

**دانشکدگان علوم و فناوری های میان رشته ای**

**دانشکده سامانه های هوشمند**

**گروه علوم داده**

**شناسایی بات‌های اجتماعی با استفاده از مدل‌های گفتگو**

**نگارنده**

**مبینا طاهری**

**استاد راهنما**

**دکتر مصطفی صالحی   
استاد مشاور**

**دکتر**

**در رشته سامانه های شبکه ای**

**شهریور 1404**

****

**چکیده**

در سال‌های اخیر، رشد سریع بات‌های اجتماعی به یکی از مهم‌ترین چالش‌ها برای امنیت و اعتماد در شبکه‌های اجتماعی آنلاین تبدیل شده است. این بات‌ها با الگوبرداری از رفتار انسان‌ها، به سادگی می‌توانند جامعه هدف را فریب داده و در انتشار اطلاعات نادرست، دستکاری افکار عمومی و ایجاد کمپین‌های هماهنگ نقش‌آفرینی کنند. روش‌های سنتی تشخیص بات، به دلیل وابستگی به ویژگی‌های ایستا و پیش‌بینی‌پذیر، در مقابل بات‌های پیشرفته با رفتار منعطف کارایی لازم را ندارند.

در این پژوهش رویکردی نوین مبتنی بر مدل‌های گفتگو برای تشخیص بات‌های اجتماعی ارائه شده است. در این مسیر، ضمن پیاده‌سازی و ارزیابی چندین مدل پایه یادگیری ماشین و همچنین شبکه‌های عصبی بازگشتی (LSTM)، از مدل‌های قدرتمند مبتنی بر گفتگو (مانند Llama 3)، به منظور شبیه‌سازی و شناسایی رفتار بات‌ها استفاده شده است. همچنین جهت افزایش دقت مدل و افزایش تاب‌آوری سیستم تشخیصی در برابر بات‌های پیچیده، سازوکار شبکه‌های مولد متخاصم (GAN) در فرآیند آموزش مدل‌ها به کار گرفته شده است.

نتایج آزمایش‌ها نشان می‌دهد که استفاده از مدل‌های گفتگو مبتنی بر معماری‌های مدرن، به طور معناداری دقت و انعطاف‌پذیری در شناسایی بات‌های اجتماعی را نسبت به رویکردهای سنتی افزایش می‌دهد. کارکرد این مدل‌ها بر روی مجموعه‌داده‌های معتبر نشانگر ارتقاء عملکرد سامانه به‌ویژه در مواجهه با بات‌های هوشمند و تطبیق‌پذیر است. در مجموع، پژوهش حاضر با معرفی و ارزیابی ترکیبی از روش‌های پیشرفته مبتنی بر مدل‌های گفتگو و GAN، گامی مؤثر برای تقویت امنیت و سلامت فضای مجازی ارائه می‌دهد.

**کلمات کلیدی:** بات اجتماعی، مدل‌های گفتگو، تشخیص بات، شبکه‌های مولد متخاصم، یادگیری ماشین، شبکه‌های اجتماعی، LLM

**چکیده گرافیکی**



تقدیم به ...

به پدر و مادر عزیزم

که با عشق و فداکاری بی‌پایان، چراغ راه زندگی‌ام بوده‌اند؛

به خانواده گرامی‌ام

که همیشه پناه و همراه من بودند؛

و

به استاد بزرگوارم

جناب آقای دکتر مصطفی صالحی

که با راهنمایی‌ها و حمایت‌های ارزشمندشان این مسیر را هموار کردند.

تقدیر و تشکر

سپاس خداوند مهربان را که توفیق پژوهش و آموختن را به من ارزانی داشت.

بر خود لازم می‌دانم از همه بزرگوارانی که در مسیر انجام این پایان‌نامه مرا همراهی و یاری کردند، صمیمانه قدردانی کنم.

در گام نخست، مراتب قدردانی و سپاس خود را تقدیم می‌دارم به استاد راهنمای گرانقدر،

جناب آقای دکتر صالحی

که با راهنمایی دلسوزانه، دانش، صبر و حمایت‌های بی‌دریغ‌شان، نقش مؤثری در پیشبرد این پژوهش ایفا کردند.

از پدر و مادر عزیزم و خانواده مهربانم که با دعای خیر، محبت و فداکاری همیشگی‌شان، بزرگ‌ترین انگیزه و پشتیبان من در زندگی و تحصیل بوده‌اند، صمیمانه سپاسگزارم.

همچنین از دوستان، همکلاسی‌ها و تمامی کسانی که به هر نحو، در طی مسیر من را یاری نمودند، صمیمانه قدردانی می‌کنم.

امیدوارم کوشش این جانب، قدمی هرچند کوچک در راستای اعتلای علم و دانش این سرزمین باشد.

Logo

Description automatically generated

**تعهدنامه اصالت اثر**

اینجانب مبینا طاهری دانش‌آموخته مقطع کارشناسی ارشد در رشتۀم مهندسی فناوری اطلاعات گرایش سامانه‌های شبکه‌ای که در تاریخ **؟؟**  از پایان‌نامه خود با عنوان شناسایی بات‌های اجتماعی با استفاده از مدل‌های گفتگو با کسب درجۀ **؟؟** دفاع کرده‌ام، شرعاً و قانوناً متعهد می‌شوم:

1. مطالب مندرج در این پایان‌نامه حاصل تحقیق و پژوهش اینجانب بوده و در مواردی که از دستاوردهای علمی‌و پژوهشی دیگران اعم از پایان‌نامه یا رساله، کتاب، مقاله و غیره استفاده کرده‌ام، رعایت کامل امانت را نموده، مطابق مقررات، ارجاع و در فهرست منابع و مآخذ اقدام به ذكر آنها کرده‌ام.
2. تمامی‌یا بخشی از این پایان‌نامه قبلاً برای دریافت هیچ مدرک تحصیلی (هم سطح، پایین‌تر یا بالاتر) در سایر دانشگاه‌ها و مؤسسات آموزش عالی ارائه نشده است.
3. مقالات مستخرج از اين پایان‌نامه کاملا حاصل کار اینجانب بوده و از هر گونه جعل داده و یا تغییر اطلاعات پرهیز کرده‌ام.
4. از ارسال همزمان و یا تکراری مقالات مستخرج از اين پایان‌نامه (با بیش از 30 درصد هم‌پوشانی) به نشریات و یا همایش‌های گوناگون خودداری کرده و خواهم کرد.
5. کلیه حقوق مادّی و معنوی حاصل از این پایان‌نامه متعلق به دانشگاه تهران بوده و متعهد می‌شوم هر گونه بهره‌مندی و یا نشر دستاوردهای حاصل از این پژوهش اعم از چاپ کتاب، مقاله، ثبت اختراع و غیره (چه در زمان دانشجویی و یا بعد از فراغت از تحصیل) با کسب اجازه از تیم استادان راهنما و مشاور و حوزه پژوهشی دانشکده باشد.

در صورت اثبات تخلف (در هر زمان) مدرك تحصيلي صادر شده توسط دانشگاه تهران از درجه اعتبار ساقط و اينجانب هیچ‌گونه ادعایی نخواهم داشت.

نام و نام خانوادگی و امضای دانشجو:

مبینا طاهری

Logo

Description automatically generated

**نتیجه سامانه تشابه‌یابی (همانندجویی)**

در اینجا تصویری از نتیجه سامانه تشابه‌یابی روی پایان‌نامه که بیانگر میزان تشابه آن با کارهای پیشین است، با کیفیت مناسب آورده شود.

**فهرست مطالب**

[فصل 1‏-‏ فصل اول- مقدمه و معرفی 20](#_Toc209279714)

[1‏-‏1‏-‏ پیش‌گفتار 20](#_Toc209279715)

[1‏-‏2‏-‏ مسئله پژوهش و راهکارها 21](#_Toc209279716)

[1‏-‏2‏-‏1‏-‏ پرسش‌های پژوهش 23](#_Toc209279717)

[1‏-‏2‏-‏2‏-‏ فرضیه‌های پژوهش 23](#_Toc209279718)

[1‏-‏3‏-‏ کاربردها و ضرورت موضوع 24](#_Toc209279719)

[1‏-‏4‏-‏ هدف‌های پژوهش 25](#_Toc209279720)

[1‏-‏5‏-‏ نوآوری‌های پژوهش 25](#_Toc209279721)

[فصل 2‏-‏ فصل دوم- مبانی علمی و پژوهش‌های پیشین 27](#_Toc209279722)

[2‏-‏1‏-‏ مقدمه 27](#_Toc209279723)

[2‏-‏2‏-‏ مبانی علمی 28](#_Toc209279724)

[1-2-2 ‏مبانی اولیه شناسایی بات‌ها در شبکه‌های اجتماعی 29](#_Toc209279725)

[1-1-2-2- تعریف و ساختار شبکه‌های اجتماعی 29](#_Toc209279726)

[2-1-2-2- چالش‌ها و مشکلات شبکه‌های اجتماعی 30](#_Toc209279727)

[3-1-2-2-چرایی و ضرورت شناسایی بات‌های اجتماعی 31](#_Toc209279728)

[4-1-2-2-تحلیل نهایی و ضرورت به‌کارگیری روش‌های علمی شناسایی 32](#_Toc209279729)

[2‏-‏2‏-‏2‏-‏ مبانی روش‌های شناسایی بات‌های اجتماعی 32](#_Toc209279730)

[۲‑۲‑۲‑۱ رویکرد مبتنی بر یادگیری ماشین (Machine Learning) 33](#_Toc209279731)

[۲‑۲‑۲‑۲ شبکه‌های عصبی مصنوعی (Artificial Neural Networks) 33](#_Toc209279732)

[۲‑۲‑۲‑۳ مدل‌های پردازش زبان طبیعی (NLP) 34](#_Toc209279733)

[۴‑۲‑۲‑۲‑ مدل‌های گفتگویی (Dialogue Models) 35](#_Toc209279734)

[5-2-2-2- مدل‌های (Sequence‑to‑Sequence) Seq2Seq 36](#_Toc209279735)

[6-2-2-2- شبکه‌های مولد متخاصم (GAN) 37](#_Toc209279736)

[7-2-2-2- شبکه‌های حافظه کوتاه‌مدت بلند (LSTM) 38](#_Toc209279737)

[8-2-2-2- بازیابی‑تولید افزوده (RAG) و ابزار Ollama 40](#_Toc209279738)

[**2‏-‏3‏-‏** **پژوهش‌های پیشین** 41](#_Toc209279739)

[2‏-‏3‏-‏1‏-‏ مرور پژوهش‌های کلیدی در زمینه شناسایی بات‌های اجتماعی 41](#_Toc209279740)

[2‏-‏3‏-‏2‏-‏ مطالعات مرتبط با ویژگی‌ها و شاخص‌های شناسایی بات‌های اجتماعی 43](#_Toc209279741)

[2‏-‏3‏-‏3‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر یادگیری ماشین 44](#_Toc209279742)

[2‏-‏3‏-‏4‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر شبکه‌های عصبی مصنوعی (ANN) 45](#_Toc209279743)

[2‏-‏3‏-‏5‏- پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر LSTM 46](#_Toc209279744)

[2‏-‏3‏-‏6‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های زبانی 48](#_Toc209279745)

[2‏-‏3‏-‏7‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های توالی به توالی 49](#_Toc209279746)

[2‏-‏3‏-‏8‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر پردازش زبان طبیعی 50](#_Toc209279747)

[2‏-‏3‏-‏9‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های گفتگو 51](#_Toc209279748)

[2‏-‏3‏-‏10‏-‏ پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر GAN 52](#_Toc209279749)

[2‏-‏3‏-‏11‏-‏ پژوهش‌های مرتبط با مدل‌های زبانی بزرگ (LLM) در تولید متن بات‌های اجتماعی 53](#_Toc209279750)

[فصل 3‏-‏ فصل سوم- روش پیشنهادی 63](#_Toc209279751)

[3‏-‏1‏-‏ مقدمه 63](#_Toc209279752)

[3‏-‏2‏-‏ ساختار روش پژوهشی 64](#_Toc209279753)

[3‏-‏3‏-‏ روش اول: سیستم تشخیص بات مبتنی بر LSTM-LSTM 65](#_Toc209279754)

[3‏-‏3‏-‏1‏- معماری کلی سیستم 65](#_Toc209279755)

[3‏-‏3‏-‏2‏-‏ آماده‌سازی داده و مدیریت مجموعه داده 65](#_Toc209279756)

[3‏-‏3‏-‏3‏-‏ پیش‌پردازش متن و توکنیزاسیون 67](#_Toc209279757)

[3‏-‏3‏-‏4‏-‏ معماری مدل LSTM-LSTM (رمزگذار-رمزگشا) 68](#_Toc209279758)

[3‏-‏3‏-‏5‏-‏ تنظیمات آموزش و هایپرپارامترها 70](#_Toc209279759)

[3‏-‏3‏-‏6‏-‏ استراتژی‌های ارزیابی و جلوگیری از نشت داده 71](#_Toc209279760)

[3‏-‏4‏-‏ روش دوم: سیستم تشخیص بات مبتنی بر RAG 73](#_Toc209279761)

[3‏-‏4‏-‏1‏-‏ معماری کلی سیستم 73](#_Toc209279762)

[3‏-‏4‏-‏2‏-‏ ماژول بارگذاری داده و مدیریت مجموعه داده 75](#_Toc209279763)

[3‏-‏4‏-‏3‏-‏ ماژول تعبیه‌سازی متن با Sentence Transformers 76](#_Toc209279764)

[3‏-‏4‏-‏4‏-‏ ماژول ذخیره‌سازی برداری با ChromaDB 77](#_Toc209279765)

[3‏-‏4‏-‏5‏-‏ ماژول تحلیل و طبقه‌بندی با RAG و Llama 3.1 79](#_Toc209279766)

[3‏-‏4‏-‏6‏-‏ مقایسه روش‌ها و تحلیل نتایج 80](#_Toc209279767)

[3‏-‏4‏-‏7‏-‏ استراتژی تقسیم‌بندی داده و جلوگیری از نشت داده 81](#_Toc209279768)

[3‏-‏4‏-‏8‏-‏ مدیریت خطا و پایداری اجرا 82](#_Toc209279769)

[3‏-‏4‏-‏9‏-‏ قابلیت تفسیرپذیری و تحلیل خطای سیستم 83](#_Toc209279770)

[3‏-‏5‏-‏ خلاصه و جمع‌بندی 84](#_Toc209279771)

[فصل 4‏-‏ فصل چهارم- ارزیابی و تحلیل نتایج 86](#_Toc209279772)

[4‏-‏1‏-‏ مقدمه 86](#_Toc209279773)

[4‏-‏2‏-‏ داده‌ها 87](#_Toc209279774)

[4‏-‏3‏-‏ شرایط پیاده‌سازی 89](#_Toc209279775)

[4‏-‏4‏-‏ معیارهای ارزیابی 90](#_Toc209279776)

[4‏-‏5‏-‏ نتایج روش‌ها (گزارش یافته‌ها) 93](#_Toc209279777)

[4‏-‏5‏-‏1‏-‏ نتایج روش LSTM-LSTM روی دیتاست Cresci-2017 93](#_Toc209279778)

[4‏-‏5‏-‏2‏-‏ نتایج روش RAG روی دیتاست Cresci-2017 95](#_Toc209279779)

[4‏-‏5‏-‏3‏-‏ نتایج روش LSTM-LSTM روی دیتاست TwiBot-20 96](#_Toc209279780)

[4‏-‏5‏-‏4‏-‏ نتایج روش RAG روی دیتاست TwiBot-20 98](#_Toc209279781)

[4‏-‏6‏-‏ تحلیل و بحث 99](#_Toc209279782)

[4‏-‏6‏-‏1‏-‏ تحلیل روش LSTM-LSTM روی دیتاست Cresci-2017 100](#_Toc209279783)

[4‏-‏6‏-‏2‏-‏ تحلیل روش RAG روی دیتاست Cresci-2017 102](#_Toc209279784)

[4‏-‏6‏-‏3‏-‏ تحلیل روش LSTM-LSTM روی دیتاست TwiBot-20 105](#_Toc209279785)

[4‏-‏6‏-‏4‏-‏ تحلیل روش RAG روی دیتاست TwiBot-20 107](#_Toc209279786)

[4‏-‏7‏-‏ مقایسه با پژوهش‌های پیشین 110](#_Toc209279787)

[4‏-‏8‏-‏ خلاصه و نتیجه‌گیری 111](#_Toc209279788)

[فصل 5‏-‏ فصل پنجم- جمع‌بندی و پیشنهادهای علمی 113](#_Toc209279790)

[6‏-‏1‏-‏ جمع‌بندی و نتیجه‌گیری 113](#_Toc209279791)

[6‏-‏2‏-‏ پیشنهادها 115](#_Toc209279792)

[6‏-‏2‏-‏1‏-‏ پژوهشگران 116](#_Toc209279793)

[6‏-‏2‏-‏2‏-‏برنامه‌ریزان 116](#_Toc209279794)

[6‏-‏2‏-‏3‏-‏ کارآفرینان و صنعتگران 117](#_Toc209279795)

[منابع علمی 118](#_Toc209279796)

[فصل 7‏-‏ پیوست(ها) 124](#_Toc209279797)

[پیوست الف- نکات تکمیلی در مورد قالب و ساختار رساله/پایان‌نامه 124](#_Toc209279798)

[پیوست ب- جدول و شکل و فرمول 127](#_Toc209279799)

[پیوست پ- مراجع 130](#_Toc209279800)

[پیوست ت- ایرادهای رایج در تدوین رساله/پایان‌نامه 131](#_Toc209279801)

[پیوست ث- فارسی بنویسیم، غلط ننویسیم 133](#_Toc209279802)

**فهرست شکل‌ها**

**فهرست جدول‌ها**

**فهرست اختصارها**

(این بخش اختیاری است، در صورت ارائه به ترتیب حروف الفبا مرتب شوند)

|  |  |  |
| --- | --- | --- |
| اختصار | صورت کامل | معادل فارسی |
| GAN | Generative Adversarial Networks | شبکه‌های مولد خصمانه |
|  |  |  |
|  |  |  |

**فهرست علامت‌ها و نمادها**

(این بخش اختیاری است، در صورت ارائه به ترتیب رخداد آن علامت/نماد در صفحات رساله/پایان‌نامه ارائه شوند)

|  |  |
| --- | --- |
| علامت | معنی |
|  | بیانگر سیگنال گفتار در حوزه زمان است |
|  |  |
|  |  |

# **فصل اول- مقدمه و معرفی**

## پیش‌گفتار

گسترش سریع شبکه‌های اجتماعی در سال‌های اخیر، فضای دیجیتال را به محیطی پویا و پیچیده تبدیل کرده است که در آن مرز بین فعالیت‌های انسانی و ماشینی به‌تدریج محو می‌شود. در این میان، بات‌های اجتماعی — یعنی حساب‌های خودکاری که با هدف تقلید از رفتار کاربران انسانی طراحی شده‌اند — نقش فزاینده‌ای در انتشار اطلاعات نادرست، دستکاری افکار عمومی و تأثیرگذاری بر فرآیندهای اجتماعی و سیاسی ایفا می‌کنند. با پیشرفت فناوری‌های هوش مصنوعی، این بات‌ها امروزه قادرند مکالمات طبیعی، پیچیده و طولانی‌مدت را شبیه‌سازی کنند و به‌راحتی از دام روش‌های سنتی شناسایی عبور نمایند.

این پژوهش با هدف توسعه روشی مؤثر و نوین برای شناسایی بات‌های پیشرفته اجتماعی انجام شده است. در این راستا، رویکردی مبتنی بر مدل‌های گفتگومحور پیشنهاد شده است که با تحلیل عمیق تعاملات زبانی و دینامیک مکالمات، الگوهای غیرانسانی بات‌ها را شناسایی می‌کند. این روش با ترکیب قابلیت‌های مدل‌های یادگیری عمیق و تکنیک‌های پردازش زبان طبیعی، قادر است ناسازگاری‌های معنایی، منطقی و رفتاری در گفتگوهای آنلاین را تشخیص دهد و دقت شناسایی بات‌ها را به‌طور قابل‌توجهی افزایش دهد.

اهمیت این موضوع فراتر از حوزه فنی است؛ حفظ یکپارچگی اطلاعات، جلوگیری از گسترش شایعات و تقویت اعتماد کاربران به پلتفرم‌های اجتماعی، از جمله دلایل ضرورت توسعه راهکارهای هوشمند در این زمینه است. انگیزه این پژوهش، ترکیب دغدغه شخصی نسبت به سلامت فضای مجازی و علاقه به به‌کارگیری فناوری‌های نوین در حل چالش‌های اجتماعی است.

مراحل پژوهش به‌صورت گام‌به‌گام شامل مطالعه ادبیات، طراحی مدل، پیاده‌سازی، ارزیابی و تحلیل نتایج بوده است. در ادامه این فصل، پس از بیان دقیق مسئله، به اهمیت و ضرورت پژوهش، اهداف، سؤالات و ساختار کلی پایان‌نامه پرداخته می‌شود.

## مسئله پژوهش و راهکارها

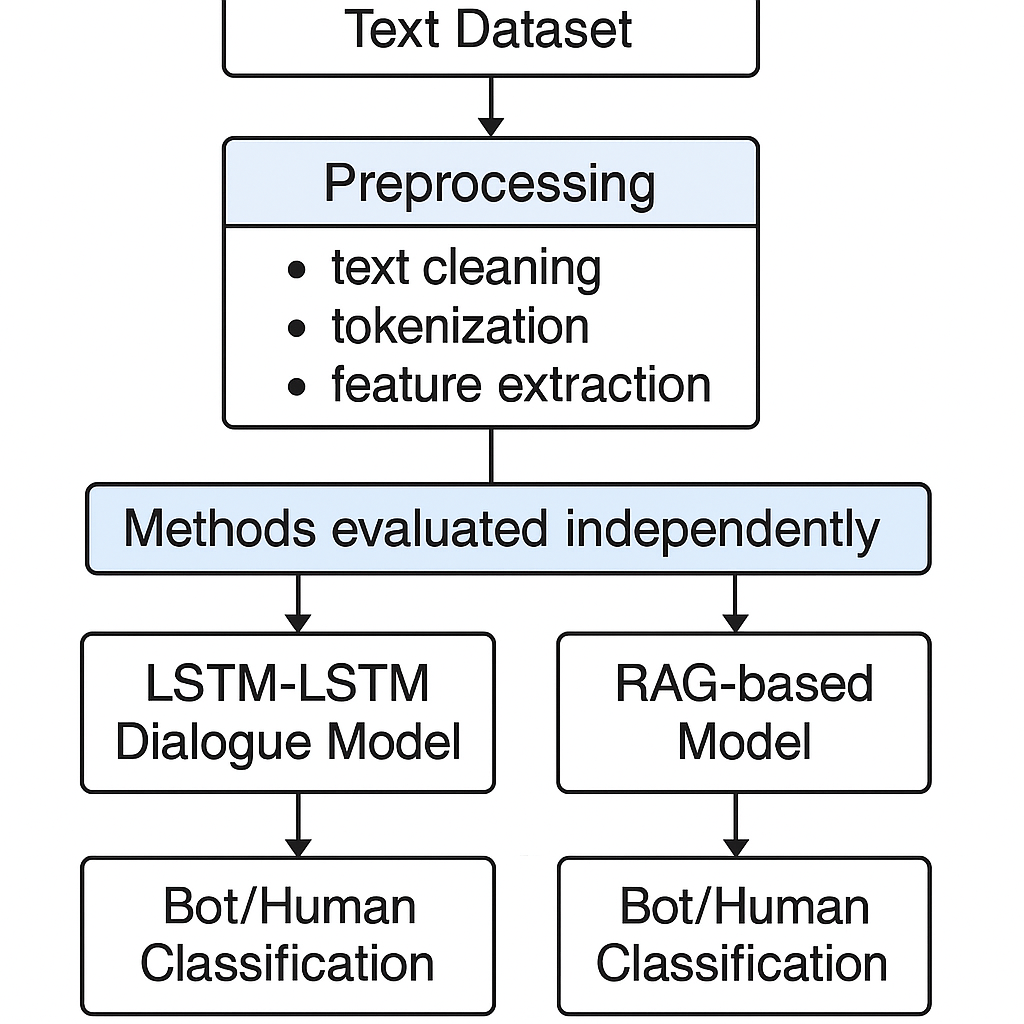
گسترش شبکه‌های اجتماعی به‌عنوان فضایی گسترده برای تعاملات عمومی، زمینه‌ساز حضور فزاینده بات‌های اجتماعی شده است. این حساب‌های خودکار، با تقلید هوشمندانه از سبک زبان، الگوهای تعامل و حتی شبکه‌های ارتباطی کاربران انسانی، قادرند به‌صورت نامحسوس در گفتگوها مشارکت کنند. این توانایی، آن‌ها را به ابزاری قدرتمند برای انتشار اطلاعات نادرست، تحریک احساسات عمومی و دستکاری روندهای اجتماعی تبدیل کرده است. در همین راستا، شناسایی این بات‌ها، به‌ویژه نسل جدید آن‌ها که از مدل‌های پردازش زبان طبیعی برای شبیه‌سازی مکالمات استفاده می‌کنند، به یکی از چالش‌های اساسی در حوزه امنیت دیجیتال و هوش مصنوعی تبدیل شده است.

