận đầu vào từ lớp trước đó mà còn nhận đầu vào từ chính nó từ bước thời gian trước đó.
* *Ứng dụng đa dạng* RNN thường được sử dụng trong các ứng dụng xử lý ngôn ngữ tự nhiên (NLP), nhận dạng giọng nói, dịch máy, phân tích chuỗi thời gian và nhiều ứng dụng khác liên quan đến dữ liệu tuần tự.

### Kiến trúc cơ bản

Một mạng RNN cơ bản có các lớp chính sau:

* Lớp đầu vào (Input Layer): Nhận dữ liệu đầu vào tuần tự.
* Lớp ẩn (Hidden Layer): Xử lý và lưu trữ thông tin từ các bước thời gian trước đó.
* Lớp đầu ra (Output Layer): Tạo ra kết quả đầu ra.



Hình 2. Công thức cơ bản của RNN

### Các biến thể của RNN

* LSTM (Long Short-Term Memory): Là một loại mạng RNN đặc biệt được thiết kế để giải quyết vấn đề biến mất gradient và ghi nhớ dài hạn tốt hơn.
* GRU (Gated Recurrent Unit): Một biến thể khác của RNN, đơn giản hơn LSTM nhưng vẫn giúp cải thiện việc ghi nhớ dài hạn.

### Thách thức của RNN

* Vấn đề biến mất và bùng nổ gradient (Vanishing and Exploding Gradients): Khi chuỗi dữ liệu quá dài, gradient trong quá trình huấn luyện có thể trở nên rất nhỏ hoặc rất lớn, dẫn đến khó khăn trong việc điều chỉnh trọng số.
* Khả năng nhớ dài hạn (Long-term Dependencies): Mặc dù RNN có khả năng ghi nhớ thông tin, nhưng khi khoảng cách giữa các thông tin quan trọng quá lớn, khả năng ghi nhớ của RNN trở nên hạn chế.

## Lý thuyết mô hình LSTM

### Giới thiệu

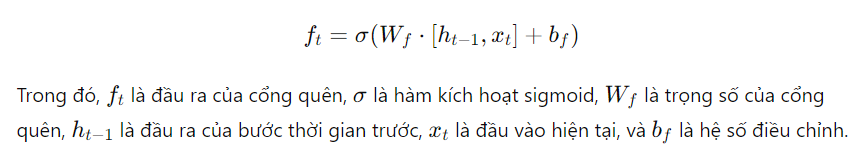
Mô hình LSTM (Long Short-Term Memory) là một loại mạng nơ-ron hồi quy (RNN) đặc biệt, được thiết kế để khắc phục các hạn chế của RNN truyền thống trong việc xử lý dữ liệu chuỗi dài. LSTM được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997 và đã trở thành một công cụ quan trọng trong lĩnh vực học sâu, đặc biệt là trong các bài toán liên quan đến xử lý ngôn ngữ tự nhiên (NLP).

RNN truyền thống gặp phải hai vấn đề chính khi xử lý các chuỗi dữ liệu dài là Biến Mất Gradient (Vanishing Gradient) và Biến Mất Gradient (Vanishing Gradient). Trong quá trình huấn luyện, gradient của hàm lỗi có thể trở nên rất nhỏ khi lan truyền ngược qua nhiều lớp, khiến cho RNN không thể học hiệu quả các mối quan hệ dài hạn. Ngoài ra, Gradient có thể trở nên quá lớn, dẫn đến sự không ổn định trong việc cập nhật các trọng số. Những vấn đề này làm cho RNN khó khăn trong việc ghi nhớ và phân tích các ngữ cảnh quan trọng trong các chuỗi dữ liệu dài.

### Cấu trúc mô hình LSTM

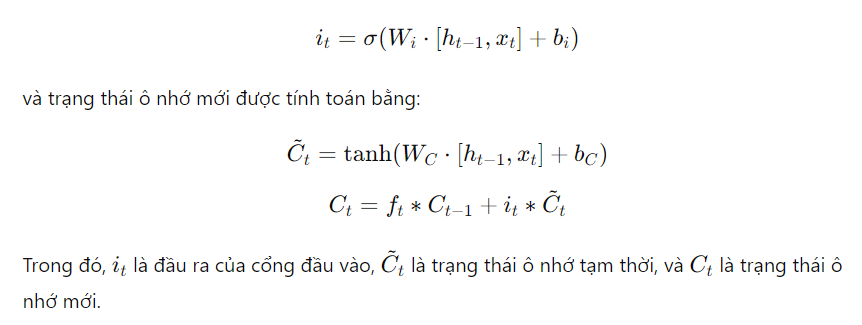
LSTM được thiết kế để khắc phục các vấn đề trên bằng cách giới thiệu một cấu trúc đặc biệt gồm các ô nhớ (memory cells) và các cổng (gates) để kiểm soát luồng thông tin. Một đơn vị LSTM cơ bản bao gồm ba loại cổng chính:

* Cổng Quên (Forget Gate): Quyết định thông tin nào từ trạng thái trước đó cần được giữ lại và thông tin nào cần bị loại bỏ. Công thức tính toán cổng quên như sau:



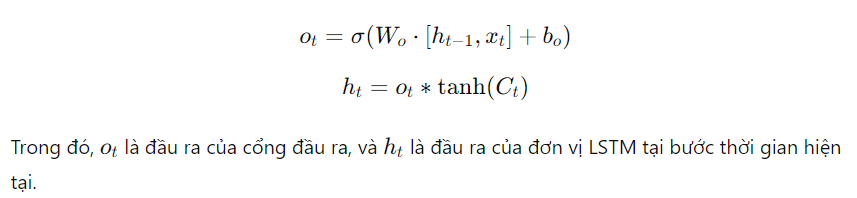
Hình 3. Cổng quên

* Cổng Đầu Vào (Input Gate): Quyết định thông tin mới nào sẽ được thêm vào ô nhớ. Công thức tính toán cổng đầu vào như sau:

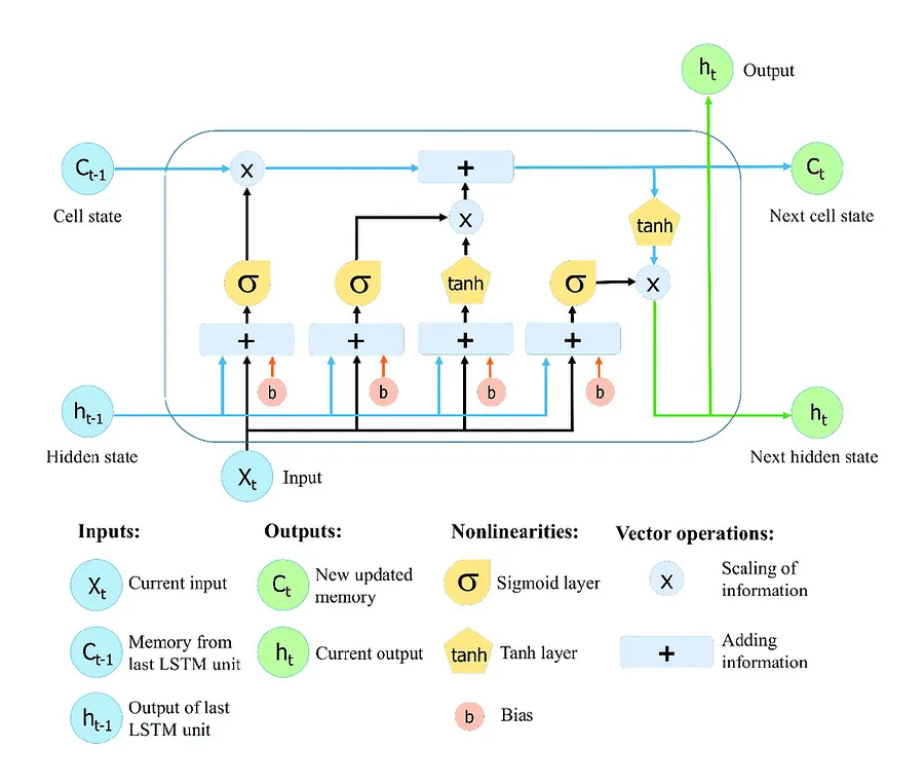


Hình 4. Cổng đầu vào

* Cổng Đầu Ra (Output Gate): Quyết định phần nào của trạng thái ô nhớ sẽ được sử dụng để tạo ra đầu ra của bước thời gian hiện tại. Công thức tính toán cổng đầu ra như sau:



Hình 5. Cổng đầu ra



Hình 6. Model của LSTM

* *Sơ đồ cấu trúc mạng nơ-ron LSTM*

+ Hàm sigmoid: (3)

+ tanh activation function:

+ Forget gate: (4)

+ Input gate: (5)

(Trong đó các phương pháp tính toán của *it* và *Ĉt* được hiển thị trong Eqs. 4 and 5)

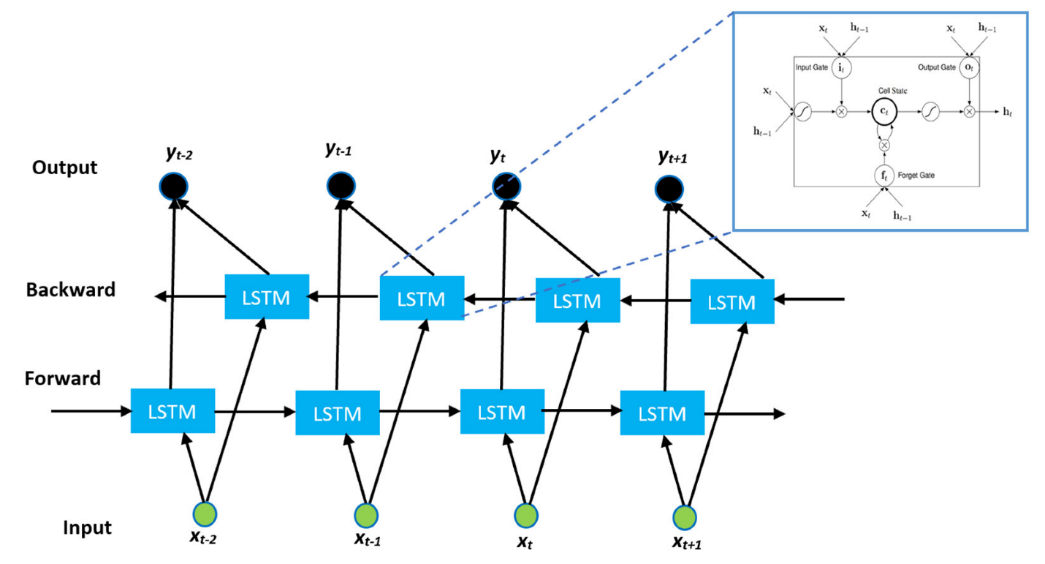
+ Output gate: (6)

(7)

(8)

(9)

* Bi-LSTM: LSTM hai chiều bao gồm hai lớp LSTM: LSTM chuyển tiếp và LSTM ngược. LSTM chuyển tiếp xử lý chuỗi đầu vào từ đầu đến cuối, trong khi LSTM ngược xử lý chuỗi đầu vào từ cuối đến đầu. Các đầu ra từ hai LSTM sau đó được nối với nhau để có được đầu ra cuối cùng. Bằng cách xử lý chuỗi đầu vào theo cả hai hướng, LSTM hai chiều có thể nắm bắt thông tin từ cả bối cảnh trong quá khứ và tương lai. Điều này có thể hữu ích trong các nhiệm vụ mà ngữ cảnh đóng vai trò quan trọng để hiểu được trình tự đầu vào, chẳng hạn như xử lý ngôn ngữ tự nhiên.



Hình 7. Bi-LSTM visualization

### Ưu Điểm của LSTM

Nhờ cơ chế cổng, LSTM có khả năng ghi nhớ thông tin quan trọng trong một khoảng thời gian dài, giúp giải quyết vấn đề biến mất và lan truyền gradient.

LSTM có thể học và ghi nhớ các mối quan hệ phức tạp trong dữ liệu chuỗi, đặc biệt là trong các bài toán xử lý ngôn ngữ tự nhiên.

LSTM đã được áp dụng thành công trong nhiều lĩnh vực khác nhau như dịch máy, nhận dạng giọng nói, phân loại văn bản, và dự đoán chuỗi thời gian.