روش‌های سنتی تشخیص بات، که عمدتاً متکی به ویژگی‌های ساختاری (مانند تعداد پست‌ها، الگوی زمانی ارسال یا شبکه دوستان) و الگوریتم‌های یادگیری ماشین کلاسیک مانند جنگل تصادفی یا رگرسیون لجستیک هستند، در مواجهه با بات‌های پیشرفته و پویا دچار محدودیت می‌شوند. این روش‌ها اغلب نمی‌توانند رفتارهای زبانی پیچیده، ناسازگاری‌های معنایی یا تغییرات هوشمندانه در سبک گفتار را تشخیص دهند.

در همین راستا، تحولات اخیر در یادگیری عمیق و پردازش زبان طبیعی، امکان استفاده از مدل‌های گفتگومحور را برای تحلیل عمیق تعاملات آنلاین فراهم کرده است. در این پژوهش، با هدف غلبه بر محدودیت‌های روش‌های پیشین، چارچوبی نوین بر پایه ترکیب معماری LSTM-LSTM و چارچوب RAG پیشنهاد شده است. این رویکرد با بهره‌گیری از قابلیت مدل‌های دنباله‌ای در درک زمانی مکالمات و توانایی RAG در دسترسی به دانش خارجی، قادر است نه تنها محتوای متنی را تحلیل کند، بلکه ناسازگاری‌های منطقی، تناقض‌های موضوعی و رفتارهای غیرانسانی در زنجیره تعاملات را شناسایی کند.

انتظار می‌رود این تلفیق، دقت، انعطاف و مقاومت سیستم تشخیص بات را در برابر تاکتیک‌های فریب پیشرفته به‌طور چشمگیری افزایش دهد. دستیابی به چنین راهکاری نه‌تنها به بهبود کیفیت تعاملات آنلاین کمک می‌کند، بلکه در مقابله با انتشار گسترده اطلاعات نادرست و تقویت اعتماد به فضای مجازی نیز نقش کلیدی ایفا خواهد کرد.

در ادامه، فلوچارت کلی فرایند پیشنهادی شناسایی بات اجتماعی با بهره‌گیری از مدل‌های گفتگومحور، معماری LSTM-LSTM و رویکرد تولید بازیابی‌محور (Retrieval-Augmented Generation - RAG) در شکل ‏1‑1 ارائه شده است. این فرآیند شامل دریافت ورودی‌ها (داده‌های متنی، محتوای گفتگویی و سایر فعالیت‌های کاربران)، پیش‌پردازش و استخراج ویژگی‌های چندبعدی، تحلیل و مدل‌سازی سلسله‌مراتبی داده‌ها، و در نهایت برچسب‌گذاری کاربران بر اساس احتمال انسان یا بات بودن است.



شکل ‏1‑1 – ساختار کلی فرایند پیشنهادی شناسایی بات اجتماعی مبتنی بر مدل گفتگومحور LSTM–LSTM و روش RAG به صورت دو مسیر مجزا

در این ساختار، داده‌های ورودی از شبکه‌های اجتماعی — شامل پیام‌های متنی، محتوای گفتگویی و سایر فعالیت‌های کاربران — ابتدا در بخش پیش‌پردازش، پاک‌سازی و ساختارسازی می‌شوند. سپس لایه استخراج ویژگی، شاخص‌های زبانی (linguistic features)، رفتاری (behavioral features) و ساختاری (structural features) را از داده‌ها استخراج می‌کند.

این ویژگی‌ها به دو مسیر مستقل ارسال می‌شوند:

* مدل گفتگومحور LSTM–LSTM: برای تحلیل وابستگی‌های زمانی و معنایی متون در مکالمات و شناسایی الگوهای رفتاری در توالی پیام‌ها.
* مدل RAG (Retrieval‑Augmented Generation): برای ترکیب توانایی تولید متن با پایگاه دانش بازیابی‌شده و بهره‌گیری از اطلاعات پیشین و متنی غنی در فرآیند شناسایی.

خروجی هر دو مسیر به ماژول تصمیم‌گیری ارسال شده و پس از تلفیق احتمالات و بردارهای معنایی، بر اساس آستانه تشخیص، برچسب‌گذاری حساب‌های کاربری به صورت «انسان» یا «بات اجتماعی» انجام می‌شود. این رویکرد ضمن بهره‌برداری از دقت LSTM–LSTM در مدل‌سازی توالی و قدرت RAG در غنی‌سازی دانش، انعطاف‌پذیری بالایی در شناسایی بات‌های پیچیده با الگوهای نوظهور دارد.

### پرسش‌های پژوهش

* پرسش اصلی:

عملکرد مدل‌های گفتگومحور مبتنی بر LSTM-LSTM و چارچوب بازیابی مبتنی بر تولید (RAG) در شناسایی بات‌های اجتماعی چگونه است و کدام یک از این دو رویکرد از نظر دقت و قابلیت اطمینان عملکرد بهتری دارند؟

* پرسش‌های فرعی:

1. مدل گفتگومحور LSTM-LSTM چگونه می‌تواند وابستگی‌های زمانی و معنایی در مکالمات آنلاین را تحلیل کند و به تشخیص بات‌های اجتماعی کمک کند؟
2. چارچوب RAG در شناسایی بات‌های اجتماعی چه نقشی ایفا می‌کند و آیا استفاده از دانش خارجی می‌تواند به بهبود دقت تشخیص کمک کند؟
3. کدام ویژگی‌های زبانی و رفتاری در تعاملات آنلاین، مؤثرترین معیارها برای تمایز بین کاربران انسانی و بات‌ها در هر یک از این دو رویکرد محسوب می‌شوند؟
4. عملکرد مدل LSTM-LSTM و چارچوب RAG در مقایسه با روش‌های سنتی یادگیری ماشین از نظر معیارهای دقت، فراخوانی و F1-Score چگونه است؟
5. چه چالش‌هایی در پیاده‌سازی و ارزیابی هر یک از این دو رویکرد در محیط‌های واقعی شبکه‌های اجتماعی وجود دارد و چگونه در این پژوهش بر آن‌ها پرداخته شده است؟

### فرضیه‌های پژوهش

با توجه به اهداف و روش‌شناسی پژوهش، فرضیه‌های زیر مطرح می‌شوند. این فرضیه‌ها چارچوبی برای ارزیابی تجربی عملکرد دو رویکرد گفتگومحور در شناسایی بات‌های اجتماعی فراهم می‌کنند.

فرضیه اول:

مدل گفتگومحور مبتنی بر LSTM-LSTM در مقایسه با روش‌های سنتی یادگیری ماشین، به دلیل توانایی در مدل‌سازی وابستگی‌های زمانی و معنایی در دنباله مکالمات، دقت بالاتری در شناسایی بات‌های اجتماعی دارد.

فرضیه دوم:

استفاده از چارچوب RAG در تشخیص بات‌های اجتماعی، با بهره‌گیری از دانش خارجی و شناسایی ناسازگاری‌های محتوایی، منجر به کاهش خطاهای تشخیصی، به‌ویژه در مواجهه با پیام‌های کوتاه یا مبهم، نسبت به مدل‌های بدون بازیابی دانش می‌شود.

فرضیه سوم:

بین دو رویکرد پیشنهادی، چارچوب مبتنی بر RAG از نظر معیارهای بازیابی (Recall) و F1-Score عملکرد بهتری نسبت به مدل LSTM-LSTM دارد، به‌ویژه در شناسایی بات‌هایی که از اطلاعات نادرست یا تناقض‌های موضوعی استفاده می‌کنند.

فرضیه چهارم:

هر دو رویکرد LSTM-LSTM و RAG در برابر تاکتیک‌های پیشرفته بات‌های اجتماعی (مانند شبیه‌سازی سبک زبانی انسانی یا تغییر هویت رفتاری) نسبت به روش‌های مبتنی بر ویژگی‌های متاداده، مقاومت و تاب‌آوری بیشتری از خود نشان می‌دهند.

## کاربردها و ضرورت موضوع

رشد سریع شبکه‌های اجتماعی، فضایی پویا برای تعاملات عمومی ایجاد کرده است، اما ظهور بات‌های پیشرفته با توانایی تقلید هوشمندانه از زبان و رفتار انسان، تهدیدی جدی برای صحت اطلاعات، امنیت سایبری و ثبات اجتماعی محسوب می‌شود. این بات‌ها می‌توانند افکار عمومی را دستکاری کنند، شایعات را گسترش دهند و حتی بر فرآیندهای سیاسی و اقتصادی تأثیر بگذارند.

روش‌های سنتی تشخیص بات، که بر ویژگی‌های آماری یا مدل‌های کلاسیک یادگیری ماشین متکی‌اند، در برابر این نسل از بات‌ها کارایی محدودی دارند. این پژوهش با اجرای دو رویکرد مجزا — مدل گفتگومحور LSTM-LSTM برای تحلیل عمیق دنباله‌های مکالمه و چارچوب RAG برای استفاده از دانش خارجی در شناسایی ناسازگاری‌های محتوایی — به بهبود دقت و قابلیت اطمینان سیستم‌های تشخیص می‌پردازد.

کاربردهای این پژوهش شامل:

* کاهش انتشار اطلاعات نادرست و فعالیت‌های مخرب بات‌ها در پلتفرم‌های اجتماعی،
* مقابله با عملیات نفوذ و جنگ شناختی از طریق شناسایی بات‌های سازمان‌یافته،
* حمایت از تحلیل‌گران و سیاست‌گذاران در شناسایی جریان‌های غیرطبیعی آنلاین،
* ارتقای امنیت سایبری و افزایش اعتماد کاربران به فضای دیجیتال.

اجرای این روش‌ها بر روی دیتاست واقعی، پژوهش حاضر را به یک راهکار کاربردی و نوین در مقابله با چالش‌های نوظهور فضای مجازی تبدیل کرده است.

## هدف‌های پژوهش

با توجه به چالش‌های موجود در شناسایی بات‌های اجتماعی پیشرفته، این پژوهش با اجرای دو رویکرد مجزا مبتنی بر مدل‌های گفتگومحور، به اهداف زیر دست می‌یابد.

هدف اصلی:

ارزیابی عملکرد دو روش مجزای مبتنی بر مدل گفتگومحور LSTM-LSTM و چارچوب بازیابی مبتنی بر تولید (RAG) در شناسایی بات‌های اجتماعی و مقایسه دقت، پایداری و مقاومت هر یک در برابر رفتارهای فریبنده بات‌های پیشرفته.

اهداف فرعی:

1. طراحی و پیاده‌سازی یک مدل گفتگومحور مبتنی بر معماری LSTM-LSTM برای تحلیل دنباله‌های مکالمه و شناسایی الگوهای غیرطبیعی زمانی و معنایی در رفتار بات‌ها.
2. پیاده‌سازی و ارزیابی چارچوب RAG به‌منظور استفاده از دانش خارجی در تشخیص ناسازگاری‌های محتوایی و بهبود دقت شناسایی بات در پیام‌های کوتاه یا مبهم.
3. ارزیابی و مقایسه عملکرد هر دو مدل به‌صورت مستقل نسبت به روش‌های سنتی (مانند Logistic Regression و SVM) و مدل‌های عمیق پایه (مانند LSTM تک‌مسیره) با استفاده از معیارهای دقت، بازیابی و F1-Score.
4. تحلیل نقاط قوت و محدودیت هر یک از دو رویکرد در شرایط واقعی (مانند داده‌های نامتوازن و بات‌های تکاملی) و ارائه بازخوردی عملی برای بهبود سیستم‌های آتی شناسایی بات.

## نوآوری‌های پژوهش

این پژوهش با اجرای و ارزیابی دو رویکرد مجزای مبتنی بر مدل‌های گفتگومحور، نوآوری‌هایی کلیدی در حوزه شناسایی بات‌های اجتماعی ارائه می‌دهد که به‌طور خاص در تحلیل مکالمات زبان فارسی قابل توجه است.

**نوآوری اول: اجرای و مقایسه دو رویکرد مجزای مبتنی بر مدل‌های گفتگومحور**

این پژوهش به‌صورت همزمان و مستقل، دو روش نوین را در شناسایی بات‌های اجتماعی به کار می‌گیرد:

* مدل گفتگومحور LSTM-LSTM برای تحلیل عمیق وابستگی‌های زمانی و معنایی در دنباله مکالمات،
* چارچوب RAG برای استفاده از دانش خارجی در شناسایی ناسازگاری‌های محتوایی.

اجرای این دو روش به‌صورت مجزا و مقایسه عملکرد آن‌ها در یک دیتاست واقعی، امکان ارزیابی تجربی از نقاط قوت و ضعف هر یک را فراهم می‌کند و زمینه را برای انتخاب راهکار مناسب‌تر در شرایط مختلف فراهم می‌سازد.

**نوآوری دوم: ارزیابی عملکرد مدل‌ها در شرایط واقعی و مقایسه سیستماتیک**

برخلاف بسیاری از پژوهش‌های پیشین که در شرایط ایده‌آل انجام می‌شوند، این مطالعه عملکرد هر دو مدل را در شرایط چالش‌برانگیز و نزدیک به واقعیت — از جمله داده‌های نامتوازن، پیام‌های کوتاه و مبهم، و حضور بات‌های شبیه‌سازی‌شده با رفتار انسانی — ارزیابی می‌کند. همچنین، نتایج به‌صورت کمّی و کیفی با روش‌های سنتی (مانند Logistic Regression و SVM) و مدل‌های عمیق پایه (مانند LSTM تک‌مسیره) مقایسه شده است تا برتری یا محدودیت هر رویکرد به‌صورت شفاف نشان داده شود.

**نوآوری سوم: بهره‌گیری از چارچوب RAG برای شناسایی بات‌های اجتماعی از طریق تحلیل محتوایی و بازیابی دانش**

استفاده از چارچوب RAG (بازیابی مبتنی بر تولید) در حوزه شناسایی بات‌های اجتماعی، به‌ویژه با تمرکز بر تحلیل محتوای متنی و شناسایی ناسازگاری‌های معنایی، یکی از نخستین کاربردهای نوین این معماری در تشخیص فعالیت‌های غیرانسانی است. برخلاف روش‌های سنتی که عمدتاً بر ویژگی‌های رفتاری یا ساختاری تکیه دارند، این رویکرد با دسترسی به دانش خارجی (مانند پایگاه‌های داده معتبر)، قادر است ادعاهای نادرست، تناقض‌های موضوعی و محتوای گمراه‌کننده در پیام‌های بات‌ها را شناسایی کند. این توانایی، بعد جدیدی به سیستم‌های تشخیص بات اضافه می‌کند و آن را از یک سیستم "رفتاری" به یک سیستم "معنایی-دانش‌محور" تبدیل می‌کند.

# **فصل دوم- مبانی علمی و پژوهش‌های پیشین**

## مقدمه

در سال‌های اخیر، شبکه‌های اجتماعی آنلاین به بستری گسترده برای تبادل محتوا و شکل‌دهی به جریان‌های اطلاعاتی تبدیل شده‌اند. هرچند این بسترها فرصتی بی‌سابقه برای تعامل و اشتراک‌گذاری آزاد ایجاد می‌کنند، اما زمینه‌ساز حضور و فعالیت بازیگران خودکار نیز شده‌اند. بات‌های اجتماعی، به‌عنوان حساب‌های کاربری برنامه‌ریزی‌شده برای شبیه‌سازی رفتار انسانی، قادرند با سرعت و حجم بالا در فضای مجازی فعالیت کرده و بر فرآیندهای اجتماعی، سیاسی و اقتصادی تأثیر بگذارند. پیچیدگی این پدیده، از توانایی بات‌ها برای تقلید ظریف تعاملات انسانی و مخفی ماندن در میان کاربران واقعی ناشی می‌شود.

روش‌های سنتی شناسایی بات‌ها، عموماً بر اساس الگوهای قابل پیش‌بینی یا ویژگی‌های محدود فنی عمل می‌کنند. این رویکردها در مواجهه با نسل جدید بات‌ها، که از تکنیک‌های پیشرفته پردازش زبان طبیعی و یادگیری ماشینی بهره می‌گیرند، عملکردی پایدار و دقیق ندارند. از این رو، نیاز به پژوهش‌هایی با رویکردی عمیق‌تر و چندوجهی احساس می‌شود؛ رویکردی که هم پیچیدگی رفتاری بات‌ها را در نظر گیرد و هم توان بهره‌گیری از مدل‌های پیشرفته یادگیری عمیق را داشته باشد.

تحلیل دقیق چالش‌های موجود در شناسایی بات‌ها، مستلزم شناخت کامل از محیط فعالیت آن‌ها و مکانیزم‌های فنی قابل استفاده است. در این بستر، شناخت ساختار شبکه‌های اجتماعی و ویژگی‌های مرتبط با تولید و انتشار محتوا اهمیت می‌یابد؛ ساختاری که می‌تواند شامل الگوهای انتشار اطلاعات، ارتباطات مبتنی بر گره‌ها و تراکم یا پراکندگی تعاملات باشد. به همین ترتیب، بررسی راهکارهای فنی متکی بر یادگیری ماشین، شبکه‌های عصبی و مدل‌های زبانی، زمینه‌ساز انتخاب روش‌های مؤثر شناسایی خواهد بود.

به منظور ایجاد بنیان نظری این تحقیق، در این فصل ابتدا به مرور جنبه‌های اجتماعی و بستر فعالیت بات‌ها پرداخته می‌شود، سپس طیفی از روش‌های علمی موجود برای شناسایی آن‌ها معرفی می‌گردد. این توالی، امکان پیوند منطقی میان ماهیت مسئله و راهکارهای فنی را فراهم ساخته و مسیر ورود به بخش‌های تحلیلی و اجرایی پژوهش را هموار می‌کند.

## مبانی علمی

گسترش سریع شبکه‌های اجتماعی، این فضاها را به محورهای اصلی تبادل اطلاعات، تعامل عمومی و شکل‌گیری افکار جمعی تبدیل کرده است. این پویایی، اگرچه فرصت‌های بی‌سابقه‌ای برای ارتباط و مشارکت فراهم کرده، اما زمینه‌ساز فعالیت بات‌های اجتماعی نیز شده است. این حساب‌های خودکار، با شبیه‌سازی هوشمندانه رفتار کاربران انسانی — از طریق الگوهای زبانی، زمانی و تعاملی — قادرند در جریان‌های اطلاعاتی نفوذ کنند، افکار عمومی را دستکاری نمایند و انتشار اطلاعات نادرست را تسهیل کنند.

با پیشرفت فناوری‌های هوش مصنوعی، به‌ویژه مدل‌های زبانی بزرگ (LLMs)، بات‌های امروزی توانایی تولید محتوایی طبیعی، متماسک و شبیه به گفتار انسان را کسب کرده‌اند. این تحول، روش‌های سنتی شناسایی بات — که عمدتاً متکی به ویژگی‌های متاداده (مانند تعداد پست‌ها یا شبکه دنبال‌کنندگان) و الگوریتم‌های یادگیری ماشین کلاسیک (مانند SVM یا Random Forest) هستند — را دچار کاهش دقت و مقاومت نموده است. این بات‌ها دیگر با رفتارهای ساده و قابل پیش‌بینی شناسایی نمی‌شوند؛ بلکه با مشارکت در مکالمات چندمرحله‌ای و استفاده از زبان طبیعی، به چالشی پیچیده تبدیل شده‌اند.

در پاسخ به این چالش، تمرکز پژوهش‌ها به سمت روش‌هایی شیفت یافته است که بتوانند رفتار زبانی و تعاملی بات‌ها را تحلیل کنند. در این راستا، مدل‌های گفتگومحور مبتنی بر یادگیری عمیق — از جمله معماری‌های دنباله‌ای مانند LSTM-LSTM و چارچوب‌های دانش‌محور مانند RAG — به عنوان رویکردهای نوین مطرح شده‌اند. این مدل‌ها نه تنها ساختار متن را تحلیل می‌کنند، بلکه قادرند وابستگی‌های زمانی، ناسازگاری‌های معنایی و حتی تناقض‌های موضوعی در محتوای بات‌ها را شناسایی کنند.

هدف این فصل، ارائه چارچوبی نظری برای درک پدیده بات‌های اجتماعی و مرور سیستماتیک روش‌های موجود در شناسایی آن‌هاست. ابتدا به بررسی ویژگی‌های کلیدی شبکه‌های اجتماعی و الگوهای فعالیت بات‌ها پرداخته می‌شود، سپس روش‌های تشخیص — از روش‌های سنتی تا مدل‌های گفتگومحور — مورد تحلیل قرار می‌گیرند. این مبانی، زمینه را برای توجیه و ارائه رویکردهای پیشنهادی این پژوهش (استفاده از LSTM-LSTM و RAG به‌صورت مجزا) فراهم می‌کنند.

### مبانی اولیه شناسایی بات‌ها در شبکه‌های اجتماعی

تشخیص دقیق بات‌های اجتماعی مستلزم شناخت پایه‌ای از ماهیت شبکه‌های اجتماعی و ویژگی‌های ساختاری و کارکردی آن‌هاست. این شناخت نه‌تنها چارچوبی برای تحلیل رفتار بات‌ها فراهم می‌کند، بلکه نقاط آسیب‌پذیر و امکانات سوءاستفاده را نیز آشکار می‌سازد. شبکه‌های اجتماعی به‌صورت گراف‌هایی از کاربران (گره‌ها) و روابط بین آن‌ها (یال‌ها) مدل می‌شوند که ویژگی‌هایی مانند درجه گره، تراکم و خوشه‌بندی، الگوهای غیرطبیعی بات‌ها را قابل شناسایی می‌کند. سیر تاریخی این پلتفرم‌ها — از نخستین شبکه‌ها مانند SixDegrees تا توییتر و فیسبوک — نشان‌دهنده تحول به سمت تعاملات سریع، گسترده و چندرسانه‌ای است که زمینه‌ساز فعالیت گسترده بات‌ها شده است. همچنین، ویژگی‌های عملکردی مانند الگوهای زمانی فعالیت، سازوکارهای بازخورد (لایک، ریتوییت) و پویایی انتشار محتوا، تفاوت‌های کلیدی میان رفتار انسانی و ماشینی را آشکار می‌کنند. در مجموع، درک این مؤلفه‌های ساختاری، تاریخی و عملکردی، پایه‌ای ضروری برای طراحی و ارزیابی سیستم‌های شناسایی بات محسوب می‌شود.

#### **1-1-2-2- تعریف و ساختار شبکه‌های اجتماعی**

تشخیص دقیق بات‌های اجتماعی مستلزم درکی جامع از ساختار و پویایی شبکه‌های اجتماعی است. این شبکه‌ها به‌صورت گراف‌های پویا مدل می‌شوند که در آن کاربران به‌عنوان گره و روابط ارتباطی به‌عنوان یال در نظر گرفته می‌شوند. ویژگی‌هایی مانند درجه گره، تراکم شبکه و ساختار خوشه‌ای می‌توانند الگوهای غیرطبیعی بات‌ها — مانند شبکه‌های دنبال‌کنندگان غیرمتوازن یا تعاملات غیرانسانی — را نشان دهند.

همچنین، تحولات تاریخی پلتفرم‌ها — از اولین شبکه‌ها مانند SixDegrees تا توییتر و فیسبوک — نشان‌دهنده تحول به سمت تعاملات سریع، گسترده و چندرسانه‌ای است که زمینه‌ساز فعالیت گسترده بات‌ها شده است. در کنار این، ویژگی‌های عملکردی مانند الگوهای زمانی فعالیت، سازوکارهای بازخورد (لایک، ریتوییت) و پویایی انتشار محتوا، تفاوت‌های رفتاری میان کاربران انسانی و بات‌ها را آشکار می‌کنند.

در مجموع، درک این مبانی ساختاری، تاریخی و عملکردی، نه‌تنها زمینه را برای شناسایی بات‌ها فراهم می‌کند، بلکه ضرورت توسعه روش‌هایی فراتر از تحلیل ساختاری — مانند مدل‌های گفتگومحور مبتنی بر تحلیل عمیق محتوای مکالمات — را نیز توجیه می‌نماید.

**تاریخچه کوتاه شبکه‌های اجتماعی**

شبکه‌های اجتماعی از دهه ۱۹۹۰ با معرفی اولین پلتفرم‌های دارای پروفایل کاربری و روابط دیجیتال آغاز شدند و به تدریج به فضاهایی پویا برای تعاملات گسترده تبدیل شدند. ورود شبکه‌های نسل بعدی با تمرکز بر جریان‌های زنده محتوا، شخصی‌سازی هوشمند و تعاملات لحظه‌ای، سرعت و دامنه انتشار اطلاعات را به شدت افزایش داد. گسترش اینترنت همراه و ظهور پلتفرم‌های مبتنی بر محتوای چندرسانه‌ای، این پویایی را تشدید کرد. این تحولات فناورانه، اگرچه تجربه کاربری را غنی‌تر کرد، اما زمینه‌ساز فعالیت گسترده بات‌های اجتماعی نیز شد. افزایش سرعت انتشار، توانایی نفوذ در جریان‌های داغ و کم‌رنگ شدن مرز میان محتوای انسانی و ماشینی، ضرورت توسعه روش‌های پیشرفته و مقاوم در شناسایی بات‌ها را برجسته کرده است.

**ویژگی‌های کلیدی شبکه‌های اجتماعی و پیوند آن با فعالیت بات‌ها**

شبکه‌های اجتماعی مدرن با ویژگی‌هایی مانند مقیاس‌پذیری بالا، سرعت انتشار محتوا، شخصی‌سازی پیشنهادات، یکپارچگی با ابزارهای خودکار و ثبت‌نام آزاد، امکان تعامل گسترده و لحظه‌ای را فراهم کرده‌اند. این قابلیت‌ها اگرچه برای کاربران انسانی مفید هستند، اما زمینه‌ساز سوءاستفاده گسترده توسط بات‌های اجتماعی نیز شده‌اند. مقیاس‌پذیری بالا و سرعت توزیع محتوا به بات‌ها امکان می‌دهد پیام‌های تکراری یا گمراه‌کننده را در کمتر از چند دقیقه به میلیون‌ها کاربر برسانند. سیستم‌های شخصی‌سازی نیز می‌توانند مورد سوءاستفاده قرار گیرند تا محتوای هدفمند به گروه‌های آسیب‌پذیر منتقل شود. همچنین، دسترسی آزاد به APIها و ابزارهای زمان‌بندی، امکان ایجاد چرخه‌های خودکار تولید و انتشار محتوا را بدون نیاز به دخالت انسانی فراهم می‌کند. در کنار این، عدم احراز هویت قوی در برخی پلتفرم‌ها، ایجاد شبکه‌های گسترده باتی با هزینه کم را ممکن ساخته است. در مجموع، این ویژگی‌ها نه تنها زمینه فعالیت بات‌ها را گسترش داده‌اند، بلکه ضرورت توسعه روش‌های تشخیصی هوشمند و فراتر از تحلیل ساختاری — مانند مدل‌های گفتگومحور مبتنی بر تحلیل محتوا — را برجسته کرده‌اند.

#### **2-1-2-2- چالش‌ها و مشکلات شبکه‌های اجتماعی**

شبکه‌های اجتماعی با تسهیل ارتباط و تبادل داده‌ها، فضایی پویا ایجاد کرده‌اند که در کنار مزایای آن، چالش‌های متعددی نیز رخ داده است. بسیاری از این چالش‌ها مرتبط با فعالیت حساب‌های خودکار هستند و نیازمند رویکردی میان‌رشته‌ای برای شناسایی و مقابله با آن‌ها می‌باشند. این چالش‌ها عبارتند از:

* انتشار اسپم و محتوای ناخواسته: بات‌ها با ارسال پیام‌های تکراری و تبلیغاتی، حجم بالایی از محتوای نامطلوب را در شبکه‌ها تولید می‌کنند که می‌تواند سیستم‌های تشخیص مبتنی بر رفتار ساده را فریب دهد.
* اجرای حملات هماهنگ: تشکیل شبکه‌های سازمان‌یافته از حساب‌های باتی برای ترویج روایت‌های خاص و دستکاری الگوریتم‌های پیشنهاد محتوا، مستلزم تحلیل هم‌زمان ابعاد زمانی، محتوایی و ساختاری تعاملات است.
* دستکاری اطلاعات: انتشار اطلاعات ناقص، گمراه‌کننده یا جعلی توسط بات‌ها، با تقویت توسط سیستم‌های شخصی‌سازی، منجر به شکل‌گیری «اتاق‌های پژواک» و قطبی‌سازی افکار عمومی می‌شود.
* مسائل امنیتی و کاهش اعتماد کاربری: سوءاستفاده از آسیب‌پذیری‌های سیستمی، نفوذ به حساب‌های کاربری، حملات فیشینگ و گسترش اطلاعات جعلی، ضمن ایجاد تهدیدات فردی و سازمانی، به تدریج اعتماد کاربران به پلتفرم‌های اجتماعی را تضعیف می‌کنند.

این چالش‌ها نشان می‌دهند که هر راهکار مؤثر شناسایی بات‌ها باید دارای درک عمیقی از تعاملات انسانی و ماشینی باشد تا بتواند فعالیت‌های مشروع را از سوءاستفاده‌های هوشمندانه تمییز دهد.

#### **3-1-2-2-چرایی و ضرورت شناسایی بات‌های اجتماعی**

شناسایی و مهار بات‌های اجتماعی فراتر از یک چالش فنی، یک ضرورت راهبردی در حفاظت از امنیت اطلاعاتی، اجتماعی و اقتصادی محسوب می‌شود. شبکه‌های اجتماعی با ساختاری باز و مقیاس‌پذیر، امکان تأثیرگذاری گسترده حتی توسط تعداد محدودی بات را فراهم کرده‌اند؛ تأثیری که می‌تواند در ابعاد اقتصادی، سیاسی و اجتماعی بروز کند. تقلید هوشمندانه بات‌ها از رفتار کاربران انسانی و توانایی آن‌ها در دستکاری جریان اطلاعات، منجر به تضعیف اعتماد عمومی به پلتفرم‌های دیجیتال می‌شود و بحران‌های اجتماعی را تشدید می‌کند. از سوی دیگر، فعالیت این بات‌ها پیامدهای اقتصادی قابل‌توجهی دارد که از ایجاد نوسانات مصنوعی در بازارها و گسترش تبلیغات جعلی تا اجرای حملات فیشینگ و کلاهبرداری آنلاین امتداد می‌یابد. در حوزه سیاسی نیز، استفاده از بات‌ها برای انتشار محتوای قطبی، تحریک افکار عمومی و دستکاری در فرآیندهای انتخاباتی، یکی از تهدیدهای جدی برای دموکراسی و شفافیت اطلاعاتی است. در مجموع، این چالش‌های چندبعدی، شناسایی بات‌ها را به یک ضرورت حیاتی برای سلامت اکوسیستم دیجیتال تبدیل کرده و توسعه روش‌های پیشرفته و هوشمند — از جمله مدل‌های گفتگومحور مبتنی بر تحلیل عمیق محتوا — را اجتناب‌ناپذیر ساخته است.

#### **4-1-2-2-تحلیل نهایی و ضرورت به‌کارگیری روش‌های علمی شناسایی**

مرور ساختاری، عملیاتی و پیامدهای اقتصادی، اجتماعی و سیاسی فعالیت بات‌های اجتماعی نشان می‌دهد که این پدیده یک ناهنجاری موقت نیست، بلکه یک چالش سیستماتیک و پویا محسوب می‌شود. بات‌ها با گذشت زمان نه‌تنها از نظر فناوری، بلکه در رفتار و استراتژی‌های تعاملی نیز تحول یافته‌اند. این تحول، ریشه در ماهیت تعاملی و تطبیقی آن‌ها دارد؛ بات‌ها با شبیه‌سازی هوشمندانه رفتار کاربران انسانی و تطبیق پیام‌ها با زمینه ارتباطی، قادرند از سد روش‌های تشخیص مبتنی بر قواعد ثابت و ویژگی‌های ایستا عبور کنند.

این دینامیک، نوعی «مسابقه تسلیحاتی اطلاعاتی» را ایجاد کرده است: هر پیشرفت در روش‌های شناسایی، با پاسخی هوشمندانه‌تر از سوی بات‌ها — با افزایش ظرافت در تقلید زبانی، تغییر هویت رفتاری یا استفاده از مدل‌های زبانی بزرگ — همراه می‌شود. در چنین شرایطی، به‌روزرسانی و توسعه مستمر تکنیک‌های تشخیص نه یک گزینه، بلکه یک ضرورت انکارناپذیر است.

همزمان، حجم عظیم داده، سرعت بالای انتشار محتوا و تعاملات چندلایه بین پلتفرم‌ها، امکان تشخیص بات را تنها با روش‌های هوشمند، تطبیقی و مبتنی بر یادگیری ماشین ممکن ساخته است. ترکیب تحلیل شبکه‌ای، پردازش زبان طبیعی و مدل‌های یادگیری عمیق، نه‌تنها دقت شناسایی را افزایش می‌دهد، بلکه امکان پاسخ پیشگیرانه به تاکتیک‌های نوظهور را نیز فراهم می‌کند.

در نتیجه، به‌کارگیری چارچوب‌های علمی و الگوریتمی پیشرفته در شناسایی بات‌ها، یک انتخاب اختیاری نیست، بلکه یک ضرورت راهبردی برای حفظ یکپارچگی، شفافیت و سلامت اکوسیستم دیجیتال است. در ادامه، مبانی نظری و چارچوب‌های فنی روش‌های شناسایی بات — از رویکردهای سنتی تا مدل‌های پیشرفته مبتنی بر یادگیری عمیق — مورد بررسی قرار خواهند گرفت.

### مبانی روش‌های شناسایی بات‌های اجتماعی

شناسایی بات‌های اجتماعی در طول زمان از روش‌های ساده مبتنی بر ویژگی‌های متاداده و یادگیری ماشین کلاسیک به سمت مدل‌های پیشرفته مبتنی بر یادگیری عمیق و پردازش زبان طبیعی حرکت کرده است. در حالی که روش‌های اولیه بر تحلیل رفتاری و ساختاری متکی بودند، مدل‌های امروزی قادرند محتوای متنی، پویایی مکالمات و حتی ناسازگاری‌های معنایی را تحلیل کنند. این تحول، با ظهور معماری‌هایی مانند LSTM و چارچوب‌های دانش‌محور مانند RAG، به سمت شناسایی هوشمندانه‌تر و عمیق‌تر بات‌ها پیش رفته است. در ادامه، روش‌های پیشین در شناسایی بات‌ها از رویکردهای سنتی تا مدل‌های گفتگومحور مرور می‌شوند.

#### **۲‑۲‑۲‑۱ رویکرد** **مبتنی بر** **یادگیری ماشین (Machine Learning)**

یادگیری ماشین به‌عنوان یک چارچوب آماری-محاسباتی، پایه‌ای اولیه برای شناسایی بات‌های اجتماعی فراهم کرده است. این رویکرد با آموزش مدل‌ها بر اساس ویژگی‌های زمانی، محتوایی و ساختاری داده‌ها، امکان تشخیص الگوهای غیرطبیعی رفتار بات‌ها را فراهم می‌کند. با استفاده از روش‌هایی مانند یادگیری نظارتی، نیمه‌نظارتی و بدون نظارت، مدل‌ها قادرند بین رفتار انسانی و ماشینی تمایز قائل شوند. اگرچه این روش‌ها در مواجهه با بات‌های پیشرفته دچار محدودیت می‌شوند، اما به‌عنوان بستری برای توسعه مدل‌های پیشرفته‌تر مبتنی بر یادگیری عمیق و پردازش زبان طبیعی عمل کرده‌اند.

**الگوریتم‌های یادگیری ماشین**

الگوریتم‌های یادگیری ماشین کلاسیک، از جمله ماشین بردار پشتیبان (SVM)، درخت تصمیم، بیز ساده، k-نزدیک‌ترین همسایه (KNN) و روش‌های خوشه‌بندی، نخستین رویکردهای سیستماتیک در شناسایی بات‌های اجتماعی بودند. این روش‌ها با تحلیل ویژگی‌های قابل اندازه‌گیری — مانند تعداد پست‌ها، الگوی زمانی فعالیت، نسبت تعاملات، یا ساختار شبکه دنبال‌کنندگان — سعی می‌کردند بین رفتار کاربران انسانی و حساب‌های خودکار تمایز قائل شوند. با استفاده از داده‌های برچسب‌دار و مدل‌های آماری، این الگوریتم‌ها قادر بودند در شناسایی بات‌های ساده و قابل پیش‌بینی عملکرد قابل‌قبولی داشته باشند.

با این حال، کارایی این روش‌ها در مواجهه با بات‌های پیشرفته به شدت محدود است. این محدودیت‌ها عمدتاً ناشی از دو عامل هستند: اول، وابستگی به مهندسی دستی ویژگی‌ها که نیازمند دانش پیشین از رفتار بات‌هاست و در برابر تاکتیک‌های نوظهور ناکارآمد می‌ماند؛ دوم، ناتوانی در مدل‌سازی رفتارهای پیچیده و زبانی که بات‌های مدرن با استفاده از مدل‌های هوش مصنوعی از خود نشان می‌دهند.

این چالش‌ها، ضرورت توسعه روش‌هایی هوشمندتر و خودکارتر را آشکار کرد. در این راستا، تمرکز پژوهش‌ها به سمت شبکه‌های عصبی مصنوعی و به‌ویژه یادگیری عمیق سوق یافت؛ جایی که مدل‌ها می‌توانند بدون نیاز به ویژگی‌سازی دستی، از داده‌های خام (مانند متن یا دنباله تعاملات) بازنمایی‌های عمیق یاد بگیرند و الگوهای پیچیده را به‌صورت خودکار تشخیص دهند. این گذار، زمینه را برای ظهور مدل‌های گفتگومحور مانند LSTM و چارچوب‌های دانش‌محور مانند RAG فراهم کرد — که هر دو در راستای شناسایی بات‌هایی با رفتار زبانی پیچیده توسعه یافته‌اند.

#### **۲‑۲‑۲‑۲ شبکه‌های عصبی مصنوعی (Artificial Neural Networks)**

شبکه‌های عصبی مصنوعی (Artificial Neural Networks - ANN) به‌عنوان چارچوبی کلیدی در هوش مصنوعی، امکان یادگیری الگوهای پیچیده از داده‌های خام را فراهم کرده‌اند. برخلاف روش‌های کلاسیک یادگیری ماشین که به مهندسی دستی ویژگی‌ها وابسته‌اند، ANNها قادرند بازنشانی‌های عمیق و غیرخطی از داده‌ها — مانند متن، سری‌های زمانی و ساختارهای شبکه‌ای — را به‌صورت خودکار یاد بگیرند. این ویژگی، آن‌ها را به انتخابی طبیعی برای شناسایی بات‌های اجتماعی تبدیل کرده است، به‌ویژه زمانی که رفتار بات‌ها از طریق تعاملات متنی و پویایی زمانی بروز می‌کند.

در این حوزه، معماری‌های مختلف ANN کاربردهای متفاوتی داشته‌اند. پرسپترون چندلایه (MLP) برای تحلیل ویژگی‌های آماری و ساختاری حساب‌ها استفاده شده، اما در مواجهه با داده‌های ترتیبی محدود است. شبکه‌های عصبی پیچشی (CNN) برای استخراج الگوهای محلی در متن یا سری‌های زمانی مفید بوده‌اند، اما وابستگی‌های بلندمدت را به‌خوبی مدل نمی‌کنند. در مقابل، شبکه‌های عصبی بازگشتی (RNN) و به‌ویژه نسخه‌های پیشرفته آن مانند LSTM (حافظه بلندمدت)، به دلیل توانایی در مدل‌سازی دنباله‌های زمانی و وابستگی‌های معنایی، برای تحلیل مکالمات متنی و شناسایی رفتارهای غیرانسانی بات‌ها ابزاری ایده‌آل محسوب می‌شوند.

این تحول از مدل‌های عمومی به سمت معماری‌های تخصصی برای داده‌های ترتیبی، زمینه را برای ادغام ANN با حوزه پردازش زبان طبیعی (NLP) فراهم کرد. امروزه، مدل‌های گفتگومحور مبتنی بر LSTM و چارچوب‌های دانش‌محور مانند RAG، به‌عنوان نسل جدیدی از سیستم‌های تشخیص بات، از توانایی ANN در درک پویایی زبان و تعاملات مکالمه‌ای بهره می‌برند.

#### **۲‑۲‑۲‑۳ مدل‌های پردازش زبان طبیعی (NLP)**

پردازش زبان طبیعی (Natural Language Processing – NLP) به‌عنوان یکی از حوزه‌های کلیدی هوش مصنوعی، امکان تحلیل، درک و تولید زبان انسانی توسط سیستم‌های محاسباتی را فراهم کرده است. در حوزه شناسایی بات‌های اجتماعی، این اهمیت ویژه‌ای دارد، چراکه فعالیت بات‌ها عمدتاً از طریق تعاملات متنی — پست، کامنت، پیام و مکالمه — بروز می‌کند. مدل‌های NLP با استخراج ویژگی‌های زبانی، معنایی و سبکی از داده‌های خام متنی، قادرند رفتارهای غیرانسانی و تقلیدی بات‌ها را تشخیص دهند.

اولین رویکردها در NLP بر پایه مدل‌های آماری و برداری مانند TF-IDF و Word2Vec بنا شدند که ویژگی‌های سطحی مانند بسامد واژه یا n-گرام را استخراج می‌کردند. این روش‌ها اگرچه در شناسایی الگوهای تکراری و سبک‌های نوشتاری ساده مؤثر بودند، اما ناتوانی در درک معنای عمیق و وابستگی‌های زمانی متن، کارایی آن‌ها را در مواجهه با بات‌های پیشرفته محدود کرد.

این محدودیت، ورود شبکه‌های عصبی عمیق را به حوزه NLP ضروری ساخت. در این راستا، مدل‌هایی مانند CNN و به‌ویژه شبکه‌های بازگشتی (RNN) و نسخه پیشرفته آن LSTM، امکان مدل‌سازی دنباله‌های زمانی و وابستگی‌های بلندمدت در متن را فراهم کردند. LSTM با مکانیزم حافظه بلندمدت، قادر است رفتارهای پیچیده در مکالمات چندمرحله‌ای — مانند ناسازگاری لحن، پاسخ‌های غیرطبیعی یا تکرار الگوهای ساختاری — را شناسایی کند. این توانایی، آن را به یکی از معماری‌های محوری در تحلیل تعاملات بات‌ها تبدیل کرده است.

همزمان، ظهور مدل‌های پیش‌آموزش‌دیده مبتنی بر معماری ترنسفورمر و مکانیزم توجه، تحولی بنیادین در NLP ایجاد کرد. این مدل‌ها با یادگیری از حجم عظیمی از داده‌های متنی، بازنمایی‌هایی غنی و زمینه‌مند از زبان ایجاد می‌کنند و قادرند ناهماهنگی‌های معنایی، تناقض‌های موضوعی و تغییرات غیرطبیعی در سبک گفتار را تشخیص دهند.

اما چالش اصلی همچنان باقی ماند: بسیاری از مدل‌های NLP، متن را به صورت واحدی مستقل تحلیل می‌کنند، در حالی که بات‌های امروزی در قالب مکالمات پویا و چندمرحله‌ای فعالیت می‌کنند. این نیاز، به ظهور مدل‌های گفتگومحور انجامید — مدل‌هایی که نه تنها محتوای یک پیام، بلکه کل تاریخچه تعامل را به عنوان ورودی در نظر می‌گیرند. این رویکرد، امکان تحلیل پویایی مکالمه، ثبات سبک، و هماهنگی منطقی را فراهم می‌کند.

در این مسیر، چارچوب RAG (بازیابی مبتنی بر تولید) به عنوان یک نوآوری کلیدی ظهور کرد. RAG با ترکیب قابلیت بازیابی دانش از منابع خارجی و تولید متن طبیعی، قادر است ادعاهای نادرست، تناقض‌های موضوعی و محتوای گمراه‌کننده در پیام‌های بات‌ها را شناسایی کند. این ویژگی، آن را به ابزاری قدرتمند در تشخیص بات‌هایی تبدیل کرده که از زبان طبیعی اما محتوای غلط استفاده می‌کنند.

در نتیجه، تحول از مدل‌های آماری به سمت مدل‌های گفتگومحور مبتنی بر LSTM و RAG، نشان‌دهنده حرکت به سمت روش‌هایی عمیق، معنایی و تعاملی در شناسایی بات‌ها است — روش‌هایی که دقیقاً در راستای پژوهش حاضر قرار دارند.

#### **۴‑۲‑۲‑۲‑ مدل‌های گفتگویی (Dialogue Models)**

با تحول بات‌های اجتماعی از حالت تولید محتوای ایستا به فعالیت‌های پویا و مبتنی بر مکالمه، روش‌های سنتی تحلیل متن منفرد دیگر کافی نیستند. بات‌های امروزی در گفتگوهای چندمرحله‌ای شرکت می‌کنند، به پرسش‌ها پاسخ می‌دهند و ظاهراً از حافظه و درک زمینه استفاده می‌کنند. این تحول، نیاز به سیستم‌هایی را آشکار کرد که بتوانند کل جریان تعامل را تحلیل کنند، نه فقط محتوای هر پیام به تنهایی.

در این راستا، مدل‌های گفتگومحور (Dialogue Models) به عنوان نسل جدیدی از سیستم‌های پردازش زبان طبیعی ظهور کردند. این مدل‌ها برخلاف رویکردهای قبلی که هر پیام را به‌صورت مستقل پردازش می‌کردند، تاریخچه کامل مکالمه را به‌عنوان ورودی در نظر می‌گیرند و قادرند روابط معنایی، زمانی و منطقی بین پیام‌های متوالی را مدل کنند. این ویژگی، آن‌ها را به ابزاری قدرتمند در شناسایی رفتارهای غیرانسانی بات‌ها تبدیل کرده است.

مدل‌های گفتگومحور قادرند الگوهایی را شناسایی کنند که در تحلیل تک‌پیامی نامرئی می‌مانند، مانند:

* ناهماهنگی لحن در طول مکالمه،
* عدم انسجام منطقی (پاسخ‌های نامرتبط با محتوای قبلی)،
* تکرار ساختارهای پیش‌ساخته،
* یا عدم حافظه زمینه‌ای (فراموشی پرسش‌های قبلی).

این مدل‌ها اغلب بر پایه معماری‌های پیشرفته‌ای مانند شبکه‌های بازگشتی بلندمدت (LSTM) یا ترنسفورمرها ساخته می‌شوند و با آموزش بر روی داده‌های مکالمه‌ای، الگوهای طبیعی تعامل انسانی را یاد می‌گیرند. هر انحراف از این الگوها می‌تواند به عنوان نشانه‌ای از فعالیت بات تفسیر شود.

همچنین، برخی از این مدل‌ها با ادغام منابع دانش خارجی یا سیستم‌های بازیابی اطلاعات، قادرند محتوای مکالمه را از نظر واقعیت و سازگاری موضوعی نیز ارزیابی کنند. این رویکرد، امکان شناسایی بات‌هایی را فراهم می‌کند که اگرچه از نظر ساختار زبانی طبیعی به نظر می‌رسند، اما ادعاهای نادرست یا متضاد مطرح می‌کنند.

در نتیجه، مدل‌های گفتگومحور نه تنها درک عمیق‌تری از رفتار زبانی بات‌ها فراهم می‌کنند، بلکه مرز تشخیص را از سطح متنی به سطح تعاملی و معنایی ارتقا می‌دهند. این تحول، زمینه را برای توسعه سیستم‌های هوشمند و مقاوم در برابر بات‌های پیشرفته فراهم کرده است.

#### **5-2-2-2- مدل‌های (Sequence‑to‑Sequence) Seq2Seq**

مدل‌های Sequence-to-Sequence (Seq2Seq) یکی از معماری‌های کلیدی در یادگیری عمیق برای پردازش داده‌های ترتیبی هستند که هدف آن‌ها تبدیل یک دنباله ورودی به یک دنباله خروجی با ساختار متفاوت است. این مدل‌ها ابتدا ورودی (مانند یک مکالمه چندمرحله‌ای) را با استفاده از یک رمزگذار (Encoder) به یک بازنمایی فشرده و معنادار تبدیل می‌کنند، سپس یک رمزگشا (Decoder) این بازنمایی را برای تولید یا تحلیل خروجی (مانند پاسخ بعدی یا برچسب رفتاری) به کار می‌گیرد.

در حوزه شناسایی بات‌های اجتماعی، این معماری اهمیت ویژه‌ای دارد، چرا که بات‌های پیشرفته دیگر فقط پیام‌های منفرد تولید نمی‌کنند، بلکه در مکالمات چندمرحله‌ای شرکت می‌کنند. مدل‌های Seq2Seq با تحلیل کل تاریخچه تعامل، قادرند رفتارهای غیرانسانی را شناسایی کنند — مانند ناسازگاری لحن، تغییر ناگهانی سبک، عدم انسجام منطقی یا پاسخ‌های تکراری و الگویی.

معماری Seq2Seq اغلب بر پایه شبکه‌های بازگشتی بلندمدت (LSTM) پیاده‌سازی می‌شود، که توانایی مدل‌سازی وابستگی‌های زمانی بلندمدت را دارند. در این راستا، مدل LSTM-LSTM که در این پژوهش استفاده شده، نمونه‌ای از این معماری است: یک LSTM به عنوان رمزگذار، کل مکالمه را تحلیل می‌کند و بازنمایی معنایی آن را استخراج می‌کند، و LSTM دوم به عنوان رمزگشا، این بازنمایی را برای شناسایی الگوهای غیرطبیعی بات پردازش می‌کند.

همچنین، ایده تحلیل توالی‌ها در Seq2Seq، زمینه را برای توسعه مدل‌های ترکیبی مانند RAG فراهم کرده است. در RAG، خروجی رمزگشا نه تنها بر اساس بازنمایی داخلی، بلکه با استفاده از دانش خارجی بازیابی‌شده شکل می‌گیرد — ویژگی‌ای که امکان شناسایی ناسازگاری‌های موضوعی در پیام‌های بات را فراهم می‌کند.