LSTM là một công cụ mạnh mẽ và hiệu quả trong việc xử lý các chuỗi dữ liệu dài và phức tạp. Với cấu trúc đặc biệt gồm các ô nhớ và các cổng kiểm soát, LSTM có khả năng ghi nhớ và phân tích các ngữ cảnh quan trọng, giúp giải quyết các vấn đề mà RNN truyền thống gặp phải. Điều này làm cho LSTM trở thành một lựa chọn lý tưởng cho các bài toán liên quan đến dữ liệu chuỗi, đặc biệt là trong lĩnh vực phát hiện tin tức giả trên mạng xã hội.

# TỔNG QUAN VỀ DỮ LIỆU ĐÀO TẠO

## Dataset

Bộ dữ liệu này chứa các mẫu tin tức giả và thật được thu thập từ các nguồn đáng tin cậy là PolitiFact và GossipCop từ năm 2018. Mục đích của bộ dữ liệu là cung cấp cơ sở để nghiên cứu và phát triển các mô hình phát hiện tin giả. Do chính sách bảo mật của Twitter và bản quyền của các nhà xuất bản tin tức, bộ dữ liệu đầy đủ không thể được phân phối công khai, và các thông tin về tương tác xã hội cùng thông tin người dùng không được tiết lộ. Tuy nhiên, mã nguồn trong kho này có thể được sử dụng để tải xuống các bài viết tin tức từ các trang web đã xuất bản và dữ liệu xã hội liên quan từ Twitter.

Phiên bản tối giản của bộ dữ liệu mới nhất được cung cấp trong thư mục "dataset" bao gồm các tệp sau:

* politifact\_fake.csv: Các mẫu tin tức giả thu thập từ PolitiFact.
* politifact\_real.csv: Các mẫu tin tức thật thu thập từ PolitiFact.
* gossipcop\_fake.csv: Các mẫu tin tức giả thu thập từ GossipCop.
* gossipcop\_real.csv: Các mẫu tin tức thật thu thập từ GossipCop.

Mỗi tệp CSV trên là tệp phân cách bằng dấu phẩy và có các cột sau:

* Id: Định danh duy nhất cho mỗi tin tức.
* url: URL của bài viết trên web đã xuất bản tin tức đó.
* Title: Tiêu đề của bài viết tin tức.
* tweet\_ids: ID của các tweet chia sẻ tin tức đó. Trường này là một danh sách các ID tweet được phân cách bằng tab.

Bộ dữ liệu này là một công cụ hữu ích cho việc nghiên cứu tin giả, cung cấp các mẫu tin tức từ các nguồn đáng tin cậy cùng với các thông tin liên quan đến việc chia sẻ trên Twitter, giúp các nhà nghiên cứu có cơ sở để phân tích và xây dựng các mô hình phát hiện tin giả hiệu quả.

Link dataset: <https://github.com/KaiDMML/FakeNewsNet>

### Đánh giá theo ngày tạo tài khoản

A graph with numbers and text

Description automatically generated with medium confidence

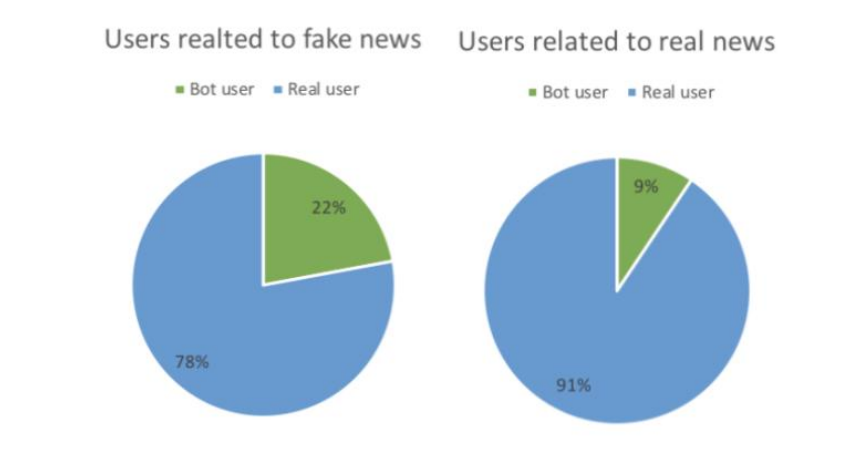
Hình 8. Đánh giá theo ngày tạo tài khoản của PolitiFact

A graph with numbers and lines

Description automatically generated with medium confidence

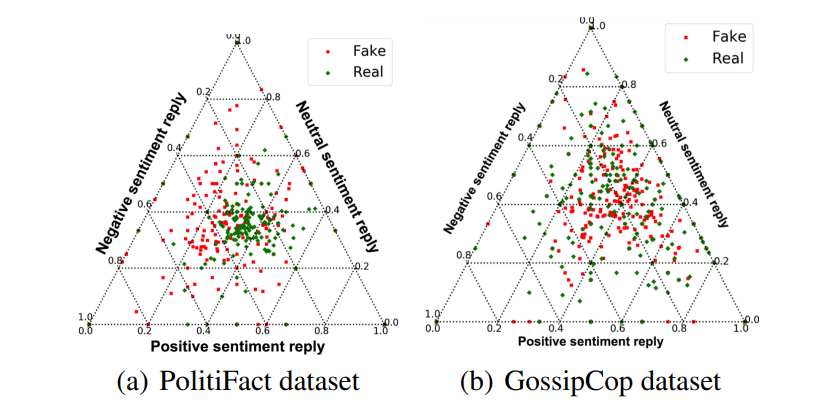
Hình 9. Đánh giá theo ngày tạo tài khoản của GossipCop