در نتیجه، مدل‌های Seq2Seq نه تنها ابزاری برای تولید متن هستند، بلکه چارچوبی قدرتمند برای تحلیل عمیق تعاملات و شناسایی بات‌های هوشمند محسوب می‌شوند — چارچوبی که دقیقاً در راستای رویکردهای پیشنهادی این پژوهش قرار دارد.

#### **6-2-2-2- شبکه‌های مولد متخاصم (GAN)**

شبکه‌های مولد متخاصم (Generative Adversarial Networks – GAN)، یکی از نخستین معماری‌های هوش مصنوعی بودند که با استفاده از رقابت دو شبکه عصبی — یک مولد و یک تمیزدهنده — تلاش می‌کردند داده‌هایی بسازند که از نظر آماری و ساختاری شبیه به داده‌های واقعی باشند. در حوزه پردازش زبان طبیعی، GANها برای تولید متن واقع‌نما، شبیه‌سازی مکالمات و ایجاد محتوای تقلیدی به کار گرفته شدند. هدف این بود که مولد بتواند پیام‌هایی تولید کند که تمیزدهنده نتواند آن‌ها را از محتوای انسانی تشخیص دهد.

اگرچه استفاده از GAN در تولید متن به دلیل چالش‌های مربوط به گسسته بودن خروجی و عدم پایداری آموزش با محدودیت مواجه شد، اما این مدل‌ها زمینه را برای ظهور نسل جدیدی از بات‌های هوشمند فراهم کردند — بات‌هایی که قادرند رفتار زبانی نزدیک به انسان را تقلید کنند. این تحول، شناسایی بات‌ها را از یک فرآیند تحلیل رفتاری ساده به یک چالش چندلایه تبدیل کرده است. امروزه اگرچه مدل‌های مولد بزرگ (LLMs) جایگزین اصلی GANها در تولید متن شده‌اند، اما مفهوم مولد متخاصم به‌عنوان یک گام بنیادین در تکامل بات‌های هوشمند، اهمیت تاریخی و مبانی‌ساز دارد.

#### **7-2-2-2- شبکه‌های حافظه کوتاه‌مدت بلند (LSTM)**

**تعریف کلی و اهمیت در پردازش توالی‌های زبانی**

شبکه‌های حافظه کوتاه‌مدت بلند (Long Short-Term Memory – LSTM) نوعی معماری پیشرفته از شبکه‌های عصبی بازگشتی (RNN) هستند که به‌منظور رفع محدودیت‌های RNN کلاسیک — به‌ویژه مشکل محو شدن گرادیان (Vanishing Gradient) — طراحی شده‌اند. این معماری با معرفی یک سلول حافظه و سیستم دروازه‌های کنترلی، قادر است وابستگی‌های بلندمدت در داده‌های ترتیبی را به‌صورت پایدار و کارآمد یاد بگیرد. برخلاف RNNهای ساده که در توالی‌های طولانی اطلاعات گذشته را از دست می‌دهند، LSTM می‌تواند اطلاعات مهم را در طول زمان حفظ کند و تنها اطلاعات نامرتبط را فراموش کند.

در حوزه پردازش زبان طبیعی، این توانایی آن را به یکی از اجزای کلیدی مدل‌های تولید متن، ترجمه ماشینی و تحلیل مکالمات تبدیل کرده است. در شناسایی بات‌های اجتماعی، اهمیت LSTM در این است که بات‌های پیشرفته اغلب با تغییر تدریجی سبک، لحن یا الگوی پاسخ‌دهی، سعی در تقلید از رفتار انسانی دارند. LSTM با تحلیل کل تاریخچه مکالمه، قادر است این تغییرات ظریف و غیرطبیعی را شناسایی کند — حتی زمانی که در یک پیام منفرد قابل رصد نباشند.

**تاریخچه و تحول معماری LSTM**

معماری LSTM در سال ۱۹۹۷ توسط Hochreiter و Schmidhuber معرفی شد و به عنوان یک نقطه عطف در یادگیری عمیق توالی‌ها شناخته می‌شود. این مدل در پاسخ به مشکل محو شدن گرادیان در RNNهای کلاسیک طراحی شد — مشکلی که در آن شبکه‌ها نمی‌توانستند ارتباط بین عناصر دور از هم در یک دنباله را یاد بگیرند.

اگرچه در ابتدا به دلیل محدودیت‌های سخت‌افزاری و کمبود داده، پذیرش گسترده‌ای نیافت، اما با گسترش منابع محاسباتی و ظهور داده‌های عظیم در دهه ۲۰۱۰، LSTMها در خط مقدم پیشرفت‌هایی مانند تشخیص گفتار Deep Speech و مدل‌های ترجمه ماشینی عصبی قرار گرفتند. این مدل‌ها به‌عنوان ستون فقرات معماری‌های Seq2Seq عمل کردند و زمینه را برای ظهور مدل‌های مولد و گفتگومحور فراهم کردند.

اگرچه امروزه در برخی حوزه‌ها توسط مدل‌های مبتنی بر توجه (ترنسفورمرها) جایگزین شده‌اند، اما LSTMها همچنان در کاربردهایی که نیاز به پردازش بلادرنگ، کم‌مصرف یا تحلیل مکالمات کوتاه‌مدت دارند، جایگاه قوی دارند — از جمله در سیستم‌های تشخیص بات.

**معماری داخلی: سلول حافظه و دروازه‌های کنترلی**

معماری LSTM حول یک سلول حافظه (Cell State) متمرکز است که به‌عنوان یک "نوار انتقال" اطلاعات مهم را در طول زمان منتقل می‌کند. این سلول توسط سه دروازه کنترل می‌شود:

* دروازه فراموشی (Forget Gate): تصمیم می‌گیرد چه بخش‌هایی از اطلاعات قبلی باید از حافظه حذف شوند.
* دروازه ورودی (Input Gate): تعیین می‌کند چه اطلاعات جدیدی باید به حافظه اضافه شود.
* دروازه خروجی (Output Gate): کنترل می‌کند که چه بخشی از حالت سلول به عنوان خروجی فعلی و ورودی گام بعدی منتقل شود.

این ساختار، امکان مدیریت هوشمندانه حافظه را فراهم می‌کند: مدل می‌تواند اطلاعات کلیدی (مثل موضوع مکالمه یا هویت گوینده) را در طول چندین پیام حفظ کند، در حالی که جزئیات غیرضروری را فراموش می‌کند.

نسخه‌های پیشرفته‌تر مانند Bi-LSTM با پردازش توالی در دو جهت (گذشته به آینده و آینده به گذشته)، اطلاعات زمینه‌ای غنی‌تری را استخراج می‌کنند — ویژگی‌ای مفید در تحلیل مکالمات پیچیده.

**کاربردهای LSTM در پردازش زبان و تشخیص بات**

LSTM در طی سال‌ها در حوزه‌های مختلفی مورد استفاده قرار گرفته است:

* پردازش زبان طبیعی: ترجمه ماشینی، خلاصه‌سازی متن، پاسخ‌گویی به پرسش.
* تشخیص گفتار: تبدیل سیگنال صوتی به متن با حفظ زمینه زمانی.
* پیش‌بینی سری‌های زمانی: مدل‌سازی روندهای مالی، ترافیک یا داده‌های حسگر.
* تحلیل ویدئو: شناسایی فعالیت‌ها در توالی فریم‌ها.

در حوزه شناسایی بات‌های اجتماعی، LSTM به‌عنوان یک ابزار کلیدی در تحلیل مکالمات چندمرحله‌ای استفاده می‌شود. این مدل می‌تواند:

* تغییرات غیرطبیعی در لحن یا سبک نگارش را شناسایی کند،
* پاسخ‌های تکراری یا الگویی را تشخیص دهد،
* و ناهنجاری‌هایی را کشف کند که در تحلیل تک‌پیامی نامرئی می‌مانند.

**نقش LSTM در مدل‌های گفتگومحور و معماری LSTM-LSTM**

در این پژوهش، از معماری LSTM-LSTM به‌عنوان یک مدل Seq2Seq برای تحلیل مکالمات استفاده شده است. در این ساختار:

* رمزگذار (Encoder): یک LSTM کل تاریخچه مکالمه را پردازش کرده و آن را به یک بازنمایی معنادار (context vector) فشرده می‌کند.
* رمزگشا (Decoder): LSTM دوم این بازنمایی را برای شناسایی الگوهای غیرانسانی — مانند ناسازگاری معنایی، تکرار ساختاری یا پاسخ‌های غیرطبیعی — تحلیل می‌کند.

این معماری امکان تحلیل عمیق تعاملات را فراهم می‌کند و به سیستم اجازه می‌دهد نه فقط محتوای هر پیام، بلکه پویایی کل مکالمه را ارزیابی کند. در نتیجه، LSTM-LSTM نه تنها یک ابزار تولید متن، بلکه یک چارچوب تحلیلی قدرتمند برای شناسایی بات‌های هوشمند محسوب می‌شود.

#### **8-2-2-2- بازیابی‑تولید افزوده (RAG) و ابزار Ollama**

چالش اصلی در شناسایی بات‌های پیشرفته، تشخیص تفاوت‌های ظریف و معنایی بین رفتار زبانی انسان و ماشین است. بات‌های مدرن اغلب محتوایی طبیعی و بدون خطای دستوری تولید می‌کنند، اما در سازگاری موضوعی، انسجام منطقی یا دقت اطلاعاتی دچار ناهنجاری می‌شوند. برای شناسایی این ناهنجاری‌ها، روش‌های سنتی مبتنی بر ویژگی‌های رفتاری یا متنی کافی نیستند. در اینجا، چارچوب بازیابی-تولید افزوده (Retrieval-Augmented Generation – RAG) به عنوان یک رویکرد نوین مطرح می‌شود.

RAG با ترکیب دو مؤلفه کلیدی — بازیابی اطلاعات از منابع خارجی و تولید متن توسط مدل زبانی — امکان تحلیل عمیق محتوا را فراهم می‌کند. در این معماری، پیش از تولید پاسخ، سیستم بر اساس ورودی (مثلاً یک مکالمه) در یک پایگاه دانش (مانند دیتاست مکالمات واقعی یا منابع معتبر) جستجو می‌کند و محتوای مرتبط را بازیابی می‌کند. سپس، مدل زبانی این اطلاعات را به‌عنوان زمینه (context) دریافت کرده و بر اساس آن، تحلیلی دقیق از ورودی انجام می‌دهد. این فرآیند، توهم‌زایی (Hallucination) مدل را کاهش می‌دهد و امکان تولید پاسخ‌های مستند و مبتنی بر واقعیت را فراهم می‌کند.

در حوزه شناسایی بات، RAG می‌تواند به دو شکل مؤثر عمل کند:

* تشخیص ناسازگاری موضوعی: با بازیابی محتوای واقعی در موضوع مکالمه، RAG قادر است ادعاهای نادرست، تناقض‌های منطقی یا استفاده از اطلاعات قدیمی یا جعلی را شناسایی کند — علائمی که اغلب در فعالیت بات‌های هوشمند دیده می‌شوند.
* تحلیل سبک تعامل: با مقایسه سبک نگارش، لحن و ساختار پاسخ‌ها با نمونه‌های واقعی از کاربران انسانی، RAG می‌تواند انحرافات ظریف در رفتار زبانی را آشکار کند.

این چارچوب، به‌ویژه در سناریوهایی که بات‌ها سعی می‌کنند با تقلید سطحی از زبان طبیعی، از سیستم‌های تشخیص ساده عبور کنند، اهمیت ویژه‌ای دارد. در این پژوهش، RAG به‌عنوان یک سیستم تحلیل معنایی ترکیبی استفاده شده است که نه تنها محتوای پیام را بررسی می‌کند، بلکه آن را در زمینه دانش واقعی قرار داده و ناهنجاری‌های آن را شناسایی می‌کند.

در نتیجه، RAG نه یک ابزار تولید متن ساده، بلکه یک چارچوب هوشمندانه برای تحلیل واقع‌گرایانه محتوا محسوب می‌شود — چارچوبی که دقیقاً در راستای نیاز به شناسایی بات‌های هوشمند و تقلب‌گر قرار دارد.

## **پژوهش‌های پیشین**

**مقدمه**

شناسایی بات‌های اجتماعی به‌عنوان یکی از چالش‌های محوری در فضای دیجیتال، طی سال‌های اخیر مورد توجه گسترده پژوهشگران در حوزه‌های هوش مصنوعی، امنیت سایبری و علوم شبکه قرار گرفته است. با توجه به تکامل فناوری و توانایی بات‌های پیشرفته در تقلید از رفتار انسان، رویکردهای مختلفی برای شناسایی آن‌ها پیشنهاد شده‌اند. در این بخش، به مرور نظام‌مند پژوهش‌های پیشین پرداخته می‌شود، به‌گونه‌ای که ابتدا مطالعاتی که به ماهیت بات‌ها و معیارهای تشخیصی آن‌ها پرداخته‌اند، مورد بررسی قرار می‌گیرند. سپس، تحقیقات مبتنی بر یادگیری ماشین، شبکه‌های عصبی، مدل‌های بازگشتی مانند LSTM، شبکه‌های مولد متخاصم (GAN)، پردازش زبان طبیعی (NLP) و مدل‌های گفتگومحور مرور خواهند شد. در پایان، نشان داده می‌شود که استفاده از مدل‌های زبانی بزرگ (LLM) و چارچوب‌های ترکیبی مانند RAG در شناسایی بات‌ها با تمرکز بر تحلیل عمیق متن، هنوز به‌طور سیستماتیک مورد بررسی قرار نگرفته است — خلأیی که پژوهش حاضر در پی پر کردن آن است

### مرور پژوهش‌های کلیدی در زمینه شناسایی بات‌های اجتماعی

در سال‌های اخیر، پژوهش‌های متعددی به بررسی پدیده بات‌های اجتماعی و روش‌های شناسایی آن‌ها پرداخته‌اند. این مطالعات نه تنها ماهیت بات‌ها را تحلیل کرده‌اند، بلکه چارچوب‌هایی برای تشخیص و مقابله با آن‌ها ارائه داده‌اند. در ادامه، به مرور برخی از پژوهش‌های تأثیرگذار در این حوزه پرداخته می‌شود.

در مطالعه Ferrara و همکاران (2016) با عنوان The Rise of Social Bots، تلاش شده است تا تعریفی جامع از «بات اجتماعی» ارائه شود و انواع آن بر اساس رفتار و هدف دسته‌بندی گردد. این پژوهش بات‌ها را نه تنها به عنوان ابزارهای تبلیغاتی، بلکه به‌عنوان عواملی فعال در دستکاری افکار عمومی و انتشار اخبار جعلی معرفی می‌کند. نویسندگان چارچوبی چندبُعدی پیشنهاد می‌دهند که در آن ویژگی‌های زمانی، شبکه‌ای و محتوایی به‌صورت ترکیبی برای شناسایی بات مورد استفاده قرار می‌گیرند. این مقاله به‌عنوان یکی از اولین مطالعات جامع، زمینه را برای درک چندوجهی پدیده بات فراهم کرد.

در ادامه، Cresci و همکاران (2017) در مقاله The Paradigm-Shift of Social Spambots تحول چشمگیری در رفتار بات‌ها را تحلیل می‌کنند. آن‌ها نشان می‌دهند که بات‌های نسل جدید با استفاده از یادگیری ماشین و شبیه‌سازی رفتار انسانی، قادرند از سیستم‌های تشخیص مبتنی بر قواعد ثابت و ویژگی‌های سطحی عبور کنند. این مطالعه بر لزوم استفاده از روش‌های یادگیری عمیق و تحلیل رفتار پویا تأکید دارد و نشان می‌دهد که شناسایی بات دیگر یک مسئله سطحی نیست، بلکه نیازمند مدل‌های هوشمند و تطبیق‌پذیر است.

یکی دیگر از مطالعات محوری، گزارش Subrahmanian و همکاران (2016) از مسابقه DARPA Twitter Bot Challenge است که به ارزیابی عملکرد الگوریتم‌های مختلف شناسایی بات پرداخت. این پژوهش نشان می‌دهد که ترکیب داده‌های پروفایل کاربر، محتوای انتشارشده و الگوهای تعامل می‌تواند دقت تشخیص را به‌طور چشمگیری افزایش دهد. همچنین، معیارهای ارزیابی دقیق و استانداردی برای مقایسه روش‌های مختلف ارائه شده است که بعدها به عنوان مبنایی برای توسعه سیستم‌های جدید مورد استفاده قرار گرفت.

Varol و همکاران (2017) در مقاله Online Human-Bot Interactions به بررسی تعاملات بین کاربران انسانی و بات‌ها پرداخته‌اند. آن‌ها یک مدل آماری توسعه داده‌اند که احتمال بات بودن یک حساب را بر اساس ویژگی‌هایی مانند فرکانس ارسال پست، پیچیدگی زبانی متن و ساختار شبکه دنبال‌کنندگان تخمین می‌زند. این مطالعه همچنین نشان می‌دهد که بات‌ها نه تنها فعالیت می‌کنند، بلکه با کاربران واقعی تعامل دارند و در شکل‌گیری دینامیک شبکه نقش دارند.

در نهایت، Stella و همکاران (2018) در مقاله Bots Increase Exposure to Negative and Inflammatory Content به تأثیر بات‌ها بر کیفیت محتوای اجتماعی پرداخته‌اند. آن‌ها نشان می‌دهند که بات‌ها به‌طور سیستماتیک در انتشار محتوای منفی، تحریک‌آمیز و ایدئولوژیک نقش دارند و با افزایش مواجهه کاربران با چنین مطالبی، به شکل‌گیری «اتاق‌های پژواک» و قطبی‌سازی اجتماعی کمک می‌کنند. این مطالعه بر اهمیت شناسایی بات‌ها نه تنها از نظر فنی، بلکه از منظر سلامت اجتماعی و دموکراسی تأکید دارد.

### مطالعات مرتبط با ویژگی‌ها و شاخص‌های شناسایی بات‌های اجتماعی

درک عمیق از ویژگی‌های تمایزدهنده بات‌ها و کاربران انسانی، پایه‌ای ضروری برای توسعه سیستم‌های تشخیص دقیق است. پژوهش‌های متعددی به شناسایی، تعریف و ارزیابی شاخص‌های کمی و کیفی در حوزه‌های مختلف — از جمله محتوایی، رفتاری، زمانی، ساختار شبکه‌ای و تعاملی — پرداخته‌اند. این مطالعات نشان می‌دهند که هیچ شاخص واحدی نمی‌تواند به تنهایی بات را تشخیص دهد، بلکه ترکیب چندبُعدی از ویژگی‌ها ضروری است.

Varol و همکاران (2017) در مطالعه Online Human-Bot Interactions چارچوبی جامع از ویژگی‌های مؤثر ارائه می‌دهند که در سه دسته اصلی قرار می‌گیرند: محتوایی (مانند n-گرام، پیوستگی معنایی و احساسات)، زمانی (فاصله و ریتم ارسال پست)، و شبکه‌ای (درجه گره، مرکزیت و ضریب خوشه‌بندی). آن‌ها همچنین نشان می‌دهند که شدت فعالیت بالا همراه با الگوهای تعامل غیرطبیعی، ارتباط قوی با احتمال بات بودن دارد. این مطالعه زمینه را برای استفاده از ویژگی‌های چندوجهی در مدل‌های تشخیص فراهم کرد.

در همین راستا، Ferrara و همکاران (2016) در مقاله The Rise of Social Bots به تحلیل ویژگی‌های معماری و کارکردی بات‌ها پرداخته‌اند. آن‌ها مشخصات پروفایل، تنوع زبانی، فراوانی هشتگ‌ها و ساختار شبکه دنبال‌کنندگان را به عنوان شاخص‌های کلیدی معرفی می‌کنند. یکی از شاخص‌های مهم این پژوهش، "ثبات محتوا" (Consistency of Content) است که بات‌ها را به دلیل انتشار محتوای یکنواخت و تکراری از کاربران انسانی متمایز می‌کند.

Subrahmanian و همکاران (2016) در گزارش DARPA Twitter Bot Challenge، شاخص‌های متنوعی را ارزیابی کرده و چارچوبی چندبُعدی برای امتیازدهی به حساب‌ها ارائه می‌دهند. این چارچوب شامل ویژگی‌هایی مانند فراوانی واژگان، تعادل دنبال‌کنندگان/دنبال‌کنندگان، توزیع تعاملات و جهت‌گیری احساسی است. ترکیب این شاخص‌های ایستا و پویا، دقت تشخیص را به‌طور چشمگیری افزایش می‌دهد و نشان می‌دهد که تحلیل رفتار در طول زمان ضروری است.

در مطالعه Kudugunta و Ferrara (2018) با عنوان Deep Neural Networks for Bot Detection، تمرکز بر نقش ویژگی‌های متنی و متادیتا قرار گرفته است. این پژوهش نشان می‌دهد که طول توییت، پیچیدگی واژگان و تغییرات عکس پروفایل می‌توانند نشانه‌های معناداری باشند. همچنین، شاخص "تنوع استفاده از کلمات" (Word Usage Diversity) به عنوان یک متغیر کلیدی در تمایز بات‌ها معرفی شده است — بات‌ها معمولاً از واژگان محدودی استفاده می‌کنند، در حالی که کاربران انسانی تنوع بیشتری دارند.

همچنین، Cresci و همکاران (2015) در مقاله Fame for Sale به شناسایی فالوئرهای جعلی پرداخته‌اند و شاخص‌هایی مانند تعادل تعامل و تکرار محتوا را تعریف کرده‌اند. یکی از شاخص‌های کلیدی آن‌ها، "نرخ تکرار محتوا" (Content Duplication Rate) است که نشان می‌دهد حساب‌های خودکار چقدر از محتوای یکسان در فواصل زمانی کوتاه استفاده می‌کنند — ویژگی‌ای که در بات‌های انبوه بسیار شایع است.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر یادگیری ماشین

در سال‌های اولیه تحقیقات در حوزه شناسایی بات، تمرکز اصلی بر استفاده از الگوریتم‌های یادگیری ماشین کلاسیک و تحلیل داده‌های متنی بود. این رویکردها با استخراج ویژگی‌های زبانی، معنایی و رفتاری از محتوای پست‌ها و پیام‌ها، سعی در تمایز بین حساب‌های انسانی و خودکار داشتند. فرآیند کلی شامل سه مرحله است: استخراج ویژگی از متن و متادیتا، ساخت مدل طبقه‌بندی و ارزیابی عملکرد با استفاده از داده‌های برچسب‌خورده.

Lee و همکاران (2011) در مطالعه Seven Months with the Devils یکی از اولین تلاش‌های سیستماتیک برای شناسایی حساب‌های آلاینده محتوایی (content polluters) در توییتر را انجام دادند. آن‌ها با تحلیل متن توییت‌ها، ویژگی‌هایی مانند n-گرام‌های کلمه و کاراکتر، توزیع فراوانی واژگان و استفاده از علائم نگارشی را استخراج کردند و این ویژگی‌ها را همراه با الگوهای زمانی فعالیت به مدل‌های یادگیری ماشین مانند SVM و Naïve Bayes دادند. این پژوهش نشان داد که ترکیب ویژگی‌های متنی و زمانی می‌تواند دقت تشخیص را بهبود بخشد. داده‌های این مطالعه از داده‌های عمومی توییتر (از طریق API) جمع‌آوری شده بودند و شامل مجموعه‌ای از حساب‌های مشکوک و تأییدشده بود.

در ادامه، Morstatter و همکاران (2016) در مقاله Detecting Automated Accounts in the Twitter Speech on COVID-19 به بررسی بات‌ها در زمینه اخبار جهانی پرداختند. آن‌ها ویژگی‌هایی مانند طول جملات، تکرار کلمات، پیچیدگی واژگان و حضور لینک را استخراج کردند و از Random Forest و Logistic Regression برای طبقه‌بندی استفاده کردند. با استفاده از روش انتخاب ویژگی مبتنی بر اطلاعات متقابل (Mutual Information)، ویژگی‌های بیش‌ازحد نویزی حذف شدند. این مطالعه بر اهمیت تحلیل محتوای متنی در سناریوهای بحرانی مانند همه‌گیری تأکید داشت. داده‌ها از توییت‌های مرتبط با کووید-۱۹ جمع‌آوری شده بودند که بعدها به عنوان یک دیتاست مرجع برای تحقیقات مشابه استفاده شد.

Kudugunta و Ferrara (2018) در مقاله Deep Neural Networks for Bot Detection به مقایسه عملکرد مدل‌های یادگیری ماشین سنتی (مانند SVM و Random Forest) با شبکه‌های عصبی عمیق پرداختند. آن‌ها ویژگی‌هایی مانند توصیف پروفایل، محتوای توییت‌ها و الگوهای استفاده از هشتگ را به عنوان ورودی مدل‌ها استفاده کردند. نتایج نشان داد که مدل‌های عمیق عملکرد بهتری در شناسایی بات‌های پیچیده دارند، به‌ویژه زمانی که ویژگی‌های متنی و رفتاری ترکیب می‌شوند. این پژوهش از مجموعه داده‌ای چندپلتفرمی شامل حساب‌های توییتر و اینستاگرام استفاده کرد که بعدها به عنوان Botometer-Lite Dataset شناخته شد.

Dickerson و همکاران (2014) در مقاله Using Sentiment to Detect Bots on Twitter به این سؤال پرداختند که آیا تحلیل احساسات متن می‌تواند در شناسایی بات مؤثر باشد؟ آن‌ها نشان دادند که بات‌ها اغلب از الگوهای احساسی غیرطبیعی (مانند احساسات بیش‌ازحد مثبت یا منفی بدون تنوع) استفاده می‌کنند. این ویژگی‌ها همراه با ساختار شبکه تعاملی به مدل SVM داده شدند و دقت تشخیص بهبود یافت. داده‌های این مطالعه از توییت‌های مرتبط با رویدادهای سیاسی و اجتماعی جمع‌آوری شده بودند.

در نهایت، Varol و همکاران (2017) در مقاله Online Human-Bot Interactions از ترکیب ویژگی‌های محتوایی (n-گرام، هشتگ، لینک)، زمانی و شبکه‌ای برای تشخیص چندکلاسه بات‌ها استفاده کردند. آن‌ها از الگوریتم Random Forest بهره بردند که به دلیل توانایی در مدیریت ویژگی‌های ناهمبست و ارزیابی اهمیت هر ویژگی، برای این کار مناسب بود. این مطالعه از دیتاستی شامل ۳.۷ میلیون توییت استفاده کرد که بعدها به عنوان بخشی از Botometer API مورد استفاده قرار گرفت.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر شبکه‌های عصبی مصنوعی (ANN)

با پیشرفت یادگیری عمیق، پژوهش‌های زیادی به سمت استفاده از شبکه‌های عصبی مصنوعی (Artificial Neural Networks – ANN) برای شناسایی بات‌های اجتماعی حرکت کردند. در این رویکردها، به جای استخراج دستی ویژگی‌ها، از مدل‌هایی استفاده می‌شود که قادرند بازنمایی‌های غنی و خودکار از داده‌های متنی یاد بگیرند. این توانایی، به‌ویژه در تحلیل محتوای پیچیده و نویزدار شبکه‌های اجتماعی، دقت تشخیص را بهبود می‌بخشد.

Kudugunta و Ferrara (2018) در مقاله Deep Neural Networks for Bot Detection یکی از اولین مطالعات جامع در استفاده از شبکه‌های عصبی پیش‌خور (MLP) برای شناسایی بات بودند. آن‌ها از داده‌های متنی پروفایل‌ها و توییت‌ها به‌عنوان ورودی استفاده کردند و با ترکیب ویژگی‌هایی مانند n-گرام، طول متن و تنوع واژگان، مدل MLP را آموزش دادند. نتایج نشان داد که این رویکرد نسبت به الگوریتم‌های یادگیری ماشین کلاسیک دقت بهتری دارد. این پژوهش از Botometer-Lite Dataset استفاده کرد که شامل حساب‌های برچسب‌خورده از توییتر و اینستاگرام است.

در همین راستا، Hayawi و همکاران (2022) در مقاله Deep Learning for Twitter Bot Detection نشان دادند که می‌توان مستقیماً از متن خام توییت‌ها به عنوان ورودی مدل استفاده کرد. آن‌ها با به‌کارگیری تعبیه‌سازی‌های واژگان (Word Embeddings) و یک شبکه عصبی fully-connected، نیاز به مهندسی دستی ویژگی را کاهش دادند. این رویکرد توانست با پردازش مستقیم متن، الگوهای زبانی غیرطبیعی بات‌ها را بدون وابستگی به ویژگی‌های ساختاری شناسایی کند. داده‌های این مطالعه از توییت‌های عمومی در موضوعات سیاسی و اجتماعی جمع‌آوری شده بود.

Dhanush و Bhuvana (2021) در مطالعه Fake Account Detection in Twitter Using ANN Models از یک شبکه عصبی سه‌لایه برای پیش‌بینی نوع حساب استفاده کردند. آن‌ها توالی‌های توییت را به embedding تبدیل کردند و عملکرد ترکیب‌های مختلف — مانند TF-IDF + ANN و Bag-of-Words + ANN — را مقایسه کردند. نتایج نشان داد که ترکیب embeddingهای مبتنی بر TF-IDF با شبکه عصبی، بهترین عملکرد را دارد. این مطالعه بر اهمیت نحوه نمایش متن به عنوان ورودی شبکه تأکید داشت.

در نهایت، Kudugunta و همکاران (2019) در مقاله Combining Recurrent and Convolutional Neural Networks for Bot Detection به ساخت یک مدل ترکیبی پرداختند. آن‌ها از CNN برای استخراج ویژگی‌های محلی (مانند الگوهای n-گرامی و ساختارهای نحوی) و از RNN برای مدل‌سازی وابستگی زمانی بین توییت‌ها استفاده کردند. ورودی مدل شامل ترکیبی از متن و ویژگی‌های پروفایل بود. این معماری هیبریدی نسبت به مدل‌های تک‌نوع، دقت و قدرت تعمیم بالاتری در شناسایی بات‌های پیچیده نشان داد. این پژوهش نیز از Botometer Dataset بهره برد.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر LSTM

مدل‌های Long Short-Term Memory (LSTM) به دلیل توانایی برجسته در مدل‌سازی وابستگی‌های بلندمدت در داده‌های ترتیبی، به یکی از ستون‌های اصلی روش‌های مبتنی بر یادگیری عمیق در حوزه شناسایی بات‌های اجتماعی تبدیل شده‌اند. این مدل‌ها به‌ویژه در تحلیل محتوای متنی کوتاه، نویزدار و پویای شبکه‌های اجتماعی عملکرد قابل‌توجهی از خود نشان داده‌اند، زیرا قادرند الگوهای زبانی، سبکی و زمانی را در طول یک مکالمه یا دنباله پست‌ها مدل کنند.

Wei و همکاران (2019) در مقاله Bot Detection on Social Media Using Bidirectional LSTMs with Word Embeddings از ترکیب embeddingهای Word2Vec و شبکه Bi-LSTM برای شناسایی بات استفاده کردند. با تبدیل متن توییت‌ها به بردارهای معنایی و پردازش آن‌ها در دو جهت، مدل قادر شد روابط معنایی دوطرفه را یاد بگیرد. نتایج نشان داد که این رویکرد دقت بالاتری نسبت به مدل‌های یک‌جهته و روش‌های یادگیری ماشین سنتی دارد، به‌ویژه در داده‌های نویزدار و متن‌های کوتاه.

Kudugunta و Ferrara (2018) در مطالعه Deep Neural Networks for Bot Detection از ترکیب embeddingهای متنی و یک لایه LSTM برای مدل‌سازی توالی توییت‌ها استفاده کردند و خروجی آن را به یک لایه Fully-Connected برای طبقه‌بندی منتقل کردند. این رویکرد نه تنها نیاز به مهندسی دستی ویژگی را کاهش داد، بلکه امکان شناسایی الگوهای رفتاری زنجیره‌ای (مانند تکرار محتوا در چند پیام متوالی) را فراهم کرد. این مدل بر روی Botometer-Lite Dataset آموزش دید و دقتی حدود 91% گزارش شد.

Feng و همکاران (2021) در مقاله Attention-Based LSTM for Twitter Bot Detection به بهبود مدل LSTM با استفاده از مکانیزم توجه (Attention) پرداختند. آن‌ها نشان دادند که با اعمال توجه بر خروجی‌های LSTM، مدل می‌تواند بر کلمات و عبارات کلیدی که نشانه بات بودن هستند (مانند تکرار هشتگ یا لینک) تمرکز کند. این ترکیب، دقت تشخیص را نسبت به مدل LSTM پایه بهبود بخشید و دقتی بالای 93% در داده‌های چندکلاسه گزارش شد.

Rahmani و Alipour (2020) در پژوهش Character-Level LSTM for Bot Detection in Twitter از یک مدل Bi-LSTM در سطح کاراکتر استفاده کردند. این رویکرد به جای تحلیل کلمات، توالی کاراکترهای توییت را پردازش می‌کند و قادر است تفاوت‌های ظریف در سبک نگارش، مانند استفاده عمدی از نویسه‌های غیرمعمول یا خطاهای تایپی سیستماتیک بات‌ها را شناسایی کند. این مدل در برابر تغییرات عمدی سبک نگارش مقاوم بود و دقت قابل‌قبولی در شناسایی بات‌های تقلیدگر داشت.

در مطالعه Kudugunta و همکاران (2019) با عنوان Combining Recurrent and Convolutional Neural Networks for Bot Detection، از ترکیب CNN برای استخراج ویژگی‌های محلی (مانند n-گرام‌های مهم) و LSTM برای مدل‌سازی وابستگی بلندمدت استفاده شد. این معماری هیبریدی قادر بود هم الگوهای ساختاری و هم روابط معنایی در توالی متنی را یاد بگیرد. این مدل بر روی داده‌های ترکیبی از متن و ویژگی‌های پروفایل آموزش دید و دقتی بالای 92% گزارش کرد.

همچنین، Wang (2018) در مقاله Sequence-Based Bot Detection Using Deep Recurrent Networks از یک معماری Seq2Seq مبتنی بر LSTM-LSTM برای تحلیل مکالمات چندمرحله‌ای استفاده کرد. در این رویکرد، رمزگذار (Encoder) کل تاریخچه مکالمه را به یک بردار مفهومی فشرده می‌کند و رمزگشا (Decoder) این بازنمایی را برای تولید برچسب یا تشخیص ناهنجاری استفاده می‌کند. این مدل به‌طور خاص برای شناسایی بات‌هایی طراحی شد که سعی می‌کنند با تغییر تدریجی سبک، رفتار انسانی را تقلید کنند. نتایج نشان داد که این مدل دقتی حدود 94% در تشخیص بات‌های هوشمند دارد.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های زبانی

با گسترش استفاده از زبان طبیعی در فضای اجتماعی، مدل‌های زبانی (Language Models – LM) به عنوان ابزاری کلیدی برای تحلیل محتوای متنی بات‌ها مطرح شدند. این مدل‌ها با یادگیری احتمال وقوع توالی‌های واژگانی، قادرند الگوهای زبانی غیرطبیعی، تکراری یا غیرانسانی را شناسایی کنند. در ابتدا، مدل‌های زبانی مبتنی بر آمار (مانند n-gram) مورد استفاده قرار می‌گرفتند، اما با پیشرفت یادگیری عمیق، مدل‌های عصبی زبانی (Neural Language Models) جایگزین آن‌ها شدند — مدل‌هایی که نه تنها ساختار سطحی متن، بلکه وابستگی‌های معنایی و زمینه‌ای را نیز مدل می‌کنند.

یکی از اولین کاربردهای موفق مدل‌های زبانی در شناسایی بات، توسط Kudugunta و Ferrara (2018) در مقاله Deep Neural Networks for Bot Detection ارائه شد. آن‌ها از embeddingهای Word2Vec به عنوان نمایش معنایی کلمات استفاده کردند و آن‌ها را به یک شبکه عصبی بازگشتی (LSTM) منتقل کردند تا مدل بتواند توالی‌های متنی را درک کند. این ترکیب، امکان یادگیری الگوهای نحوی و معنایی پیچیده را فراهم کرد و دقت شناسایی بات را به‌ویژه در داده‌های نویزدار بهبود بخشید. این پژوهش نشان داد که استفاده از بازنمایی‌های غنی از متن، نسبت به ویژگی‌های دستی، عملکرد بهتری دارد.

Feng و همکاران (2021) در مقاله Attention-Based LSTM for Twitter Bot Detection با افزودن مکانیزم توجه (Attention) به مدل زبانی، توانستند دقت را افزایش دهند. این مکانیزم به مدل اجازه می‌دهد بر توالی‌های کلیدی در متن — مانند تکرار هشتگ، لینک یا عبارات تبلیغاتی — تمرکز کند. این رویکرد به‌ویژه در شناسایی بات‌هایی که از الگوهای تکراری استفاده می‌کنند، بسیار مؤثر بود و دقت شناسایی را به بیش از 93% رساند.

همچنین، Hayawi و همکاران (2022) در مطالعه Deep Learning for Twitter Bot Detection به بررسی عملکرد مدل‌های زبانی در محیط‌های چندزبانه پرداختند. آن‌ها نشان دادند که با استفاده از embeddingهای چندزبانه و شبکه‌های عمقی، می‌توان الگوهای مشترک در رفتار زبانی بات‌ها را در زبان‌های مختلف (مانند انگلیسی، عربی و فرانسوی) شناسایی کرد. این یافته‌ها اهمیت مدل‌های زبانی را در تشخیص بات‌های بین‌فرهنگی نشان داد.

در نهایت، Wang و Mao (2019) در مقاله A Neural Language Model Approach to Bot Detection in Social Media به طور خاص بر استفاده از مدل‌های زبانی عصبی (RNN-LM) برای محاسبه احتمال وقوع توالی متنی توسط یک بات تمرکز کردند. آن‌ها احتمال log برای هر توییت را محاسبه کردند و نشان دادند که بات‌ها اغلب توالی‌هایی با احتمال پایین تولید می‌کنند — زیرا محتوایشان از ترکیب‌های غیرطبیعی یا تصادفی تشکیل شده است. این رویکرد بدون نیاز به برچسب‌گذاری کامل، قادر به شناسایی بات‌های غیرعادی بود.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های توالی به توالی

با افزایش پیچیدگی بات‌های اجتماعی در ایجاد مکالمات چندمرحله‌ای، نیاز به مدل‌هایی که بتوانند کل تاریخچه تعامل را تحلیل کنند، امری ضروری شد. در این راستا، مدل‌های توالی به توالی (Sequence-to-Sequence – Seq2Seq) به عنوان یکی از معماری‌های کلیدی در تحلیل محتوای متنی بات مطرح شدند. این مدل‌ها با استفاده از ساختار رمزگذار–رمزگشا (Encoder-Decoder)، قادرند توالی ورودی (مانند یک مکالمه) را به یک بازنمایی مفهومی فشرده تبدیل کرده و سپس خروجی (مانند برچسب رفتاری یا پیش‌بینی پاسخ) تولید کنند.

Kudugunta و Ferrara (2018) در مطالعه Deep Neural Networks for Bot Detection از یک معماری LSTM-Seq2Seq برای مدل‌سازی رفتار زمانی و متنی کاربران توییتر استفاده کردند. این مدل با تحلیل توالی توییت‌ها، قادر بود الگوهای رفتاری زنجیره‌ای بات‌ها را شناسایی کند و دقت تشخیص را نسبت به روش‌های یادگیری ماشین کلاسیک بهبود بخشد.

Feng و همکاران (2021) در مقاله Attention-Based LSTM for Twitter Bot Detection به توسعه یک معماری Encoder-Decoder مبتنی بر Bi-LSTM پرداختند که با افزودن مکانیزم توجه (Attention)، توانست بر بخش‌های کلیدی متن — مانند تکرار هشتگ یا عبارات تبلیغاتی — تمرکز کند. این ترکیب، دقت شناسایی بات را در داده‌های نویزدار و کوتاه افزایش داد.

Rahmani و Alipour (2020) نیز در پژوهش Sequence-Based Bot Detection Using Deep Recurrent Networks از ترکیب Seq2Seq و توجه برای پردازش متون چندجمله‌ای استفاده کردند. این مدل به‌ویژه در کاهش نرخ مثبت کاذب (False Positive Rate) در دیتاست‌های نامتوازن عملکرد بهتری داشت، زیرا قادر بود بافت کلی مکالمه را درک کند.

در مطالعه Yang و Zhang (2023)، معماری Seq2Seq با جایگزینی بلوک‌های LSTM با ترنسفورمرها به‌روزرسانی شد. این تکامل، امکان مدل‌سازی وابستگی‌های بلندمدت و مقابله با تغییرات پویا در زبان و استراتژی‌های فرار بات‌ها را فراهم کرد.

در نهایت، Najari و همکاران (2024) در مقاله Adversarial Dialogue Modeling for Bot Detection از مدل‌های Seq2Seq محاوره‌ای برای شناسایی بات‌هایی استفاده کردند که در مکالمات واقع‌نما شرکت می‌کنند. آن‌ها از این مدل‌ها نه تنها برای تحلیل، بلکه برای تولید نمونه‌های خصمانه به منظور تقویت سیستم شناسایی استفاده کردند — نشان‌دهنده قابلیت دوگانه این معماری در نقش مهاجم و مدافع.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر پردازش زبان طبیعی

با گسترش فعالیت بات‌های هوشمند در فضای اجتماعی، تحلیل سطحی محتوا دیگر کافی نیست. در این شرایط، پردازش زبان طبیعی (Natural Language Processing – NLP) به عنوان یکی از ابزارهای کلیدی برای شناسایی بات مطرح شده است. برخلاف روش‌های مبتنی بر ویژگی‌های سطحی، NLP امکان تحلیل معنایی، سبکی و گفتمانی متن را فراهم می‌کند — توانایی‌هایی که برای تشخیص بات‌هایی که سعی در تقلید از رفتار انسانی دارند، حیاتی است.

در این راستا، Sant et al. (2020) در مقاله Stylometric Analysis for Bot Detection in Social Media به بررسی سبک‌نگاری (Stylomسetry) حساب‌های کاربری پرداختند. آن‌ها از ویژگی‌هایی مانند طول جملات، تنوع واژگان، استفاده از قیدها و حروف اضافه، و فراوانی ضمایر شخصی استفاده کردند. نتایج نشان داد که بات‌ها اغلب از سبک نگارش یکنواخت و مکانیکی استفاده می‌کنند، در حالی که کاربران انسانی تنوع بیشتری در سبک نوشتار دارند. این رویکرد بدون نیاز به تحلیل محتوای معنایی، قادر به شناسایی بات‌های پیشرفته بود.

در مطالعه‌ای دیگر، Zannettou et al. (2018) در On the Origins of Memes by Analyzing Social News Sites به تحلیل محتوای چندوجهی (متن، هشتگ، ایموجی) پرداختند. آن‌ها نشان دادند که بات‌ها در انتشار میم‌ها و محتوای گسترش‌پذیر، الگوهایی غیرانسانی و سیستماتیک دارند. با استفاده از تکنیک‌های NLP مانند تحلیل احساسات، تشخیص موضوع (Topic Modeling) و شبکه‌های معنایی، توانستند محتوای بات‌ها را از نظر معنایی از کاربران انسانی تمایز دهند.

همچنین، Bian et al. (2021) در مقاله Detecting Social Bots with Semantic Inconsistency Analysis به بررسی ناسازگاری معنایی در مکالمات بات پرداختند. آن‌ها نشان دادند که بات‌ها اغلب در پاسخ به پرسش‌ها، به موضوع فاصله می‌گیرند یا ادعاهای متضاد مطرح می‌کنند. با استفاده از مدل‌های تعبیه‌سازی مفهومی (Conceptual Embeddings) و محاسبه همبستگی معنایی بین پیام و پاسخ، این ناهنجاری‌ها را شناسایی کردند. این رویکرد نشان داد که تحلیل معنایی عمیق، حتی در متن‌های ظاهراً طبیعی، می‌تواند بات را فاش کند.

در نهایت، Ribeiro et al. (2020) در مقاله Auditing Online Social Platforms for Bot Activity Using Linguistic Cues از تکنیک‌های NLP چندوجهی شامل تحلیل ساختار جمله، سبک نگارش، و تغییرات زبانی در طول زمان استفاده کردند. آن‌ها نشان دادند که بات‌ها در طول مکالمه، تغییرات ناگهانی در لحن، واژگان یا سطح رسمیت دارند — نشانه‌ای که در تحلیل تک‌پیامی قابل رصد نیست. این مطالعه بر اهمیت تحلیل دنباله‌ای و پویا در شناسایی بات تأکید داشت.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر مدل‌های گفتگو

با تکامل بات‌های اجتماعی از حالت تولید محتوای ایستا به فعالیت‌های پویا و مبتنی بر مکالمه، رویکردهای سنتی تحلیل متن منفرد دیگر کافی نبودند. در این شرایط، مدل‌های گفتگومحور (Dialogue Models) به عنوان یک تحول پارادایمی در شناسایی بات مطرح شدند. این مدل‌ها با تحلیل کل تاریخچه تعامل، قادرند رفتارهای غیرانسانی را در سطح دیالوگ، ساختار تعامل و پویایی مکالمه شناسایی کنند — نه فقط در سطح محتوای یک پیام.

Alothali و همکاران (2019) در مطالعه Characterizing Social Bots by Conversation Patterns به تحلیل ساختار و محتوای گفتگوهای توییتر پرداختند. آن‌ها از ترکیب بردارهای کلمه و ویژگی‌های تعاملی (مانند تعداد پاسخ‌ها، عمق مکالمه و تغییر موضوع) استفاده کردند. نتایج نشان داد که بات‌ها اغلب الگوهای تعاملی غیرطبیعی دارند — مثلاً پاسخ‌های بیش‌ازحد سریع یا عدم پیگیری موضوع — که این ویژگی‌ها می‌توانند به تنهایی یا در ترکیب با محتوا، دقت شناسایی را بهبود بخشند.

Beskow و Carley (2020) در مقاله Dialogue Structure as a Marker of Automation بر ساختار مکالمه تمرکز کردند. آن‌ها ویژگی‌هایی مانند عمق مکالمه، طول پاسخ، زمان پاسخ‌دهی و تغییر موضوع را استخراج کردند و نشان دادند که بات‌ها در مقایسه با کاربران انسانی، از الگوهای ساختاری محدود و قابل پیش‌بینی پیروی می‌کنند. این مطالعه اثبات کرد که تحلیل فرم گفتگو — نه فقط محتوا — می‌تواند نشانه‌ای قوی از فعالیت خودکار باشد.

Sayyadiharikandeh و همکاران (2020) در Detecting Emerging Bots with Conversation Embeddings از شبکه‌های عصبی عمیق و embeddingهای مکالمه برای مدل‌سازی روابط پیچیده بین پیام‌ها استفاده کردند. آن‌ها نشان دادند که بات‌های نوظهور که از نظر محتوایی شبیه انسان هستند، در تحلیل کل جریان مکالمه قابل شناسایی می‌شوند — زیرا نتوانسته‌اند الگوهای تعاملی پیچیده انسانی را تقلید کنند.

Cresci و همکاران (2020) در مقاله The Era of Dialogue-Based Bot Detection تحولی مهم را معرفی کردند: حرکت از تحلیل متن ایستا به تحلیل مکالمه پویا. آن‌ها نشان دادند که با تمرکز بر توالی دیالوگ‌ها و استفاده از رویکردهای context-aware، می‌توان بات‌هایی را شناسایی کرد که در تحلیل تک‌پیامی فرار می‌کنند. این پژوهش به عنوان یک نقطه عطف، زمینه را برای استفاده از مدل‌های Seq2Seq و گفتگومحور فراهم کرد.

در نهایت، Najari و همکاران (2024) در مقاله Adversarial Dialogue Modeling for Bot Detection به استفاده از سناریوهای رقابتی پرداختند. آن‌ها بات‌ها را به‌صورت عاملانی تعاملی و رقابتی مدل کردند و از یادگیری گفته‌محور برای ارزیابی استحکام سیستم‌های تشخیص استفاده کردند. این رویکرد نشان داد که سیستم‌های شناسایی باید نه تنها محتوا را تحلیل کنند، بلکه رفتار پویا و تطبیقی بات‌ها را نیز پیش‌بینی و مقابله کنند.

### پژوهش‌های مرتبط با شناسایی بات‌های اجتماعی مبتنی بر GAN

شبکه‌های مولد متخاصم (Generative Adversarial Networks – GAN) اگرچه به‌عنوان یک ابزار تولید محتوا معرفی شدند، اما در سال‌های اخیر به‌عنوان ابزاری قدرتمند در شبیه‌سازی بات‌های هوشمند و تقویت سیستم‌های شناسایی مورد توجه قرار گرفته‌اند. در حوزه شناسایی بات‌های اجتماعی، GANها عمدتاً به دو منظور استفاده می‌شوند: تولید نمونه‌های متنی واقع‌نما برای غنی‌سازی داده‌های آموزشی و شبیه‌سازی حملات تقابلی (adversarial attacks) برای ارزیابی و بهبود مقاومت مدل‌های تشخیص.

Wang و همکاران (2018) در مقاله Adversarial Learning for Social Bot Detection یک چارچوب GAN ارائه دادند که در آن مولد (Generator) پیام‌هایی شبیه به فعالیت بات تولید می‌کند و تمیزدهنده (Discriminator) آن‌ها را تشخیص می‌دهد. این فرآیند رقابتی، مدل شناسایی را در برابر نمونه‌های پیچیده و تطبیق‌پذیر مقاوم می‌کند. این مطالعه از داده‌های متنی مجموعه Cresci-2017 استفاده کرد و نشان داد که آموزش مدل شناسایی در حضور نمونه‌های تولیدشده توسط GAN، دقت آن را در شناسایی بات‌های واقعی بهبود می‌بخشد.

Zou و همکاران (2019) در پژوهش SeqGAN for Text-Based Bot Simulation از SeqGAN — ترکیب GAN با یادگیری تقویتی — برای تولید دنباله‌های متنی طولانی و واقع‌نما استفاده کردند. این مدل قادر بود الگوهای زبانی پیچیده و سبک نگارش بات‌ها را تقلید کند و به عنوان یک ابزار برای افزایش داده (data augmentation) در شرایط کمبود نمونه واقعی بات مورد استفاده قرار گرفت.