### Đánh giá theo tỉ lệ người dùng thật hay bot



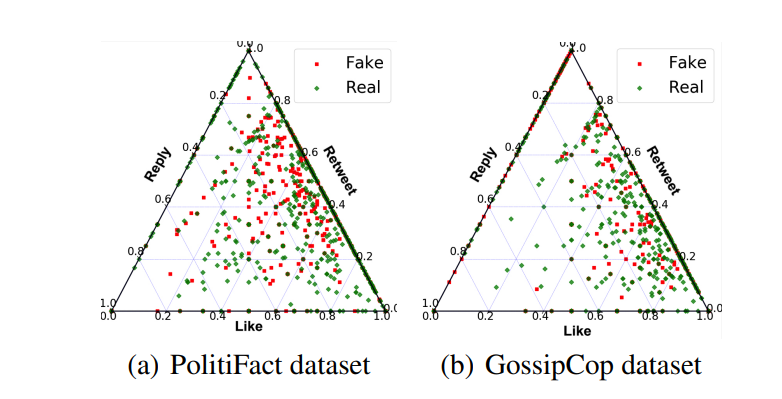
Hình 10. Đánh giá theo tỉ lệ người dùng thật hay bot

### Đánh giá theo tỉ lệ cảm xúc của các phản hồi



Hình 11. Đánh giá theo tỉ lệ cảm xúc của các phản hồi

### Đánh giá theo tỉ lệ lượt thích, retweet, phản hồi của bài đăng



Hình 12. Đánh giá theo tỉ lệ lượt thích, retweet, phản hồi của bài đăng

### Đánh giá theo sự phân bổ số lượng người theo dõi tài khoản đăng tin

A group of graphs showing a number of users

Description automatically generated

Hình 13. Đánh giá theo sự phân bổ số lượng người theo dõi tài khoản đăng tin

### Đánh giá theo tỉ lệ tiếp cận tin tức thật so với giả

A screenshot of a graph

Description automatically generated

Hình 14. Đánh giá theo tỉ lệ tiếp cận tin tức thật so với giả

### Đánh giá theo phân bố không gian của người đăng tin giả và tin thật

**A map of the world with red circles

Description automatically generated**

Hình 15. Đánh giá theo phân bố không gian của người đăng tin giả và tin thật

# CHI TIẾT THỰC HIỆN

## Data

Xử lý data:

* Gồm 4 file csv

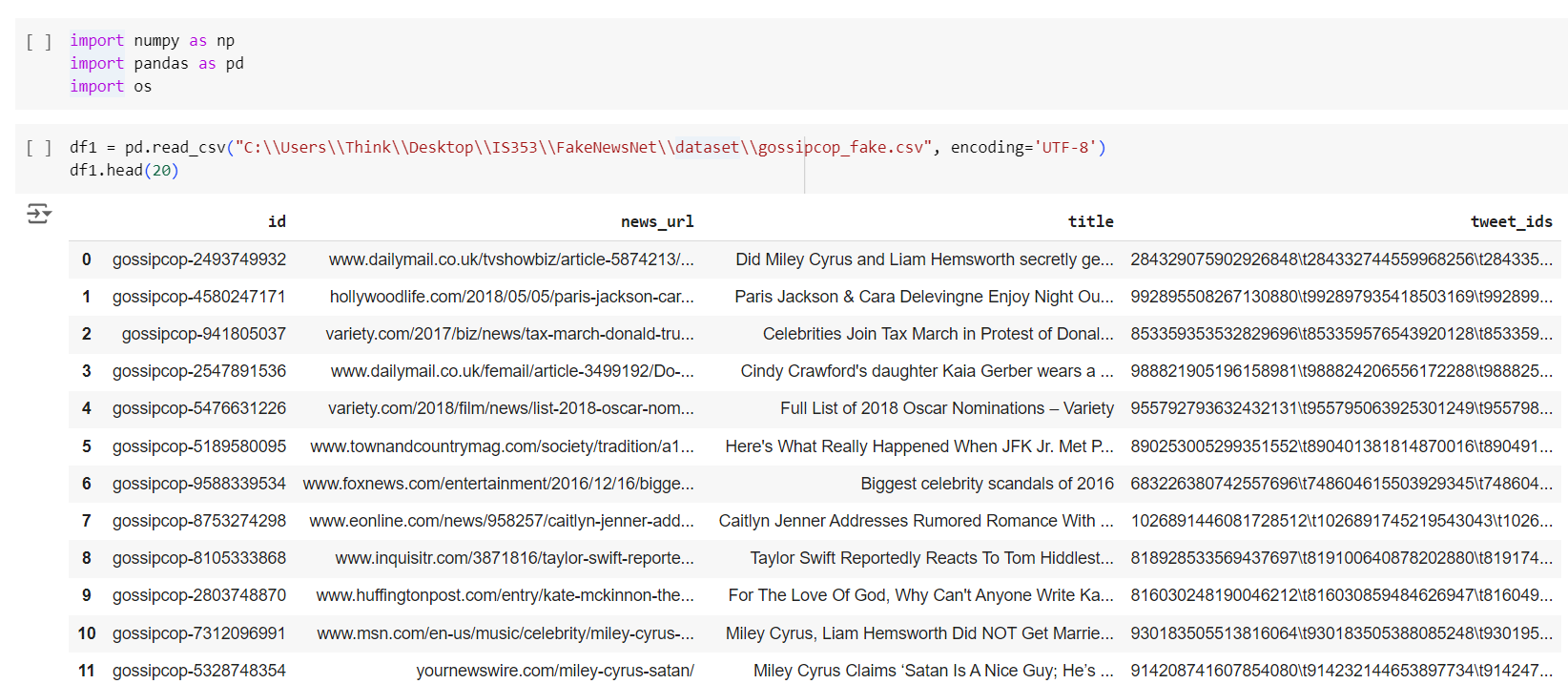
A group of files with text

Description automatically generated

Hình 16. Data

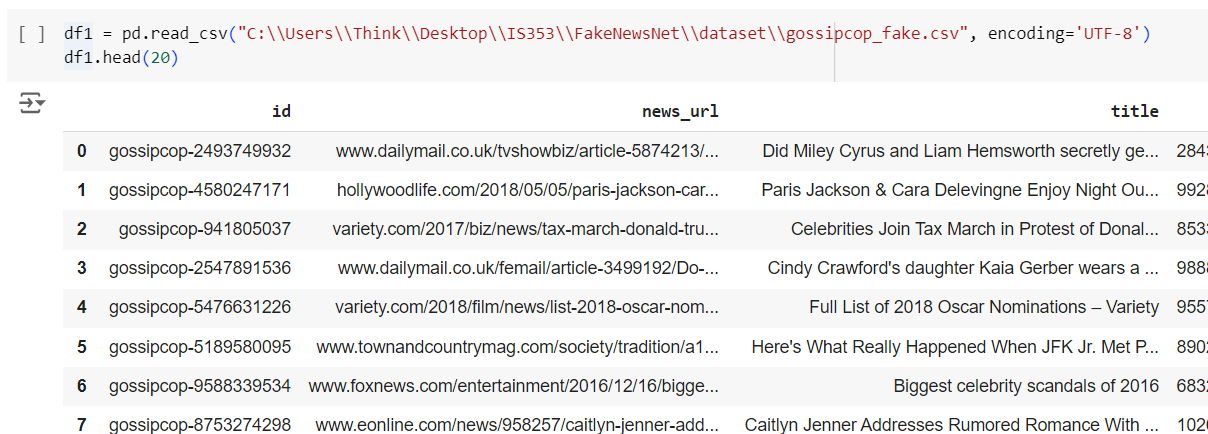
* Ý tưởng: Tạo 1 file Dataset\_join.ipynb để gộp 4 file data trên thành 1 file data và lưu DataFrame vào tệp CSV trong Google Drive.

**Bước 1:** Import thư viện ‘numpy’ để làm việc với mảng và ‘pandas’ để load, lưu trữ, xử lý, và phân tích dữ liệu, ‘os’ để tương tác với hệ điều hành.



Hình 17. Data bước 1

**Bước 2:** Sử dụng thư viện ‘pandas’ để đọc dữ liệu từ 4 tệp CSV và hiển thị 20 hàng đầu tiên của các DataFrame đó. Và mã hóa văn bản trong tệp CSV được chỉ định là 'UTF-8'.



Hình 18. Data bước 2

A screenshot of a computer

Description automatically generated

Hình 19. Data bước 2.1

A screenshot of a computer

Description automatically generated

Hình 20. Data bước 2.2

A screenshot of a computer

Description automatically generated

Hình 21. Data bước 2.3

**Bước 3:** Sử dụng phương thức info() của DataFrame trong thư viện Pandas được sử dụng để cung cấp thông tin tổng quan về 4 DataFrame, bao gồm: Số lượng hàng và cột, tên của mỗi cột, tổng số giá trị không thiếu (non-null) trong mỗi cột, kiểu dữ liệu của mỗi cột.

A screenshot of a computer code

Description automatically generated

Hình 22. Data bước 3

A screenshot of a computer code

Description automatically generated

Hình 23. Data bước 3.1

A screenshot of a computer code

Description automatically generated

Hình 24. Data bước 3.2

A screenshot of a computer code

Description automatically generated

Hình 25. Data bước 3.3

**Bước 4:** Thêm một cột mới có tên là "label" vào mỗi DataFrame, và gán giá trị 0 cho các hàng của df1 và df3, cũng như gán giá trị 1 cho các hàng của df2 và df4. Với 0 là thông tin giả và 1 là thông tin thật.

A screenshot of a computer code

Description automatically generated

Hình 26. Data bước 4

A screenshot of a computer

Description automatically generated

Hình 27. Data bước 4.1

**Bước 5:** Sử dụng hàm **concat()** của Pandas để nối các DataFrame df1, df2, df3, và df4 thành một DataFrame lớn hơn. Tham số **ignore\_index=True** được sử dụng để đặt lại chỉ số của hàng của DataFrame mới sau khi nối, đảm bảo rằng chỉ số hàng không bị trùng lặp giữa các DataFrame.

A black and white text

Description automatically generated

Hình 28. Data bước 5

**Bước 6:** Lưu DataFrame df thành một tệp CSV mới có tên là "Fakenewsnet.csv" mà không bao gồm cột chỉ số hàng (index).

A close-up of a logo

Description automatically generated

Hình 29. Data bước 6

## Phân loại bằng LSTM

* Ý tưởng:

**Đầu vào:** **Tập dữ liệu** dùng tiêu đề của các bài báo để dùng cho mô hình LSTM học.

Tiếp theo là tiền xử lý dữ liệu bao gồm loại bỏ các ký tự không phải chữ cái, chuyển đổi chữ hoa thành chữ thường, tách các từ và loại bỏ từ dừng, và cuối cùng là stem từng từ. Các từ sau khi tiền xử lý được biểu diễn bằng mã hóa one-hot và sau đó thực hiện padding để có độ dài đồng nhất. Và mô hình LSTM được xây dựng bằng cách sử dụng các lớp nhúng, LSTM và kết nối đầy đủ. Mô hình này được huấn luyện chỉ ra xác suất nằm ở khoảng ranh giới tin thật và giả. Ví dụ: Xác suất = (0.5;1] là tin thật và [0;0.5] là tin giả với độ chính xác hơn 80%.

**Đầu ra:** Sau khi các đoạn văn bản dùng để **kiểm thử** đã mã hóa thành mảng các chuỗi số nguyên có chiều dài đồng nhất chạy qua mô hình mạng nơ-ron LSTM đã được huấn luyện, nếu xác suất của đoạn văn bản đó nằm ở khoảng ranh giới nào thì sẽ hiển thị kết quả xem tin đó có phải giả hay không. Ví dụ: Dùng xác suất ở ví dụ đầu ra thì nếu đoạn văn bản kiểm thử nằm vào khoảng [0;0.5] thì là tin giả, sẽ hiển thị kết quả là [0] tượng trưng cho tin giả.

**Bước 1:** Import thư viện ‘pandas’ để load, lưu trữ, xử lý, phân tích dữ liệu và sử dụng "from google.colab import drive" và "drive.mount('/content/drive')" trong môi trường Google Colab để kết nối với Google Drive

**A screenshot of a computer

Description automatically generated**

Hình 30. Phân loại bằng LSTM bước 1

**Bước 2:** Đọc tập dữ liệu CSV từ Google Drive vào một DataFrame Pandas, sau đó hiển thị các hàng đầu tiên của dữ liệu với phương thức head().

**A screenshot of a computer

Description automatically generated**

Hình 31. Phân loại bằng LSTM bước 2

**Bước 3:** Tiền xử lý dữ liệu :

* Dùng phương thức dropna() của Pandas để loại bỏ các hàng chứa giá trị NaN (null) khỏi DataFrame news.
* Sử dụng phương thức drop() của Pandas với tham số axis=1 để tạo một DataFrame mới X bằng cách loại bỏ cột 'label' từ DataFrame news.
* Tạo một Series mới y bằng cách chọn cột 'label' từ DataFrame news.
* X.shape được sử dụng để lấy kích thước của DataFrame hoặc mảng Numpy X
* y.shape sẽ trả về số lượng phần tử trong Series y.
* Import thư viện ‘TensorFlow’ và xem phiên bản của ‘TensorFlow’

A screenshot of a computer

Description automatically generated

Hình 32. Phân loại bằng LSTM bước 3

* Import thư viện **Embedding** sử dụng trong các tác vụ xử lý ngôn ngữ tự nhiên để chuyển đổi từ (được biểu diễn dưới dạng số nguyên) thành các vector dày đặc có kích thước cố định. Thư viện **pad\_sequences** sử dụng để đảm bảo rằng tất cả các chuỗi trong một lô có cùng độ dài bằng cách thêm các số không hoặc cắt bớt các chuỗi ngắn hơn. **Sequential** là một cấu trúc tuyến tính của các lớp trong Keras. Hàm **one\_hot** sử dụng để chuyển đổi văn bản thành biểu diễn mã hóa one-hot. **LSTM** là một loại lớp mạng nơ-ron hồi tiếp. **Dense** là một lớp kết nối đầy đủ trong Keras, trong đó mỗi neuron được kết nối với mọi neuron trong các lớp trước và sau đó.
* Thiết lập kích thước của từ vựng là 5000 bằng cách gán giá trị này cho biến voc\_size.

A screenshot of a computer program

Description automatically generated

Hình 33. Phân loại bằng LSTM bước 3.1

* **messages=X.copy()** được sử dụng để tạo một bản sao của DataFrame X và gán nó cho biến messages.
* **messages['title'][1]** truy cập phần tử ở dòng thứ hai và cột có nhãn 'title' trong DataFrame messages
* **messages.reset\_index(inplace=True)** được sử dụng để đặt lại chỉ số của DataFrame **messages**. Khi **inplace=True**, nó sẽ làm thay đổi trực tiếp trên DataFrame messages mà không cần gán kết quả trở lại một biến khác.
* Import thư viện **nltk** cung cấp các công cụ để tokenize, stemming, lemmatize văn bản. Thư viện **re** sử dụng để thực hiện các thao tác biểu thức chính quy, giúp tìm kiếm và thay thế các mẫu chuỗi trong văn bản. **stopwords** dùng để loại bỏ các từ phổ biến như "is", "the", "and", không mang ý nghĩa quan trọng trong việc phân tích văn bản.
* **nltk.download('stopwords')** được sử dụng để tải xuống tập tin dữ liệu "stopwords" từ thư viện NLTK.

A screenshot of a computer

Description automatically generated

Hình 34. Phân loại bằng LSTM bước 3.2

* **from nltk.stem.porter import** **PorterStemmer** được sử dụng để chuyển các từ trong văn bản về dạng gốc của chúng.
* **ps = PorterStemmer()** dùng để tạo một đối tượng PorterStemmer.
* **corpus = []** dùng để khởi tạo một danh sách để lưu trữ văn bản đã tiền xử lý.
* Vòng lặp for dùng để duyệt qua mỗi hàng trong DataFrame messages
* Đối với mỗi vòng lặp của từng tiêu đề:
  + **review = re.sub('[^a-zA-Z]', ' ', messages['title'][i])** dùng để loại bỏ các ký tự không phải là chữ cái từ tiêu đề văn bản và thay thế chúng bằng khoảng trắng.
  + **review = review.lower()** dùng để chuyển đổi tất cả các chữ cái trong văn bản thành chữ thường.
  + **review = review.split()** dùng để tách văn bản thành một danh sách các từ.
  + **review = [ps.stem(word) for word in review if not word in stopwords.words('english')]** dùng để stem từng từ trong văn bản bằng cách sử dụng PorterStemmer, sau đó loại bỏ các từ dừng (stopwords).
  + **review = ' '.join(review)** dùng để ghép các từ đã được stem lại thành một chuỗi văn bản.
  + **corpus.append(review)** dùng để thêm văn bản đã được xử lý vào danh sách corpus.

A screenshot of a computer program

Description automatically generated

Hình 35. Phân loại bằng LSTM bước 3.3

* Biến **corpus** là danh sách các văn bản đã được xử lý và tiền xử lý từ tập dataset ban đầu. Mỗi phần tử trong danh sách corpus là một chuỗi văn bản đã qua tiền xử lý, đã được chuyển thành chữ thường, loại bỏ ký tự không phải chữ cái, loại bỏ từ dừng, và stem từng từ.

A screenshot of a computer screen

Description automatically generated

Hình 36. Phân loại bằng LSTM bước 3.4

* **one\_hot(words, voc\_size) for words in corpus** dùng vòng lặp để duyệt qua mỗi chuỗi văn bản trong corpus. Dùng hàm **one\_hot** từ thư viện Keras để mã hóa mỗi từ trong chuỗi thành một số nguyên, với số lượng từ vựng tối đa là voc\_size.
* Biến **onehot\_repr** dùng để lưu trữ kết quả của quá trình mã hóa one-hot cho tất cả các văn bản trong corpus. Kết quả cuối cùng là một danh sách các danh sách số nguyên, mỗi danh sách số nguyên biểu diễn một văn bản đã được mã hóa one-hot.

A screenshot of a computer

Description automatically generated

Hình 37. Phân loại bằng LSTM bước 3.5

* **embedded\_docs=pad\_sequences(onehot\_repr,padding='pre',maxlen=sent\_length)** sử dụng để chuyển đổi danh sách các số nguyên biểu diễn các văn bản đã được mã hóa one-hot thành các chuỗi số nguyên có chiều dài đồng nhất bằng cách sử dụng padding.
* Biến **embedded\_docs** dùng để lưu trữ kết quả cuối cùng của quá trình chuyển đổi.
* **embedded\_docs[0]** là một ví dụ về chuỗi số nguyên biểu diễn văn bản đầu tiên trong dataset sau khi đã được chuyển đổi và thêm padding.