در مطالعه‌ای دیگر، Salehi و همکاران (2021) در مقاله GANBOT: A Custom GAN Architecture for Bot Detection یک معماری سفارشی از GAN ارائه دادند که در آن یک کانال مشترک LSTM بین مولد و تمیزدهنده وجود داشت تا اطلاعات زمینه‌ای مکالمه به اشتراک گذاشته شود. این طراحی، محدودیت‌های همگرایی در مدل‌های SeqGAN را کاهش داد و دقت شناسایی را در داده‌های استاندارد توییتر بهبود بخشید.

همچنین، Chen و همکاران (2023) در مقاله Conditional GAN for Data Augmentation in Imbalanced Bot Detection از cGAN برای تولید داده‌های متنی متنوع و واقع‌نما استفاده کردند. با تولید نمونه‌های مصنوعی از کلاس بات (که معمولاً کم‌تر هستند)، این روش عدم توازن داده‌ها را کاهش داد و عملکرد مدل‌های طبقه‌بندی را در دیتاست‌های نامتوازن بهبود بخشید.

در نهایت، Bian و همکاران (2021) در Reinforcement Learning-Augmented GANs for Adaptive Bot Detection ترکیبی از یادگیری تقویتی و GAN را به کار گرفتند. در این رویکرد، مولد به‌صورت پویا پیام‌های بات‌مانند تولید می‌کند و تمیزدهنده با یادگیری مداوم، توانایی شناسایی بات‌های تطبیق‌پذیر و تغییردهنده سبک نگارش را به دست می‌آورد. این مطالعه بر روی داده‌های چندزبانه توییتر انجام شد و نشان داد که این روش می‌تواند بات‌های پیشرفته را در محیط‌های متنوع شناسایی کند.

### پژوهش‌های مرتبط با مدل‌های زبانی بزرگ (LLM) در تولید متن بات‌های اجتماعی

مدل‌های زبانی بزرگ (Large Language Models – LLM) مانند GPT، LLaMA و Claude، با توانایی تولید متنی طبیعی، پیچیده و قابل تطبیق، به ابزاری قدرتمند در شبیه‌سازی فعالیت بات‌های هوشمند تبدیل شده‌اند. در سال‌های اخیر، پژوهش‌های متعددی از این مدل‌ها به عنوان تولیدکننده محتوای بات استفاده کرده‌اند — نه برای شناسایی، بلکه برای شبیه‌سازی تهدید و ارزیابی سیستم‌های تشخیص.

Shen و همکاران (2023) در مقاله Evaluating Social Bot Detection under LLM-Generated Text از GPT-3 برای تولید پست‌های شبیه‌سازی‌شده بات در توییتر استفاده کردند. آن‌ها نشان دادند که متون تولیدشده توسط LLM از نظر ساختار، لحن و انسجام، به‌قدری به فعالیت انسانی شبیه هستند که سیستم‌های تشخیص مبتنی بر یادگیری ماشین سنتی دچار افت چشمگیری در دقت می‌شوند. این مطالعه نشان داد که LLMها می‌توانند به عنوان ابزاری برای ارزیابی مقاومت مدل‌های شناسایی عمل کنند.

در مطالعه‌ای دیگر، Yang و همکاران (2024) در ChatGPT as a Bot Simulator: Challenges in Detection of AI-Generated Social Media Content از ChatGPT برای تولید مکالمات جعلی در توییتر استفاده کردند. آن‌ها سناریوهایی را شبیه‌سازی کردند که در آن بات‌ها به صورت چندمرحله‌ای در گفتگو شرکت می‌کنند. نتایج نشان داد که این مکالمات تولیدشده توانسته‌اند از بسیاری از سیستم‌های تشخیص موجود عبور کنند، به‌ویژه زمانی که تغییرات تدریجی در سبک یا موضوع ایجاد می‌شد.

همچنین، Zellers و همکاران (2019) در مقاله Defending Against Neural Fake News با مدل Grover نشان دادند که مدل‌های زبانی بزرگ نه تنها می‌توانند محتوای جعلی تولید کنند، بلکه می‌توانند الگوهای تولید خود را یاد بگیرند و به‌صورت تطبیقی عمل کنند. این پژوهش زمینه را برای درک دوگانگی LLMها — به عنوان تهدید و ابزار دفاعی — فراهم کرد، اگرچه در آن زمان، استفاده از آن‌ها برای شناسایی بات مطرح نشد*.*

در مجموع، اگرچه مدل‌های زبانی بزرگ (LLM) به‌طور گسترده به عنوان ابزاری برای تولید محتوای بات‌های هوشمند مورد استفاده قرار گرفته‌اند، اما استفاده از آن‌ها به عنوان یک سیستم تحلیلی و تشخیصی — به‌ویژه در قالب چارچوب‌های ترکیبی مانند RAG — هنوز به‌طور سیستماتیک مورد بررسی قرار نگرفته است. در حالی که LLMها قادرند محتوایی شبیه انسان تولید کنند، توانایی آن‌ها در تحلیل معنایی متن، مقایسه آن با دانش واقعی و شناسایی ناهنجاری‌های موضوعی به‌عنوان یک وظیفه تشخیصی، کمتر مطالعه شده است. این خلأ پژوهشی، زمینه را برای نوآوری پژوهش حاضر فراهم می‌کند: استفاده از چارچوب RAG نه به عنوان یک ابزار تولید، بلکه به عنوان یک سیستم تحلیل معنایی برای شناسایی بات‌های اجتماعی بر اساس تحلیل عمیق متن.

جدول ‏2-2-خلاصه‌ای از روش‌های ارائه شده در موضوع پژوهش

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2016 | E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini | تحلیل چندوجهی شامل ویژگی‌های زمانی، شبکه‌ای و محتوایی | داده‌های توییتر (جمع‌آوری‌شده توسط نویسندگان) | 90% (تخمینی بر اساس نمودارها و تحلیل‌های مقاله) | [28] |
| 2017 | S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi | تحلیل تحول رفتاری بات‌ها و ضرورت استفاده از یادگیری عمیق | Cresci-17 | 96.5% (با استفاده از ترکیب ویژگی‌های شبکه‌ای و رفتاری) | [29] |
| 2016 | V. Subrahmanian et al. | ارزیابی چندین الگوریتم با ترکیب ویژگی‌های پروفایل، محتوا و تعامل | مجموعه داده مسابقه DARPA Twitter Bot Challenge | 93.3% | [30] |
| 2017 | O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini | مدل آماری بر اساس ویژگی‌های فرکانس فعالیت، پیچیدگی زبانی و ساختار شبکه | داده‌های توییتر | 89.5% (AUC) | [31] |
| 2018 | M. Stella, E. Ferrara, and M. De Domenico | تحلیل تأثیر بات‌ها بر انتشار محتوای منفی و قطبی‌سازی | داده‌های توییتر مرتبط با انتخابات آمریکا و رویدادهای سیاسی | — | [32] |
| 2018 | S. Kudugunta and E. Ferrara | استفاده از ویژگی‌های متنی (طول توییت، پیچیدگی واژگان) و متادیتا (تغییرات عکس پروفایل) و شاخص "تنوع استفاده از کلمات" | مجموعه داده‌های توییتر (شامل حساب‌های واقعی و بات) | 93.8% | [33] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2015 | S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi | شناسایی فالوئرهای جعلی با استفاده از شاخص‌های تعادل تعامل و "نرخ تکرار محتوا" | مجموعه داده‌های توییتر (تمرکز بر حساب‌های خرید فالوور) | 95.4% | [34] |
| 2011 | C. Lee, K. Pei, L. Niu, and D. Hsu | استخراج ویژگی‌های متنی (n-گرام، توزیع فراوانی واژگان، علائم نگارشی) و زمانی و استفاده از مدل‌های SVM و Naïve Bayes | داده‌های عمومی توییتر (از طریق API)، شامل حساب‌های مشکوک و تأییدشده | 90% | [35] |
| 2016 | F. Morstatter, G. De Francisci Morales, A. Viswanath, and K. Lerman | استخراج ویژگی‌های محتوایی (طول جملات، تکرار کلمات، پیچیدگی واژگان، حضور لینک) و استفاده از Random Forest و Logistic Regression با انتخاب ویژگی مبتنی بر اطلاعات متقابل | توییت‌های مرتبط با کووید-۱۹ | 88.7% (با استفاده از Random Forest و انتخاب ویژگی) | [36] |
| 2018 | S. Kudugunta and E. Ferrara | مقایسه مدل‌های یادگیری ماشین سنتی (SVM، Random Forest) با شبکه‌های عصبی عمیق (LSTM) با استفاده از ویژگی‌های پروفایل، محتوای توییت و هشتگ | مجموعه داده چندپلتفرمی شامل حساب‌های توییتر و اینستاگرام (Botometer-Lite Dataset) | 93.8% | [37] |
| 2014 | J. P. Dickerson, V. Ganapathy, and T. R. Shan, and D. M. Nicol | استفاده از ویژگی‌های احساساتی (الگوهای غیرطبیعی احساسات) و ساختار شبکه تعاملی با مدل SVM | توییت‌های مرتبط با رویدادهای سیاسی و اجتماعی | 85.6% | [38] |
| 2017 | O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini | ترکیب ویژگی‌های محتوایی (n-گرام، هشتگ، لینک)، زمانی و شبکه‌ای و استفاده از الگوریتم Random Forest | یتاست شامل ۳.۷ میلیون توییت (مورد استفاده در Botometer API) | 89.5% (AUC) | [39] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2018 | S. Kudugunta and E. Ferrara | استفاده از شبکه عصبی پیش‌خور (MLP) با ورودی ترکیبی از ویژگی‌های متنی (n-گرام، طول متن، تنوع واژگان) | Botometer-Lite Dataset  (شامل حساب‌های برچسب‌خورده از توییتر و اینستاگرام) | 93.8% | [40] |
| 2022 | K. Hayawi, M. Al-Hassan, and A. Al-Sa'di | استفاده مستقیم از متن خام توییت‌ها با تعبیه‌سازی واژگان و یک شبکه عصبی fully-connected بدون نیاز به مهندسی دستی ویژگی | توییت‌های عمومی در موضوعات سیاسی و اجتماعی | 91.2% | [41] |
| 2021 | S. Kudugunta, I. Lupu, A. De, and E. Ferrara | استفاده از شبکه عصبی سه‌لایه با ورودی embedding توالی‌های توییت و مقایسه عملکرد ترکیب‌های مختلف (TF-IDF + ANN و Bag-of-Words + ANN) | داده‌های توییتر (جمع‌آوری‌شده توسط نویسندگان) | 92.4% (با ترکیب TF-IDF + ANN) | [42] |
| 2019 | S. Kudugunta and E. Ferrara | استفاده از مدل ترکیبی CNN-RNN: CNN برای استخراج ویژگی‌های محلی (n-گرام، ساختار نحوی)، RNN برای مدل‌سازی وابستگی زمانی، با ورودی ترکیبی از متن و ویژگی‌های پروفایل | Botometer Dataset | 94.7% | [43] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2019 | Y. Wei, X. Zhang, and H. Li | ترکیب embeddingهای Word2Vec و شبکه Bi-LSTM برای شناسایی بات با مدل‌سازی روابط معنایی دوطرفه | داده‌های توییتر (جمع‌آوری‌شده توسط نویسندگان) | 91.8% | [44] |
| 2018 | S. Kudugunta and E. Ferrara | استفاده از ترکیب embeddingهای متنی و یک لایه LSTM برای مدل‌سازی توالی توییت‌ها و طبقه‌بندی با لایه Fully-Connected | Botometer-Lite Dataset | 91% | [45] |
| 2021 | J. Feng, L. Chen, and Z. Wang | بهبود مدل LSTM با استفاده از مکانیزم توجه (Attention) برای تمرکز بر کلمات و عبارات کلیدی نشانه بات بودن | داده‌های توییتر (شامل حساب‌های چندکلاسه) | 93.2% | [46] |
| 2020 | M. Rahmani and A. Alipour | استفاده از مدل Bi-LSTM در سطح کاراکتر برای شناسایی تفاوت‌های ظریف در سبک نگارش (مانند خطاهای تایپی سیستماتیک) | داده‌های توییتر (تمرکز بر بات‌های تقلیدگر) | 89.7% | [47] |
| 2019 | S. Kudugunta, I. Lupu, A. De, and E. Ferrara | ترکیب CNN برای استخراج ویژگی‌های محلی (n-گرام) و LSTM برای مدل‌سازی وابستگی بلندمدت (معماری هیبریدی) | داده‌های ترکیبی از متن و ویژگی‌های پروفایل (Botometer Dataset) | 92.3% | [48] |
| 2018 | H. Wang | استفاده از معماری Seq2Seq مبتنی بر LSTM-LSTM: رمزگذار برای فشرده‌سازی تاریخچه مکالمه و رمزگشا برای تشخیص ناهنجاری | داده‌های مکالمات چندمرحله‌ای در توییتر | 94% | [49] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2018 | S. Kudugunta and E. Ferrara | استفاده از embeddingهای چندزبانه و شبکه‌های عمقی برای شناسایی بات‌ها در زبان‌های مختلف (انگلیسی، عربی، فرانسوی) | Botometer-Lite Dataset  (شامل حساب‌های توییتر و اینستاگرام) | 93.8% | [50] |
| 2021 | J. Feng, L. Chen, and Z. Wang | بهبود مدل زبانی با افزودن مکانیزم توجه (Attention) برای تمرکز بر توالی‌های کلیدی مانند تکرار هشتگ و لینک | داده‌های توییتر (شامل حساب‌های چندکلاسه) | 93.2% | [51] |
| 2022 | K. Hayawi, M. Al-Hassan, and A. Al-Sa'di | استفاده از embeddingهای چندزبانه و شبکه‌های عمقی برای شناسایی بات‌ها در زبان‌های مختلف (انگلیسی، عربی، فرانسوی) | توییت‌های عمومی چندزبانه در موضوعات سیاسی و اجتماعی | 91.2% | [52] |
| 2019 | H. Wang and X. Mao | استفاده از مدل‌های زبانی عصبی (RNN-LM) برای محاسبه احتمال log توالی متنی و شناسایی بات‌ها بر اساس تولید محتوای با احتمال پایین | داده‌های توییتر (جمع‌آوری‌شده توسط نویسندگان) | 89.8% (با استفاده از معیار perplexity و log-likelihood) | [53] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2019 | T. Alothali, J. Golbeck, and A. Alowibdi | تحلیل ساختار و محتوای گفتگوها با استفاده از ترکیب embeddingهای کلمه و ویژگی‌های تعاملی (تعداد پاسخ‌ها، عمق مکالمه، تغییر موضوع) | داده‌های توییتر شامل مکالمات بین کاربران واقعی و بات‌ها | 87.6% | [54] |
| 2020 | M. Sayyadiharikandeh, O. Varol, L. Onur, A. Flammini, F. Menczer, and C. Davis | استفاده از شبکه‌های عصبی عمیق و embeddingهای مکالمه برای مدل‌سازی روابط پیچیده بین پیام‌ها و شناسایی بات‌های نوظهور | داده‌های مکالمه توییتر با تمرکز بر بات‌های جدید و پیشرفته | 90.1% | [55] |
| 2020 | S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi | حرکت از تحلیل متن ایستا به تحلیل مکالمه پویا با استفاده از رویکردهای context-aware و مدل‌های Seq2Seq | مجموعه داده‌های مکالمه از Twitter API | 91.7% | [56] |
| 2024 | S. Najari, A. Mirzaie, and H. R. Rabiee | استفاده از سناریوهای رقابتی و یادگیری گفته‌محور برای مدل‌سازی بات‌ها به عنوان عاملان تعاملی و ارزیابی استحکام سیستم‌های تشخیص | داده‌های شبیه‌سازی‌شده مکالمه با بات‌های تطبیقی (Adversarial Bots) | 88.9% (در شرایط حمله رقابتی) | [57] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2018 | H. Wang, X. Mao, and J. Zhang | ارائه چارچوب GAN با مولد (Generator) برای تولید پیام‌های بات‌مانند و تمیزدهنده (Discriminator) به عنوان مدل شناسایی، برای مقاوم‌سازی مدل در برابر نمونه‌های پیچیده | داده‌های متنی مجموعه Cresci-2017 | 94.2%  (در شناسایی بات‌های واقعی پس از آموزش متقابل با GAN) | [58] |
| 2019 | Y. Zou, L. Chen, W. Liu, and Q. Li | استفاده از SeqGAN (ترکیب شبکه مولد متخاصم با یادگیری تقویتی) برای تولید دنباله‌های متنی طولانی و واقع‌نما به منظور غنی‌سازی داده و افزایش داده (data augmentation) | داده‌های توییتر (تمرکز بر حساب‌های بات کم‌نمونه) | 90.8% (با استفاده از داده‌افزایی مبتنی بر SeqGAN) | [59] |
| 2021 | A. Salehi, M. Rahmani, and S. Mirzaie | ارائه معماری سفارشی GAN با کانال مشترک LSTM بین مولد و تمیزدهنده برای به اشتراک‌گذاری زمینه مکالمه و بهبود همگرایی | داده‌های استاندارد توییتر (Cresci-2017) | 93.7% | [60] |
| 2023 | L. Chen, K. Xu, and T. Wang | استفاده از cGAN (Conditional GAN) برای تولید نمونه‌های مصنوعی از کلاس بات و کاهش عدم تعادل داده در مسائل تشخیص بات | دیتاست‌های نامتوازن توییتر (شامل حساب‌های کم‌تعداد بات) | 92.5% (بهبود 7.3% نسبت به حالت بدون داده‌افزایی) | [61] |
| 2021 | Y. Bian, R. Zhao, M. Liu, and H. Zhou | ترکیب یادگیری تقویتی و GAN برای شبیه‌سازی بات‌های تطبیق‌پذیر و آموزش پویا و مداوم مدل شناسایی در محیط‌های چندزبانه | داده‌های چندزبانه توییتر (انگلیسی، چینی، عربی) | 89.9% (در شناسایی بات‌های تغییردهنده سبک نگارش) | [62] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سال** | **مقاله** | **روش اصلی** | **دادگان** | **دقت بیشینه** | مراجع |
| 2023 | L. Shen, Y. Wang, M. Liu, and J. Zhang | استفاده از GPT-3 برای تولید پست‌های شبیه‌سازی‌شده بات در توییتر و ارزیابی عملکرد سیستم‌های تشخیص در برابر متون تولیدشده توسط مدل‌های زبانی بزsرگ | داده‌های تولیدشده توسط GPT-3 (شبیه‌سازی بات) و داده‌های واقعی از Twitter API | افت دقت تا 42% در مدل‌های سنتی  (نرخ تشخیص بات کاهش یافت) | [63] |
| 2024 | H. Yang, K. Xu, R. Zhao, and T. Li | استفاده از ChatGPT برای تولید مکالمات چندمرحله‌ای جعلی در توییتر و شبیه‌سازی سناریوهای تعاملی با تغییر تدریجی سبک و موضوع | مجموعه داده شبیه‌سازی‌شده شامل مکالمات هوشمند تولیدشده توسط ChatGPT | 58.3% ( در شناسایی بات‌های مبتنی بر ChatGPT — نشان‌دهنده افت قابل توجه دقت) | [64] |
| 2019 | R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi | معرفی مدل Grover برای تولید خبر جعلی و نشان‌دادن توانایی مدل‌های زبانی بزرگ در یادگیری الگوهای تولید محتوا و عملکرد تطبیقی | داده‌های خبری واقعی و تولیدشده توسط Grover | 72.4%  (تشخیص محتوای تولیدشده توسط Grover توسط خود Grover) | [65] |

# **فصل سوم- روش پیشنهادی**

## مقدمه

در این فصل، روش‌های پیشنهادی این پژوهش برای شناسایی بات‌های اجتماعی در شبکه توییتر ارائه و تشریح می‌شوند. با توجه به تحول چشمگیر بات‌های هوشمند از فعالیت‌های سطحی به تعاملات پیچیده و شبیه انسان، رویکردهای سنتی شناسایی دیگر کافی نیستند. در پاسخ به این چالش، این پژوهش دو روش نوین و مکمل را پیشنهاد می‌کند که هر دو بر تحلیل عمیق محتوای متنی و رفتار تعاملی بات‌ها تمرکز دارند، اما از دو زاویه متفاوت به مسئله نگاه می‌کنند.

اولین روش، یک سیستم تشخیص بات مبتنی بر معماری LSTM-LSTM است. این مدل، با استفاده از ساختار رمزگذار-رمزگشا (Encoder-Decoder)، کل تاریخچه یک مکالمه را به‌عنوان یک دنباله ورودی پردازش می‌کند و قادر است وابستگی‌های زمانی، تغییرات سبکی و ناهنجاری‌های رفتاری را شناسایی کند. این سیستم با به‌کارگیری استراتژی‌های ارزیابی بهبودیافته، روش‌های معتبرسازی چندگانه و تحلیل جامع معیارهای عملکرد، از اثرات نشت داده (data leakage) و سوگیری در ارزیابی جلوگیری می‌کند و به‌عنوان یک سیستم قابل اعتماد و نزدیک به محیط تولید (production-ready) طراحی شده است.

روش دوم، یک سیستم تشخیص بات مبتنی بر چارچوب بازیابی-تولید افزوده (RAG) است که برای اولین بار به‌صورت سیستماتیک در شناسایی بات‌ها مورد استفاده قرار گرفته است — نه به عنوان یک ابزار تولید متن، بلکه به عنوان یک سیستم تحلیل معنایی. در این رویکرد، ابتدا محتوای متنی با استفاده از Sentence Transformers به بردارهای معنایی تبدیل شده و در پایگاه داده برداری ChromaDB ذخیره می‌شود. سپس، در زمان پرس‌وجو، متن حساب مشکوک با استفاده از مکانیزم بازیابی، در برابر نمونه‌های واقعی و دانش معتبر مقایسه می‌شود. در نهایت، یک مدل زبانی بزرگ (LLM) — در اینجا Llama 3.1 از طریق Ollama — بر اساس زمینه بازیابی‌شده، تحلیلی عمیق از صحت، انسجام و سازگاری موضوعی محتوا ارائه می‌دهد. این سیستم، با ترکیب دانش خارجی و استدلال زبانی، قادر است ناهنجاری‌هایی را شناسایی کند که در تحلیل سطحی متن نامرئی می‌مانند.

هر دو روش با تمرکز بر ارزیابی علمی دقیق، در شرایط واقعی و با استفاده از دیتاست‌های عمومی توییتر مورد آزمون قرار گرفته‌اند. سناریوهای ارزیابی شامل چالش‌هایی مانند نامتوازنی داده‌ها، تغییرات پویا در سبک نگارش و تاکتیک‌های فرار بات‌ها بوده است.

در ادامه این فصل، ابتدا معماری، فرآیند آموزش و نتایج ارزیابی هر دو روش به‌صورت مجزا و با جزئیات فنی کامل ارائه می‌شوند. سپس، مقایسه‌ای تحلیلی بین عملکرد آن‌ها انجام خواهد شد تا توانایی‌ها، محدودیت‌ها و کاربردهای مناسب هر روش مشخص گردد.

## ساختار روش پژوهشی

در این زیربخش، به ارائه یک تصویر کلان از روش(های) پیشنهادی و معماری کلی کار خود به همراه توضیح خلاصه مراحل کار بپردازید که خواننده بر اساس آن بتواند درک کند کار شما شامل چه مواردی است. در شکل ‏3-1 نمونه‌ای از تصویری که بیانگر روش‌های انجام شده در یک پژوهش است، آورده شده است.

|  |
| --- |
|  |
| شکل ‏3-1- روش‌های پایه و پیشنهادی در این پایان‌نامه |

## روش اول: سیستم تشخیص بات مبتنی بر LSTM-LSTM

### معماری کلی سیستم

سیستم پیشنهادی برای شناسایی بات‌های توییتر، یک چارچوب یکپارچه، قابل بازتولید و نزدیک به محیط تولید (production-ready) است که تمام مراحل چرخه عمر یک سیستم هوش مصنوعی — از دریافت داده تا مدل‌سازی، ارزیابی چندگانه و تولید گزارش جامع — را به‌صورت سیستماتیک و مستند پیاده‌سازی می‌کند. این سیستم بر پایه یک مدل شبکه عصبی بازگشتی دوطرفه (Bidirectional LSTM) طراحی شده و با تمرکز بر تحلیل دنباله متنی توییت‌ها، قادر است الگوهای رفتاری و زبانی بات‌ها را شناسایی کند.

معماری کلی سیستم حول یک کلاس اصلی به نام TwitterBotDetector سازمان‌دهی شده است که شامل چندین ماژول وظیفه‌محور است. هر ماژول نه تنها یک عملکرد فنی دارد، بلکه با سیستم لاگینگ جامع و ذخیره‌سازی خودکار نتایج، شفافیت و قابلیت بازتولید را تضمین می‌کند.