A screenshot of a computer

Description automatically generated

Hình 38. Phân loại bằng LSTM bước 3.6

**Bước 4:** Tạo mô hình:

* **embedding\_vector\_features=40**: Đây là số lượng các tính năng (đặc trưng) trong không gian nhúng. Trong mô hình này, mỗi từ sẽ được biểu diễn dưới dạng một vectơ có 40 chiều trong không gian nhúng.
* **model=Sequential()**: Tạo một mô hình tuần tự, trong đó các lớp được xếp chồng lên nhau theo thứ tự tuần tự.
* **model.add(Embedding(voc\_size, embedding\_vector\_features))**: Thêm một lớp nhúng vào mô hình. Lớp nhúng sẽ chuyển đổi từng từ đã được mã hóa one-hot thành một vectơ nhúng có số chiều là **embedding\_vector\_features**.
* **model.add(LSTM(100))**: Thêm một lớp LSTM vào mô hình với 100 đơn vị LSTM. LSTM là một kiến trúc mạng nơ-ron hồi tiếp được sử dụng để xử lý dữ liệu tuần tự như văn bản hoặc chuỗi.
* **model.add(Dense(1, activation='sigmoid'))**: Thêm một lớp kết nối đầy đủ (fully connected) với một neuron đầu ra và kích hoạt sigmoid. Điều này sẽ cho ra một dự đoán nhị phân (binary prediction).
* **model.compile(loss='binary\_crossentropy',optimizer='adam', metrics=['accuracy'])**: Biên dịch mô hình với hàm mất mát là binary cross-entropy, thuật toán tối ưu hóa là Adam, và độ đo là độ chính xác (accuracy).
* **print(model.summary())**: In ra tóm tắt của mô hình, bao gồm thông tin về kiến trúc và số lượng tham số. Điều này giúp kiểm tra lại xem mô hình đã được định nghĩa đúng hay chưa.

A screenshot of a computer program

Description automatically generated

Hình 39. Phân loại bằng LSTM bước 4

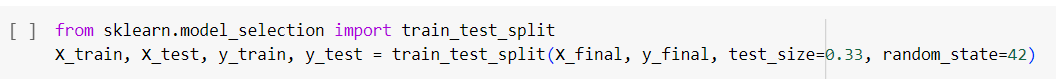
* **len(embedded\_docs)** sẽ trả về số lượng văn bản trong **embedded\_docs**, trong khi **y.shape** sẽ trả về hình dạng (shape) của dữ liệu nhãn **y**
* **np.array(embedded\_docs**) dùng để chuyển đổi **embedded\_docs** thành một mảng NumPy.
* **np.array(y)** dùng để chuyển đổi **y** thành một mảng NumPy.
* **X\_final.shape** và **y\_final.shape** sẽ trả về hình dạng (shape) của mảng **X\_final** và **y\_final** tương ứng.

A screenshot of a computer code

Description automatically generated

Hình 40. Phân loại bằng LSTM bước 4.1

* Hàm **train\_test\_split** từ thư viện **scikit-learn** dùng để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* **X\_final** và **y\_final** là dữ liệu đầu vào và nhãn đã được chuẩn bị trước đó.
* **test\_size=0.33**: Đối số này chỉ định tỷ lệ dữ liệu sẽ được chia thành tập kiểm tra, 33% dữ liệu sẽ được sử dụng cho tập kiểm tra.
* **random\_state=42** là đối số đảm bảo rằng việc chia dữ liệu sẽ được thực hiện một cách ngẫu nhiên nhưng sẽ đảm bảo rằng kết quả của quá trình chia sẽ nhất quán giữa các lần chạy.



Hình 41. Phân loại bằng LSTM bước 4.2

**Bước 5:** Huấn luyện mô hình:

* Phương thức **fit** của mô hình được sử dụng để huấn luyện mô hình trên dữ liệu với đối số:
  + **X\_train** và **y\_train**: Dữ liệu huấn luyện và nhãn tương ứng.
  + **validation\_data=(X\_test, y\_test)**: Dữ liệu kiểm tra được sử dụng để đánh giá hiệu suất của mô hình sau mỗi epoch.
  + **epochs=10**: Số lượng epochs (vòng lặp qua toàn bộ tập dữ liệu huấn luyện) mà mô hình sẽ được huấn luyện.
  + **batch\_size=64**: Kích thước của các batch dữ liệu được sử dụng trong quá trình huấn luyện. Kích thước batch này là 64, nghĩa là mỗi lần mô hình sẽ được huấn luyện trên 64 mẫu dữ liệu trước khi cập nhật trọng số.

A screenshot of a computer

Description automatically generated

Hình 42. Phân loại bằng LSTM bước 5

**Bước 6:** Đánh giá độ chính xác của tập test:

* **y\_pred = model.predict(X\_test)** được sử dụng để dự đoán nhãn trên tập kiểm tra **X\_test** bằng cách sử dụng mô hình đã huấn luyện.
* **y\_pred\_binary = (y\_pred > 0.5).astype(int)** được sử dụng để chuyển đổi các dự đoán xác suất từ mô hình thành các dự đoán nhị phân:
  + **(y\_pred > 0.5)**: Điều này sẽ tạo ra một mảng boolean với các giá trị True cho các dự đoán có xác suất lớn hơn 0.5 và False cho các dự đoán còn lại.
  + **.astype(int**): Điều này chuyển đổi các giá trị boolean sang dạng số nguyên, trong đó True sẽ được chuyển thành 1 và False sẽ được chuyển thành 0.

A screenshot of a computer program

Description automatically generated

Hình 43. Phân loại bằng LSTM bước 6

* Import **confusion\_matrix** từ thư viện **scikit-learn** để tính ma trận nhầm lẫn (confusion matrix)
* Tính toán ma trận nhầm lẫn bằng cách so sánh nhãn thực tế (**y\_test**) với dự đoán của mô hình (**y\_pred\_binary**)
* Hàm **accuracy\_score** từ thư viện **scikit-learn** được sử dụng để tính toán độ chính xác của mô hình dựa trên dự đoán và nhãn thực tế.
* Hàm **accuracy\_score** trả về tỉ lệ phần trăm của các dự đoán đúng với hơn 80%.