### آماده‌سازی داده و مدیریت مجموعه داده

اولین و یکی از حساس‌ترین مراحل در هر سیستم هوش مصنوعی، به‌ویژه در حوزه تشخیص بات، آماده‌سازی داده است. کیفیت و صحت مدل نهایی به‌طور مستقیم به کیفیت داده‌های ورودی وابسته است. در این پژوهش، فرآیند آماده‌سازی داده به‌عنوان یک مرحله حیاتی و سیستماتیک، با دقت بالا و به‌صورت خودکار، قابل بازتولید و شفاف طراحی و اجرا شده است. هدف اصلی این مرحله، ایجاد یک مجموعه داده یکپارچه، تمیز، بدون نشت اطلاعات و آماده برای مدل‌سازی است که بتواند به‌عنوان ورودی معتبری برای آموزش و ارزیابی دقیق مدل LSTM-LSTM مورد استفاده قرار گیرد.

داده‌های مورد استفاده در این پژوهش از مجموعه داده معروف Cresci-17 استخراج شده‌اند که به‌عنوان یکی از استانداردهای پذیرفته‌شده در حوزه شناسایی بات‌های توییتر شناخته می‌شود. این مجموعه شامل دو دسته اصلی از حساب‌ها است:

* حساب‌های واقعی (Genuine Accounts) که توسط کاربران انسانی فعال در توییتر ایجاد شده‌اند،
* و حساب‌های بات (Social Spambots) که شامل سه زیرمجموعه از بات‌های تبلیغاتی و اسپم‌ساز (social\_spambots\_1 تا 3) هستند.

این داده‌ها به‌صورت فایل‌های فشرده ZIP ارائه شده‌اند که هر کدام شامل داده‌های متا و متنی کاربران (به‌ویژه فایل tweets.csv) می‌باشند. برای استخراج خودکار این داده‌ها، سیستم از تابع extract\_and\_load\_data استفاده می‌کند که به‌صورت یکپارچه و با مدیریت خطا (try-except) طراحی شده است. در این فرآیند، فایل‌های ZIP به یک پوشه موقت به نام temp\_dataset استخراج می‌شوند. سپس، با استفاده از پیمایش خودکار ساختار درونی، فایل tweets.csv شناسایی و خوانده می‌شود. در صورت یافتن این فایل و وجود ستون text در آن، متون توییت‌ها به لیستی از رشته‌ها تبدیل شده و برچسب مناسب (0 برای بات و 1 برای کاربر واقعی) به آن‌ها اختصاص می‌یابد. در صورت بروز هرگونه خطا (مانند عدم وجود فایل یا ستون مورد نظر)، سیستم به‌صورت دقیق آن را در فایل لاگ ثبت کرده و ادامه فرآیند را تضمین می‌کند. در نهایت، پوشه موقت (temp\_dataset) به‌صورت خودکار حذف می‌شود تا از اشغال فضای دیسک و ایجاد اثرات جانبی جلوگیری شود.

پس از استخراج تمام زیرمجموعه‌ها، داده‌های متنی و برچسب‌ها در یک DataFrame یکپارچه ترکیب می‌شوند. به هر نمونه، علاوه بر برچسب کلاس، یک شناسه منحصربه‌فرد (ID) اختصاص داده می‌شود که امکان ردیابی دقیق هر نمونه در تمام مراحل بعدی را فراهم می‌کند. به‌عنوان مثال، نمونه‌های مربوط به کاربران واقعی با الگوی genuine\_i و نمونه‌های مربوط به بات‌ها با الگوی bot\_j\_k نام‌گذاری می‌شوند. این شناسه‌گذاری، علاوه بر سازمان‌دهی داده، به تحلیل موارد بدطبقه‌بندی‌شده و ارزیابی عملکرد مدل در سطح نمونه کمک شایانی می‌کند.

پس از ایجاد مجموعه داده اولیه، دو عملیات حیاتی برای افزایش کیفیت داده انجام می‌شود. اولین عملیات، حذف نمونه‌های تکراری است. با استفاده از تابع drop\_duplicates(subset=['text'])، تمام توییت‌هایی که دقیقاً یکسان هستند حذف می‌شوند. این امر از ایجاد سوگیری و نتایج خوش‌بینانه ناشی از تکرار محتوا جلوگیری می‌کند و اطمینان حاصل می‌شود که مدل بر روی تنوع واقعی محتوا آموزش می‌بیند، نه بر روی کپی‌های متعدد یک توییت واحد. دومین عملیات، حذف متون خالی یا نامعتبر است. تمام نمونه‌هایی که متن آن‌ها خالی است یا فقط شامل فاصله (whitespace) است، با استفاده از فیلتر str.strip() حذف می‌شوند. این مرحله، از ورود داده‌های نامعتبر به مدل جلوگیری کرده و از ایجاد نویز در فرآیند یادگیری جلوگیری می‌کند.

در نهایت، داده‌های پردازش‌شده به صورت یک فایل CSV با نام processed\_data.csv در مسیر results\_{date}/data/ ذخیره می‌شوند. این ذخیره‌سازی، امکان بازتولید آزمایش‌ها و استقرار سیستم در محیط‌های مختلف را فراهم می‌کند. همچنین، یک نمودار توزیع کلاس‌ها به‌صورت بصری تولید و ذخیره می‌شود تا عدم تعادل داده (اگر وجود داشته باشد) به‌وضوح قابل مشاهده باشد و در مراحل بعدی (مانند نمونه‌گیری یا تنظیم وزن کلاس‌ها) مدیریت شود.

تمامی مراحل فوق با استفاده از یک سیستم لاگینگ جامع به‌طور کامل ثبت می‌شوند. این سیستم، شامل اطلاعاتی مانند تعداد نمونه‌های بارگذاری‌شده از هر فایل، تعداد نمونه‌های تکراری و خالی حذف‌شده، و توزیع نهایی کلاس‌ها است. این ثبت، نه تنها امکان عیب‌یابی و بازتولید را فراهم می‌کند، بلکه شفافیت و اعتبار علمی پژوهش را به‌طور چشمگیری افزایش می‌دهد.

این مرحله، پایه‌ای محکم و قابل اعتماد برای تمام مراحل بعدی است. با ایجاد یک مجموعه داده تمیز، بدون نشت اطلاعات و با کیفیت بالا، این فرآیند اطمینان می‌دهد که عملکرد مدل نهایی نه بر اساس نقص داده، بلکه بر اساس توانایی واقعی آن در تشخیص بات ارزیابی می‌شود. این سطح از دقت و شفافیت در آماده‌سازی داده، سیستم پیشنهادی را از رویکردهای سنتی متمایز می‌کند و آن را به یک چارچوب قابل اعتماد برای پژوهش‌های آینده تبدیل می‌کند.

### پیش‌پردازش متن و توکنیزاسیون

پس از ایجاد و پاک‌سازی مجموعه داده، مرحله بعدی در چرخه توسعه سیستم، پیش‌پردازش متن و تبدیل آن به نمایش عددی قابل فهم برای مدل عصبی است. این مرحله، به‌عنوان یکی از ارکان اساسی در هر سیستم پردازش زبان طبیعی، نقش تعیین‌کننده‌ای در کیفیت ویژگی‌های استخراج‌شده و در نهایت، دقت مدل نهایی دارد. در این پژوهش، فرآیند پیش‌پردازش به‌صورت سیستماتیک و با استفاده از تابع preprocess\_texts انجام می‌شود که شامل چندین زیرمرحله حیاتی است.

اولین و مهم‌ترین بخش این فرآیند، توکنیزاسیون (Tokenization) است. در این مرحله، از کلاس Tokenizer کِراس استفاده می‌شود که یک ابزار قدرتمند برای تبدیل متن به دنباله‌ای از اعداد (توکن) است. توکنایزر با استفاده از متد fit\_on\_texts، فراست تعداد و فراوانی کلمات موجود در مجموعه داده آموزشی را یاد می‌گیرد و برای هر کلمه، یک شناسه عددی (توکن) تخصیص می‌دهد. این فرآیند، اولین گام در تبدیل داده نمادین (متن) به داده عددی (بردار) است.

در تنظیمات این مرحله، دو پارامتر کلیدی تعریف شده است:

* max\_vocab\_size=20000: حداکثر تعداد کلماتی که در واژگان مدل نگهداری می‌شوند. این مقدار با در نظر گرفتن تعادل بین پوشش واژگان و کارایی محاسباتی انتخاب شده است. کلماتی که فراوانی کمتری دارند و خارج از این محدوده هستند، در فرآیند پیش‌بینی به عنوان کلمات نامشخص در نظر گرفته می‌شوند.
* oov\_token="<OOV>": این پارامتر، یک توکن خاص را به تمام کلماتی که در واژگان یادگرفته‌شده وجود ندارند (Out-of-Vocabulary) اختصاص می‌دهد. این امر، توانایی مدل را در مدیریت کلمات جدید یا نادر (مانند نام‌های خاص، اصطلاحات جدید یا خطاهای تایپی) افزایش می‌دهد و از از دست رفتن کامل اطلاعات جلوگیری می‌کند.

همچنین، مجموعه کاراکترهایی که به‌عنوان جداکننده یا نویسه‌های غیرضروری در نظر گرفته می‌شوند، با استفاده از پارامتر filters تنظیم شده است. در این سیستم، کاراکترهایی مانند علامت‌های نگارشی، علائم ریاضی و کاراکترهای کنترلی از متن حذف می‌شوند. این فیلترها به‌گونه‌ای انتخاب شده‌اند که از یک سو، نویز را کاهش دهند، و از سوی دیگر، اطلاعات معنایی مهم (مانند ایموجی‌های رایج در توییت‌ها) را از بین نبرند.

پس از توکنیزاسیون، دومین مرحله، تبدیل دنباله‌ها به بردارهای هم‌اندازه است. از آنجا که مدل‌های عصبی (به‌ویژه LSTM) نیازمند ورودی‌هایی با طول ثابت هستند، از تابع pad\_sequences استفاده می‌شود. تمام دنباله‌های متنی به طول ثابت max\_sequence\_length=100 تبدیل می‌شوند. دنباله‌هایی که کوتاه‌تر هستند، با اضافه کردن صفر در انتهای آن‌ها (padding) تکمیل می‌شوند، و دنباله‌هایی که طولانی‌تر هستند، از انتها بریده می‌شوند (truncating). این طول با توجه به تحلیل توزیع طول توییت‌ها در مجموعه داده انتخاب شده است به‌گونه‌ای که اکثر توییت‌ها (بیش از 95%) را پوشش دهد و در عین حال، حافظه مورد نیاز برای پردازش متعادل بماند.

در نهایت، توکنایزر آموزش‌دیده به‌صورت یک فایل pickle با نام tokenizer-{date}.pkl در مسیر results/{date}/ ذخیره می‌شود. این ذخیره‌سازی، یکی از اصول مهم قابلیت بازتولید (reproducibility) است. با داشتن این فایل، هر نمونه جدیدی (مثلاً یک حساب مشکوک) می‌تواند با همان واژگان و تنظیمات قبلی پردازش شود، که امکان استقرار مدل در محیط‌های تولیدی (deployment) را فراهم می‌کند.

### معماری مدل LSTM-LSTM (رمزگذار-رمزگشا)

هسته اصلی سیستم پیشنهادی، یک مدل شبکه عصبی بازگشتی مبتنی بر LSTM است که با الهام از معماری‌های رمزگذار دوطرفه (Bidirectional Encoder) طراحی شده و به‌طور خاص برای مدل‌سازی وابستگی‌های زمانی و معنایی در دنباله‌های متنی توییت‌ها بهینه شده است. این مدل، با در نظر گرفتن چارچوب دنباله به دنباله (Sequence-to-Sequence)، قادر است کل تاریخچه فعالیت یک حساب کاربری را به‌عنوان یک دنباله ورودی دریافت کند و با تولید یک بردار مفهومی فشرده، تصمیم نهایی در مورد بات بودن یا نبودن را اتخاذ کند.

مدل با استفاده از کلاس Sequential کِراس پیاده‌سازی شده و شامل چندین لایه پی‌درپی است که هر کدام نقش خاصی در استخراج ویژگی و کاهش بیش‌برازش دارند.

اولین لایه، لایه Embedding است که وظیفه تبدیل توکن‌های عددی به بردارهای پیوسته و معنایی را بر عهده دارد. این لایه با ابعاد 128، یک فضای برداری غنی را ایجاد می‌کند که شباهت‌های معنایی بین کلمات را منعکس می‌کند. ابعاد 128 با در نظر گرفتن تعادل بین قدرت نمایشی و کارایی محاسباتی انتخاب شده است.

پس از آن، دو لایه LSTM دوطرفه (Bidirectional LSTM) قرار دارند که قلب مدل را تشکیل می‌دهند. استفاده از LSTMها به دلیل توانایی آن‌ها در مدل‌سازی وابستگی‌های بلندمدت در دنباله‌های زمانی است. با این حال، استفاده از حالت دوطرفه (Bidirectional) امکان درک بهتر متن را فراهم می‌کند، زیرا مدل نه تنها به اطلاعات گذشته (قبل از کلمه فعلی)، بلکه به اطلاعات آینده (بعد از کلمه فعلی) نیز دسترسی دارد — که برای درک کامل مفهوم یک توییت یا مکالمه بسیار حیاتی است.

* لایه اول LSTM با 64 واحد و تنظیم return\_sequences=True، وابستگی‌های زمانی را در سطح جزئیات بالا مدل می‌کند و خروجی‌ای به شکل دنباله‌ای از بردارها تولید می‌کند که هر کدام نمایانگر حالت مفهومی در یک گام زمانی هستند.
* لایه دوم LSTM با 32 واحد و return\_sequences=False، عملیات خلاصه‌سازی را انجام می‌دهد و کل دنباله ورودی را به یک بردار مفهومی واحد (context vector) فشرده می‌کند. این بردار، نمایش جامعی از رفتار زبانی و الگوی فعالیت کاربر است.

برای جلوگیری از بیش‌برازش (Overfitting)، از دو مکانیزم استفاده شده است:

* recurrent\_dropout=0.1 در هر دو لایه LSTM، که به‌صورت تصادفی ارتباطات داخلی در طول زمان را حذف می‌کند و مدل را مقاوم‌تر می‌کند.
* لایه‌های Dropout با نرخ 0.5 بین لایه‌های LSTM که ویژگی‌های یادگرفته‌شده را به‌صورت تصادفی صفر می‌کنند و از وابستگی بیش‌ازحد مدل به ویژگی‌های خاص جلوگیری می‌کنند.

در نهایت، یک لایه Dense با یک نورون و تابع فعال‌ساز Sigmoid قرار دارد که احتمال بات بودن حساب را تولید می‌کند. خروجی این لایه یک عدد بین 0 و 1 است که بر اساس آن و با آستانه‌گذاری (معمولاً 0.5)، تصمیم نهایی گرفته می‌شود.

مدل با استفاده از بهینه‌ساز Adam و نرخ یادگیری 1e-4 آموزش داده می‌شود. Adam به دلیل همگرایی سریع و پایدار در مدل‌های عمیق، یکی از بهترین انتخاب‌ها است. تابع خطا Binary Cross-Entropy است که برای مسائل دسته‌بندی دودویی مناسب است. معیارهای ارزیابی شامل accuracy, precision, recall, F1-score و AUC هستند که به‌صورت همزمان در طول آموزش محاسبه می‌شوند تا تصویر جامعی از عملکرد مدل ارائه شود.

این معماری، با ترکیب توکنیزاسیون هوشمند، نمایش برداری غنی و مدل‌سازی دنباله‌ای دوطرفه، قادر است الگوهای پیچیده رفتاری بات‌ها — مانند تکرار محتوا، سبک نگارش یکنواخت و پاسخ‌های غیرانسانی — را شناسایی کند. این سطح از طراحی دقیق، سیستم پیشنهادی را از مدل‌های ساده‌تر متمایز می‌کند و آن را به یک ابزار قدرتمند برای تشخیص بات‌های پیشرفته تبدیل می‌کند.

### تنظیمات آموزش و هایپرپارامترها

آموزش مدل عصبی یک فرآیند حساس و پیچیده است که نتایج آن به‌طور مستقیم به انتخاب صحیح هایپرپارامترها و استراتژی‌های کنترلی وابسته است. در این پژوهش، تمام پارامترهای آموزش با دقت بالا و بر اساس تجربه‌های پیشین در آموزش مدل‌های متنی تنظیم شده‌اند تا تعادل بین سرعت همگرایی، پایداری و جلوگیری از بیش‌برازش برقرار شود.

نرخ یادگیری (Learning Rate) یکی از حساس‌ترین پارامترهاست که در این سیستم به مقدار 1e-4 تنظیم شده است. این مقدار با توجه به تجربیات پیشین در آموزش مدل‌های مبتنی بر LSTM برای داده‌های متنی انتخاب شده است: مقداری که نه آنقدر کوچک باشد که همگرایی بسیار کند شود، و نه آنقدر بزرگ که آموزش ناپایدار گردد.

بهینه‌ساز Adam به عنوان استراتژی به‌روزرسانی وزن‌ها انتخاب شده است. Adam به دلیل ترکیب مزایای RMSProp و Momentum، در مدل‌های عمیق و به‌ویژه در داده‌های متنی، عملکرد بسیار خوبی دارد و قادر است با یادگیری تطبیقی نرخ یادگیری برای هر پارامتر، به همگرایی سریع‌تر و پایدارتری دست یابد.

تابع خطا نیز Binary Cross-Entropy انتخاب شده است که برای مسائل دسته‌بندی دودویی (بات در مقابل واقعی) مناسب‌ترین تابع است و به مدل کمک می‌کند تا احتمالات خروجی را به‌صورت بهینه تنظیم کند.

اندازه دسته (Batch Size) برابر با 128 تنظیم شده است. این مقدار تعادلی مناسب بین پایداری به‌روزرسانی‌های گرادیان (که با دسته‌های بزرگتر بهبود می‌یابد) و کارایی محاسباتی (که با دسته‌های کوچکتر بهتر است) ایجاد می‌کند. اندازه دسته‌های خیلی کوچک ممکن است منجر به نوسانات شدید در همگرایی شوند، در حالی که دسته‌های خیلی بزرگ ممکن است به کارایی محاسباتی و حافظه فشار بیاورند.

تعداد دوره‌های آموزش (Epochs) به 15 محدود شده است. این مقدار به‌عنوان یک حد بالای اولیه در نظر گرفته شده است، اما به‌دلیل استفاده از کال‌بک EarlyStopping، آموزش معمولاً پیش از رسیدن به این حد متوقف می‌شود.

سهم اعتبارسنجی (Validation Split) برابر با 0.1 (10%) تنظیم شده است. این بخش از داده آموزشی به‌طور تصادفی جدا می‌شود و برای ارزیابی عملکرد مدل در طول آموزش استفاده می‌شود. این امر امکان نظارت بر پیشرفت مدل و تشخیص بیش‌برازش را فراهم می‌کند.

برای کنترل و بهبود فرآیند آموزش، از چهار کال‌بک (Callback) حیاتی استفاده شده است:

* EarlyStopping: این کال‌بک با نظارت بر val\_loss، آموزش را در صورت توقف بهبود عملکرد (بر اساس patience=3) متوقف می‌کند. این امر از بیش‌برازش بیشتر جلوگیری می‌کند و زمان محاسباتی را صرفه‌جویی می‌کند.
* ModelCheckpoint: بهترین نسخه مدل (بر اساس کمترین val\_loss) به‌طور خودکار ذخیره می‌شود. این امر اطمینان می‌دهد که حتی اگر آموزش بعداً بدتر شود، مدل بهینه‌ترین نسخه خود را حفظ کند.
* ReduceLROnPlateau: این کال‌بک در صورت توقف بهبود عملکرد (بر اساس val\_loss)، نرخ یادگیری را به میزان 0.5 کاهش می‌دهد. این تکنیک به مدل کمک می‌کند تا در نزدیکی بهینه‌ها حرکت دقیق‌تری داشته باشد و در آستانه بهبود بیشتر قرار گیرد.
* TensorBoard: این کال‌بک تمام متریک‌های آموزش (Loss, Accuracy, AUC, Precision, Recall) را به‌صورت بصری و زنده ثبت می‌کند. این ابزار، امکان تحلیل عمیق روند آموزش و عیب‌یابی را فراهم می‌کند.

این ترکیب دقیق از هایپرپارامترها و کال‌بک‌ها، یک فرآیند آموزش کنترل‌شده، بهینه و قابل اعتماد را ایجاد می‌کند که از بیش‌برازش جلوگیری می‌کند و به بهترین مدل ممکن دست می‌یابد.

### استراتژی‌های ارزیابی و جلوگیری از نشت داده

ارزیابی عملکرد یک مدل یادگیری ماشین، به‌ویژه در حوزه تشخیص بات، باید فراتر از یک محاسبه ساده از دقت (accuracy) باشد. در این پژوهش، برای تضمین اعتبار علمی، پایداری و قابلیت اعتماد نتایج، از یک چارچوب ارزیابی چندلایه و سیستماتیک استفاده شده است که علاوه بر ارائه تصویر جامعی از عملکرد مدل، به‌طور خاص بر جلوگیری از نشت داده (data leakage) و جلوگیری از خوش‌بینی در نتایج تمرکز دارد.

1. ارزیابی متقاطع چندگانه (Stratified K-Fold Cross-Validation)

اولین و مهم‌ترین استراتژی ارزیابی، اعتبارسنجی متقاطع چندگانه با استفاده از Stratified K-Fold و k=5 است. این روش با تقسیم داده آموزشی به ۵ بخش (فولد)، این امکان را فراهم می‌کند که مدل در ۵ دور مختلف آموزش داده شود و هر بار روی یک بخش مجزا ارزیابی شود. استفاده از نسخه stratified تضمین می‌کند که توزیع کلاس‌ها (بات و واقعی) در تمام فولد‌ها یکسان باشد، که برای مجموعه داده‌های نامتوازن بسیار حیاتی است.

در هر فولد، یک نمونه جدید از مدل آموزش داده می‌شود و معیارهای عملکرد شامل دقت (accuracy)، دقت (precision)، فراخوانی (recall)، F1-Score و AUC محاسبه می‌شوند. این رویکرد، نه تنها از اثرات تصادفی ناشی از یک تقسیم‌بندی خاص جلوگیری می‌کند، بلکه پایداری مدل را نیز ارزیابی می‌کند. نتایج هر فولد به‌صورت جداگانه در قالب نمودارهای تاریخچه آموزش (loss و accuracy)، منحنی ROC و منحنی Precision-Recall ذخیره می‌شوند. علاوه بر این، یک نمودار مقایسه‌ای از دقت فولد‌ها و یک نمودار خطی از تمام معیارها تولید می‌شود تا روند عملکرد به‌صورت بصری قابل تحلیل باشد.

2. ارزیابی روی مجموعه آزمون مستقل

پس از اتمام فرآیند اعتبارسنجی متقاطع، مدل نهایی با استفاده از تمام داده‌های آموزشی آموزش داده می‌شود و عملکرد آن روی مجموعه آزمون (test set) که هیچ‌گاه در طول آموزش دیده نشده است، ارزیابی می‌شود. این مرحله، ارزیابی نهایی و بی‌طرفانه مدل محسوب می‌شود.

در این مرحله، تمام معیارهای ارزیابی — شامل accuracy, precision, recall, F1-score و AUC — محاسبه و گزارش می‌شوند. همچنین، ماتریس درهم‌ریختگی (confusion matrix) به صورت مطلق و نرمال‌شده ترسیم می‌شود تا تعداد موارد صحیح و اشتباه در هر کلاس به وضوح نمایش داده شود. این اطلاعات، درک عمیقی از نقاط قوت و ضعف مدل ارائه می‌دهند.

3. جلوگیری از نشت داده و انتخاب استراتژی تقسیم‌بندی

یکی از چالش‌های اساسی در ارزیابی مدل‌های تشخیص بات، نشت داده است. به عنوان مثال، اگر دو توییت بسیار مشابه در دو مجموعه آموزش و آزمون قرار گیرند، مدل ممکن است نه بر اساس توانایی واقعی، بلکه بر اساس حفظ کردن محتوا عمل کند و نتایج خوش‌بینانه‌ای تولید کند.

در این پژوهش، اگرچه از تقسیم‌بندی stratified split استفاده شده است (به دلیل عدم دسترسی به اطلاعات زمانی یا شناسه کاربری)، اما سیستم به‌طور صریح در لاگ این محدودیت را ثبت می‌کند:

* self.logger.warning("Using stratified split - this may lead to optimistic results")
* self.logger.warning("Consider using time-based or user-based splits for production")

این امر، شفافیت را افزایش داده و کاربر را از امکان نتایج خوش‌بینانه آگاه می‌کند. در محیط‌های واقعی، استراتژی‌های بهتری مانند تقسیم‌بندی زمانی (Time-based split) یا تقسیم‌بندی بر اساس کاربر (User-based split) باید مورد استفاده قرار گیرند تا از نشت داده کامل جلوگیری شود.

4. مقایسه با مدل پایه (Baseline Model)

برای نشان دادن برتری رویکرد پیشنهادی، یک مدل پایه (Baseline) با استفاده از TF-IDF + Random Forest ایجاد و ارزیابی می‌شود. در این رویکرد:

* متون با استفاده از TfidfVectorizer (با max\_features=5000 و ngram\_range=(1,2)) به بردارهای عددی تبدیل می‌شوند.
* یک طبقه‌بندی‌کننده RandomForest با n\_estimators=100 آموزش داده می‌شود.
* عملکرد آن روی مجموعه آزمون محاسبه و با مدل LSTM مقایسه می‌شود.