A screenshot of a computer code

Description automatically generated

Hình 44. Phân loại bằng LSTM bước 6.1

**Bước 7:** Tạo biểu đồ:

**A screen shot of a graph

Description automatically generated**

Hình 45. Phân loại bằng LSTM bước 7

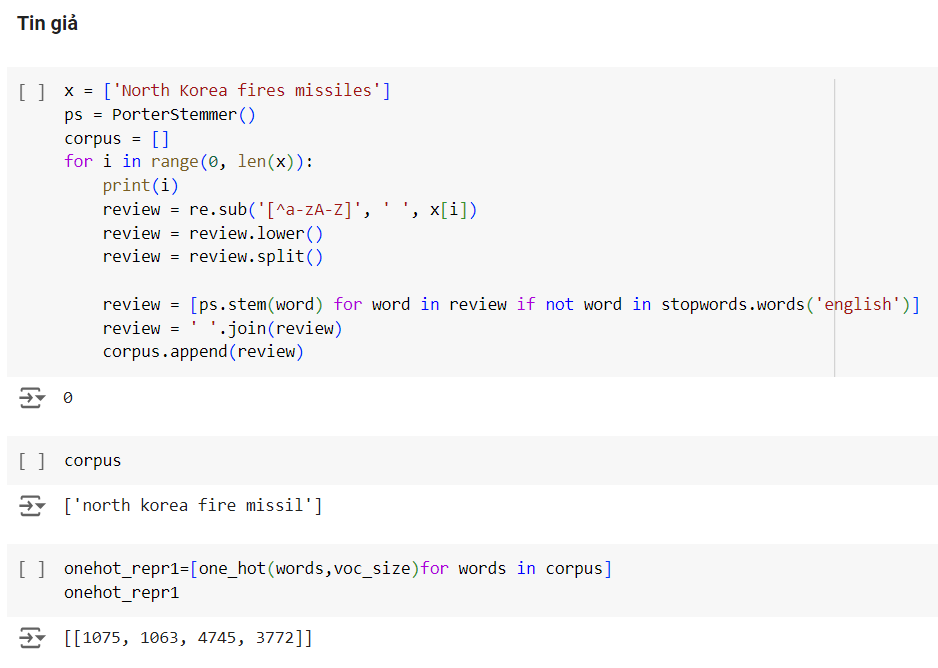
**A screen shot of a graph

Description automatically generated**

Hình 46. Phân loại bằng LSTM bước 7.1

**Bước 8:** Đưa ra ví dụ và kiểm thử:

**Tin giả:** Với chuỗi kí tự gồm thông tin không có thật và kết quả = [[0]]

****

Hình 47. Phân loại bằng LSTM bước 8

**A screenshot of a computer program

Description automatically generated**

Hình 48. Phân loại bằng LSTM bước 8.1

**Tin thật:** Với chuỗi kí tự gồm thông tin có thật của BBC News và kết quả = [[1]]



Hình 49. Phân loại bằng LSTM bước 8.2

A screenshot of a computer program

Description automatically generated

Hình 50. Phân loại bằng LSTM bước 8.3

A screenshot of a computer program

Description automatically generated

Hình 51. Phân loại bằng LSTM bước 8.4

# KẾT QUẢ NHẬN ĐƯỢC

1. **Độ chính xác**

Hệ thống mang lại độ chính xác tối đa. Chúng tôi đã đạt được một độ chính xác phân loại xấp xỉ 80% của mô hình này trên bộ thử nghiệm. Độ chính xác trọng lượng trung bình của mô hình này là từ 0,80 đến 0,83. Độ chính xác tối đa khi giá trị **epochs** lấy trong khoảngtừ 0 đen 2.

A screenshot of a computer code

Description automatically generated

Hình 52. Độ chính xác

**A screen shot of a graph

Description automatically generated**

Hình 53. Độ chính xác 1

Trong đó, hàng ma trận nhầm lẫn đại diện cho lớp thực tế (nhãn) của dữ liệu thử nghiệm và cột đại diện cho những gì phân loại dự đoán. Hàng thứ hai của Ma trận nhầm lẫn cho thấy trong số 5351 bài báo tin tức thật trong dữ liệu kiểm thử thì 4687 bài báo tin tức thật và 664 bài báo tin tức giả được dự đoán.

**A screen shot of a graph

Description automatically generated**

Hình 54. Confusion Matrix

# VII. PHẦN KẾT LUẬN

Trong những năm gần đây, nội dung gây hiểu lầm ngày càng gia tăng và tác động của chúng đối với người dùng trực tuyến ngày càng trầm trọng. Trong bài viết này, chúng tôi đã trình bày một mô hình cụ thể để dự đoán tin tức giả trên mạng xã hội. Lựa chọn các tính năng từ bộ dữ liệu là phần thiết yếu của quy trình này vì nó được mô hình mạng nơ ron (LSTM) sử dụng để phân loại bài viết tin tức trên mạng xã hội. Trong nghiên cứu trước, tác giả đã sử dụng tập dữ liệu khác nhau[3] và khi chúng tôi so sánh mô hình của mình với mô hình đó thì độ chính xác của mô hình sẽ tốt hơn nhiều. Cách tiếp cận này đạt được độ chính xác phân loại tối đa là 83%. Trong tương lai, chúng tôi sẽ cố gắng giải quyết vấn đề phát hiện tin giả với sự trợ giúp của các thuật toán phân loại khác nhau và so sánh chúng để đưa ra kỹ thuật phân loại tốt nhất.

# VIII. TÀI LIỆU THAM KHẢO

[1] “FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media” https://www.semanticscholar.org/reader/eed1a4b3ec3b6de0fd1f0b8b2ec969b540fe41a0

[2] “KaiDMML/FakeNewsNet” https://github.com/KaiDMML/FakeNewsNet/tree/master

[3] “Fake news detection using naive Bayes classifier” <https://ieeexplore.ieee.org/document/8100379>

[4] “krishnaik06/Fake-New-LSTM” <https://github.com/krishnaik06/Fake-New-LSTM/tree/master>

[5] “Fake News Detection Analysis using Python | LSTM Classification | Deep Learning Project Tutorial” <https://www.hackersrealm.net/post/fake-news-detection-analysis-using-python>

[6] “Fake News Detection using Deep Learning – LSTM Approach” https://wisdomml.in/fake-news-detection-using-deep-learning-lstm-approach/