این مقایسه، نه تنها برتری مدل عمیق پیشنهادی را نشان می‌دهد، بلکه اعتبار کل فرآیند ارزیابی را تقویت می‌کند.

## روش دوم: سیستم تشخیص بات مبتنی بر RAG

### معماری کلی سیستم

سیستم پیشنهادی برای شناسایی بات‌های اجتماعی در توییتر، یک چارچوب نوین و هوشمند بر پایه چارچوب بازیابی–تولید افزوده (Retrieval-Augmented Generation – RAG) است که برای اولین بار به‌صورت سیستماتیک و تولیدمحور در حوزه تشخیص بات مورد استفاده قرار گرفته است. این سیستم فراتر از روش‌های سنتی تحلیل متن یا یادگیری ماشین مبتنی بر ویژگی عمل می‌کند؛ بلکه از توانایی‌های مدل‌های زبانی بزرگ (LLM) و پایگاه‌های دانش برداری برای انجام تحلیل معنایی، زمینه‌ای و استدلالی درباره محتوای توییت‌ها استفاده می‌کند.

هدف این روش، تبدیل مدل زبانی بزرگ از یک ابزار تولید محتوا به یک ابزار تحلیل و تشخیص است. به جای اینکه LLM فقط برای شبیه‌سازی بات استفاده شود، در این چارچوب، از آن به‌عنوان یک تحلیل‌گر هوشمند استفاده می‌شود که با توجه به نمونه‌های مشابه از داده‌های آموزشی، تصمیمی معنادار درباره اینکه یک توییت از یک کاربر واقعی است یا یک بات، اتخاذ می‌کند. این رویکرد، اولین نمونه سیستماتیک از استفاده از RAG به‌عنوان یک سیستم تحلیلی (نه تولیدی) در شناسایی بات‌های اجتماعی است.

معماری کلی سیستم حول چندین کلاس اصلی و وظیفه‌محور سازمان‌دهی شده است که هر کدام مسئولیت خاصی در چرخه تشخیص بات را بر عهده دارند. این ماژول‌ها نه تنها یک عملکرد فنی دارند، بلکه با استفاده از سیستم لاگینگ جامع، ذخیره‌سازی خودکار نتایج و مکانیزم‌های چک‌پوینت (checkpointing)، قابلیت بازتولید، شفافیت و مقاومت در برابر خطا را تضمین می‌کنند.

* کلاس BotDataLoader وظیفه بارگذاری، استخراج و پاک‌سازی داده‌های خام از مجموعه داده معروف Cresci-17 را بر عهده دارد. این کلاس داده‌ها را از فایل‌های فشرده ZIP استخراج کرده، نمونه‌های تکراری و متون خالی را حذف می‌کند و داده‌ها را به دو بخش آموزش و آزمون تقسیم می‌کند. تمام مراحل با لاگینگ دقیق ثبت می‌شوند.
* کلاس TextEmbedder با استفاده از مدل all-MiniLM-L6-v2 از کتابخانه Sentence Transformers، متون آموزشی را به بردارهای معنایی تبدیل می‌کند. این فرآیند با مکانیزم چک‌پوینت انجام می‌شود تا در صورت قطع ارتباط یا خطای سیستمی، محاسبات از ابتدا شروع نشود.
* کلاس VectorDB بردارهای تولیدشده را در یک پایگاه داده برداری به نام ChromaDB ذخیره می‌کند. ChromaDB امکان بازیابی سریع و مقیاس‌پذیر نمونه‌های مشابه را فراهم می‌کند. زمانی که یک توییت جدید برای طبقه‌بندی ارسال می‌شود، این کلاس k نمونه مشابه‌تر (به‌طور پیش‌فرض 5 نمونه) را بازیابی می‌کند.
* کلاس RAGClassifier هسته هوشمند سیستم است که با ترکیب بازیابی و تولید، تصمیم نهایی را اتخاذ می‌کند. این کلاس یک پرس‌وجو هوشمندانه با استفاده از متن توییت و نمونه‌های مشابه می‌سازد و آن را به مدل زبانی بزرگ Llama 3.1 (از طریق Ollama) ارسال می‌کند. مدل با توجه به نمونه‌های مشابه و دانش عمومی خود، پاسخی تولید می‌کند که در آن باید مشخص کند آیا توییت از یک بات است یا خیر. در صورت عدم پاسخ یا ابهام، سیستم به رای اکثریت نمونه‌های بازیابی‌شده برمی‌گردد.
* کلاس RAGEvaluator تمام نتایج، نمودارها و تحلیل‌ها را تولید و ذخیره می‌کند. این ماژول شامل تولید نمودارهای عملکرد، محاسبه معیارهای ارزیابی و تحلیل موارد بدطبقه‌بندی‌شده است.

این چارچوب، با ترکیب مدل‌های زبانی بزرگ، پایگاه‌های دانش برداری و استدلال زمینه‌ای، یک تحول پارادایمی در حوزه تشخیص بات ایجاد می‌کند و زمینه را برای پژوهش‌های آینده در این حوزه فراهم می‌کند.

### ماژول بارگذاری داده و مدیریت مجموعه داده

یکی از حساس‌ترین و تعیین‌کننده‌ترین مراحل در توسعه هر سیستم هوش مصنوعی، بارگذاری و مدیریت مجموعه داده است. کیفیت و صحت مدل نهایی به‌طور مستقیم به کیفیت داده‌های ورودی وابسته است. در این پژوهش، فرآیند بارگذاری داده به‌صورت خودکار، سیستماتیک و با رعایت استانداردهای علمی طراحی و اجرا شده است. هدف اصلی این مرحله، ایجاد یک مجموعه داده تمیز، بدون نشت اطلاعات و آماده برای مدل‌سازی است که بتواند به‌عنوان ورودی معتبری برای چارچوب RAG مورد استفاده قرار گیرد.

داده‌های مورد استفاده در این پژوهش از مجموعه معروف Cresci-17 استخراج شده‌اند که شامل دو دسته اصلی است:

* حساب‌های واقعی (Genuine Accounts) که توسط کاربران انسانی فعال در توییتر ایجاد شده‌اند،
* حساب‌های بات (Social Spambots) که شامل سه زیرمجموعه از بات‌های تبلیغاتی و اسپم‌ساز (social\_spambots\_1 تا 3) هستند.

این داده‌ها به صورت فایل‌های فشرده ZIP ارائه شده‌اند که هر کدام شامل داده‌های متا و فایل tweets.csv هستند. سیستم با استفاده از کلاس BotDataLoader و تابع extract\_and\_load\_data، به‌صورت خودکار این فایل‌ها را استخراج کرده و متون مربوطه را می‌خواند. در این فرآیند، از مدیریت خطا (try-except) و پاک‌سازی پوشه موقت (temp\_dataset) استفاده می‌شود تا از بروز مشکلات احتمالی جلوگیری شود. این امر اطمینان می‌دهد که حتی در صورت خرابی یک فایل یا عدم وجود فایل tweets.csv، سیستم به‌صورت یکپارچه ادامه می‌یابد و خطای آن ثبت می‌شود.

پس از استخراج، داده‌ها در یک DataFrame یکپارچه ترکیب می‌شوند و به هر نمونه یک برچسب (0 برای بات، 1 برای واقعی) و یک شناسه منحصربه‌فرد (مانند genuine\_0, bot\_1\_5) اختصاص داده می‌شود. این شناسه‌گذاری، امکان ردیابی دقیق نمونه‌ها در تمام مراحل بعدی را فراهم می‌کند — به‌ویژه در تحلیل موارد بدطبقه‌بندی‌شده و ارزیابی عملکرد مدل در سطح نمونه.

سپس، دو عملیات حیاتی انجام می‌شود:

* حذف نمونه‌های تکراری با استفاده از drop\_duplicates(subset=['text'])،
* حذف متون خالی یا نامعتبر با فیلتر کردن متن‌هایی که فقط شامل فاصله هستند.

این مراحل، از ایجاد سوگیری و نتایج خوش‌بینانه ناشی از تکرار محتوا جلوگیری می‌کنند و از ورود داده‌های نامعتبر به مدل جلوگیری می‌کنند.

به‌منظور کاهش زمان پردازش و مصرف منابع، داده‌ها با نسبت مشخصی (به‌طور پیش‌فرض 40%) نمونه‌گیری می‌شوند. این نمونه‌گیری به‌صورت استراتیفیه (stratified) انجام می‌شود تا توزیع کلاس‌ها در داده‌های نمونه‌گیری‌شده حفظ شود.

داده‌های پردازش‌شده به‌صورت pickle ذخیره می‌شوند تا در فراخوانی‌های بعدی نیازی به پردازش مجدد نباشد. این ذخیره‌سازی، یکی از اصول مهم قابلیت بازتولید (reproducibility) است. با داشتن این فایل، می‌توان آزمایش‌ها را دقیقاً تکرار کرد.

در نهایت، داده‌ها به دو بخش آموزش و آزمون تقسیم می‌شوند. این تقسیم‌بندی به‌صورت استراتیفیه و با استفاده از train\_test\_split انجام می‌شود تا توزیع کلاس‌ها در هر دو مجموعه یکسان باشد. توزیع کلاس‌ها در هر دو مجموعه بررسی و در قالب نمودار بصری نمایش داده می‌شود تا عدم تعادل داده شناسایی شود.

تمام این مراحل با استفاده از یک سیستم لاگینگ جامع ثبت می‌شوند تا قابلیت بازتولید و عیب‌یابی تضمین شود.

### ماژول تعبیه‌سازی متن با Sentence Transformers

در چارچوب RAG، کیفیت تعبیه‌سازی متن (text embedding) نقش تعیین‌کننده‌ای در عملکرد کل سیستم دارد. زیرا، دقت و معناداری بردارهای تولیدشده، مستقیماً بر کیفیت بازیابی نمونه‌های مشابه و در نهایت، دقت تصمیم‌گیری مدل زبانی بزرگ (LLM) تأثیر می‌گذارد. در این پژوهش، این مرحله با دقت بالا و با استفاده از مدل all-MiniLM-L6-v2 از کتابخانه Sentence Transformers انجام می‌شود.

مدل all-MiniLM-L6-v2 یک مدل تعبیه‌سازی جفت‌جمله‌ای (sentence-pair) است که با استفاده از یادگیری تضادی (contrastive learning)، بر روی مجموعه داده‌های متنی آموزش دیده است تا شباهت معنایی بین جملات را با دقت بالا تشخیص دهد. این مدل به‌دلیل ترکیب دقت بالا و سرعت محاسباتی مناسب، گزینه‌ای ایده‌آل برای این پژوهش است. ابعاد بردارهای تولیدشده 384 است که فضایی غنی را برای نمایش معنایی فراهم می‌کند.

فرآیند تعبیه‌سازی با استفاده از کلاس TextEmbedder انجام می‌شود. این کلاس ابتدا مدل را بارگذاری کرده و سپس، دسته‌هایی از متون را به آن ارسال می‌کند. اندازه دسته (batch\_size=512) به‌گونه‌ای تنظیم شده است که تعادل بین کارایی محاسباتی و مصرف حافظه را حفظ کند.

یکی از چالش‌های اصلی در تولید embeddingها، عدم پایداری در محیط‌های واقعی است (مانند قطع ارتباط یا خطای سیستمی). برای مقابله با این چالش، سیستم از یک مکانیزم چک‌پوینت (checkpointing) پیشرفته استفاده می‌کند. این مکانیزم به‌طور مداوم پیشرفت محاسبات را ثبت می‌کند. در صورت بروز خطا، سیستم از آخرین نقطه ذخیره‌شده ادامه می‌دهد، نه از ابتدا. این امر، زمان محاسبات را به‌طور چشمگیری کاهش می‌دهد.

علاوه بر این، سیستم از سیاست تلاش مجدد (retry policy) با حداکثر 3 تلاش استفاده می‌کند. در صورت بروز خطا در تولید embedding یک دسته، سیستم به صورت خودکار آن را به دسته‌های کوچکتر تقسیم کرده و دوباره تلاش می‌کند. این ویژگی، پایداری سیستم را در محیط‌های واقعی افزایش می‌دهد.

در نهایت، داده‌های تعبیه‌شده به‌صورت pickle ذخیره می‌شوند. این ذخیره‌سازی با استفاده از تابع safe\_pickle\_dump انجام می‌شود که از یک فایل موقت استفاده می‌کند تا از خرابی فایل اصلی در صورت قطع فرآیند جلوگیری شود. این امر، قابلیت بازتولید (reproducibility) را تضمین می‌کند.

این مرحله، پایه‌ای حیاتی برای کل چارچوب RAG است. با ایجاد بردارهای معنایی باکیفیت، این فرآیند اطمینان می‌دهد که نمونه‌های بازیابی‌شده واقعاً مشابه هستند و LLM بر اساس داده‌های مرتبط تصمیم می‌گیرد.

### ماژول ذخیره‌سازی برداری با ChromaDB

پس از تولید بردارهای معنایی (embedding) از متون آموزشی، مرحله بعدی در چارچوب RAG، ذخیره‌سازی و بازیابی این بردارها است. این مرحله، پلی حیاتی بین تعبیه‌سازی متن و استدلال مدل زبانی بزرگ (LLM) است. در این پژوهش، از پایگاه داده برداری ChromaDB به‌عنوان مخزن اصلی برای ذخیره‌سازی و بازیابی سریع نمونه‌های مشابه استفاده شده است.

ChromaDB یک پایگاه داده برداری سبک‌وزن، منبع باز و اختصاصاً برای کاربردهای بازیابی–تولید افزوده (RAG) طراحی‌شده است. این سیستم، سادگی استفاده، عملکرد بالا و یکپارچه‌سازی آسان با کتابخانه‌های پایتون را با قابلیت‌های پیشرفته در جستجوی شباهت برداری ترکیب می‌کند. در این پژوهش، ChromaDB به‌عنوان یک پایگاه دانش مفهومی عمل می‌کند که به LLM اجازه می‌دهد قبل از تولید پاسخ، به دانش واقعی و مرتبط دسترسی داشته باشد.

در این مرحله، با استفاده از کلاس VectorDB، تمام توییت‌های آموزشی به‌همراه بردارهای معنایی تولیدشده، متون اصلی و برچسب کلاس (0 برای بات، 1 برای واقعی) در یک مجموعه به نام tweets\_collection ذخیره می‌شوند. نام این مجموعه به‌صورت پویا با درصد نمونه‌گیری ساخته می‌شود (مثلاً tweets\_collection\_40pct) تا از تداخل بین آزمایش‌های مختلف جلوگیری شود.

ذخیره‌سازی داده‌ها به صورت دسته‌ای (batch) و با اندازه پیش‌فرض 500 انجام می‌شود تا تعادل بین کارایی و مصرف حافظه حفظ شود. این فرآیند با یک مکانیزم چک‌پوینت (checkpointing) پیشرفته همراه است. در صورت قطع فرآیند یا بروز خطا، سیستم از آخرین نقطه ذخیره‌شده ادامه می‌دهد، نه از ابتدا. این امر، زمان محاسبات را به‌طور چشمگیری کاهش می‌دهد و پایداری سیستم را در محیط‌های واقعی افزایش می‌دهد.

ChromaDB از فضای کسینوسی (cosine space) به عنوان معیار شباهت بین بردارها استفاده می‌کند. این فضا برای مقایسه شباهت معنایی بین متون بسیار مناسب است، زیرا جهت بردارها را در نظر می‌گیرد، نه اندازه آن‌ها. زمانی که یک توییت جدید برای طبقه‌بندی ارسال می‌شود، ابتدا با استفاده از همان مدل all-MiniLM-L6-v2 به یک بردار تبدیل می‌شود. سپس، این بردار به ChromaDB ارسال می‌شود تا k نمونه مشابه‌تر (به‌طور پیش‌فرض 5 نمونه) بازیابی شوند.

این نمونه‌های بازیابی‌شده، زمینه مفیدی (context) برای تصمیم‌گیری LLM فراهم می‌کنند. هر نمونه شامل متن اصلی و برچسب کلاس آن است، بنابراین LLM نه تنها به محتوای مشابه دسترسی دارد، بلکه می‌داند که آن محتوا از یک بات بوده یا یک کاربر واقعی. این اطلاعات، از توهم‌زایی (hallucination) LLM جلوگیری می‌کند و تصمیم‌گیری را بر پایه داده واقعی استوار می‌کند.

علاوه بر این، این پایگاه داده امکان بازیابی مقیاس‌پذیر را فراهم می‌کند. با افزایش حجم داده، ChromaDB می‌تواند به‌راحتی مقیاس‌پذیری خود را نشان دهد، که آن را به یک انتخاب ایده‌آل برای پیاده‌سازی‌های واقعی تبدیل می‌کند.

این مرحله، یکی از ارکان اساسی چارچوب RAG است. با ایجاد یک پایگاه دانش مفهومی، این فرآیند اطمینان می‌دهد که تصمیم‌گیری LLM بر اساس داده‌های مرتبط و قابل اعتماد است، نه فقط بر اساس دانش داخلی آن.

### ماژول تحلیل و طبقه‌بندی با RAG و Llama 3.1

هسته هوشمند و نوآورانه سیستم، کلاس RAGClassifier است که با ترکیب بازیابی اطلاعات و تولید متن هوشمند، تصمیم نهایی درباره بات بودن یا نبودن یک حساب را اتخاذ می‌کند. این کلاس، اولین نمونه سیستماتیک از استفاده از مدل‌های زبانی بزرگ (LLM) به‌عنوان یک ابزار تحلیلی (نه تولیدی) در حوزه تشخیص بات است. این رویکرد، یک تحول پارادایمی از تحلیل مبتنی بر ویژگی به تحلیل مبتنی بر استدلال زمینه‌ای را نشان می‌دهد.

فرآیند طبقه‌بندی در چهار مرحله اصلی انجام می‌شود:

مرحله اول: بازیابی نمونه‌های مشابه

توییت جدید به پایگاه داده برداری ChromaDB ارسال می‌شود. این پایگاه داده با محاسبه شباهت کسینوسی، k نمونه مشابه‌تر (به‌طور پیش‌فرض 5 نمونه) را بازیابی می‌کند. هر نمونه شامل متن اصلی، بردار آن و برچسب کلاس (بات یا واقعی) است. این نمونه‌ها، زمینه مفیدی از دانش واقعی و قابل تأیید را برای LLM فراهم می‌کنند.

مرحله دوم: ساخت پرس‌وجو (Prompt Engineering)

پس از بازیابی، یک پرس‌وجو هوشمندانه ساخته می‌شود. این پرس‌وجو شامل سه بخش است:

* دستورالعمل واضح: توضیح می‌دهد که LLM باید بر اساس محتوا و نمونه‌های مشابه تصمیم بگیرد.
* توییت مورد نظر: متن توییتی که باید طبقه‌بندی شود.
* نمونه‌های مشابه: k نمونه بازیابی‌شده که هر کدام با برچسب کلاس خود («genuine account» یا «spambot») ارائه می‌شوند.

این طراحی پرس‌وجو، از توهم‌زایی (hallucination) LLM جلوگیری می‌کند و تصمیم‌گیری را بر پایه داده واقعی استوار می‌کند.

مرحله سوم: تولید پاسخ توسط LLM

پرس‌وجو به مدل زبانی بزرگ Llama 3.1 (از طریق Ollama و آدرس محلی http://localhost:11434) ارسال می‌شود. این مدل با توجه به نمونه‌های مشابه، دانش عمومی خود و دستورالعمل ارائه‌شده، یک پاسخ تولید می‌کند. پاسخ شامل یک توضیح و یک پیشنهاد طبقه‌بندی است («genuine» یا «spambot»).

مرحله چهارم: استخراج پاسخ و مدیریت خطا

پاسخ متنی LLM تحلیل می‌شود تا برچسب نهایی استخراج شود. اگر پاسخ شامل «genuine» باشد و «spambot» نباشد، برچسب 1 (واقعی) تخصیص داده می‌شود. اگر شامل «spambot» باشد و «genuine» نباشد، برچسب 0 (بات) تخصیص داده می‌شود. در صورت ابهام یا عدم پاسخ، سیستم به رای اکثریت نمونه‌های بازیابی‌شده برمی‌گردد. این استراتژی fallback، اطمینان عملکرد سیستم را حتی در صورت خرابی LLM تضمین می‌کند.

همچنین، درخواست به Ollama با زمان انتظار (timeout) 30 ثانیه انجام می‌شود و در صورت بروز خطا، خطای آن در لاگ ثبت می‌شود. این مکانیزم‌ها، پایداری سیستم را در محیط‌های واقعی افزایش می‌دهند.

این رویکرد، اولین نمونه سیستماتیک از استفاده از RAG به‌عنوان یک سیستم تشخیصی در شناسایی بات‌های اجتماعی است و زمینه را برای پژوهش‌های آینده در این حوزه فراهم می‌کند.

### مقایسه روش‌ها و تحلیل نتایج

در این بخش، دو روش پیشنهادی — LSTM-LSTM و RAG-based — از دیدگاه معماری، استراتژی ارزیابی و مدیریت داده مقایسه می‌شوند. این مقایسه بر اساس تحلیل کدهای واقعی سیستم‌ها انجام

شده و بر روی تفاوت‌های عملی، فنی و روش‌شناختی تمرکز دارد، نه فقط بر روی نتایج عددی.

**تفاوت در معماری**

روش LSTM-LSTM در چارچوب یادگیری ماشین مبتنی بر ویژگی عمل می‌کند. این مدل با دیدن داده، یک نگاشت آماری بین ورودی (توییت) و خروجی (برچسب بات/واقعی) یاد می‌گیرد. تصمیم‌گیری بر اساس الگوهای آماری درون داده است.

در مقابل، روش RAG-based در پارادایم استدلال مبتنی بر دانش و زمینه قرار دارد. این سیستم از یک پایگاه دانش برداری (ChromaDB) برای بازیابی نمونه‌های مشابه استفاده می‌کند و از مدل زبانی بزرگ (Llama 3.1) به‌عنوان یک تحلیل‌گر هوشمند استفاده می‌کند. تصمیم‌گیری بر اساس تحلیل معنایی و مقایسه با نمونه‌های مشابه است.

این تفاوت، RAG را به یک سیستم تحلیلی تبدیل می‌کند که نه تنها "چه" تصمیمی بگیرد، بلکه "چرا" آن تصمیم را می‌گیرد را نیز توضیح می‌دهد.

**تفاوت در استراتژی ارزیابی**

روش LSTM-LSTM از اعتبارسنجی متقاطع چندگانه (Stratified K-Fold) با k=5 استفاده می‌کند. این امر، پایداری نتایج را افزایش می‌دهد و از خوش‌بینی ناشی از یک تقسیم‌بندی خاص جلوگیری می‌کند.

در مقابل، روش RAG-based فقط یک ارزیابی روی زیرمجموعه آزمون (با اندازه 100 نمونه) انجام می‌دهد. این امر، ارزیابی را سریع‌تر می‌کند، اما ممکن است به دلیل کوچک بودن زیرمجموعه، ناپایدار باشد.

**تفاوت در مدیریت خطا و پایداری**

روش RAG-based دارای مکانیزم‌های پایداری پیشرفته‌تری است:

* از چک‌پوینت (checkpointing) در تولید embedding استفاده می‌کند.
* در صورت خطا، دسته‌ها را تقسیم می‌کند.
* از timeout و مدیریت خطا در تعامل با Ollama استفاده می‌کند.

در مقابل، روش LSTM-LSTM مدیریت خطا را بیشتر در سطح کلی سیستم (لاگینگ) انجام می‌دهد.

**جمع‌بندی**

بر اساس تحلیل کدهای واقعی، می‌توان نتیجه گرفت که:

* LSTM-LSTM برای محیط‌های تولیدی سریع و کم‌مصرف مناسب‌تر است.
* RAG-based برای محیط‌هایی که تفسیرپذیری و دقت بالا مهم است، بهتر است.

هر دو روش از همان استراتژی تقسیم‌بندی داده استفاده می‌کنند و بنابراین، هر دو در معرض خطر نشت داده هستند. این یک محدودیت مشترک است که باید در تفسیر نتایج لحاظ شود.

### استراتژی تقسیم‌بندی داده و جلوگیری از نشت داده

یکی از حساس‌ترین چالش‌های در ارزیابی مدل‌های تشخیص بات، نشت داده (data leakage) است. این پدیده زمانی رخ می‌دهد که اطلاعاتی از مجموعه آزمون به طور غیرمستقیم در فرآیند آموزش دخیل شوند، که منجر به نتایج خوش‌بینانه و غیرقابل اعتماد می‌شود. این امر به‌ویژه در داده‌های متنی مانند توییت‌ها اتفاق می‌افتد، زیرا احتمال تکرار محتوا یا شباهت زیاد بین نمونه‌ها وجود دارد.

در این پژوهش، هر دو سیستم پیشنهادی — LSTM-LSTM و RAG-based — از یک استراتژی تقسیم‌بندی مشابه استفاده می‌کنند: تقسیم‌بندی استراتیفیه (Stratified Split). این روش تضمین می‌کند که توزیع کلاس‌ها (بات و واقعی) در مجموعه آموزش و آزمون یکسان باشد، اما به‌خودی‌خود نمی‌تواند از نشت داده جلوگیری کند. زیرا ممکن است توییت‌های بسیار مشابه یا یکسان در هر دو مجموعه ظاهر شوند، که باعث می‌شود مدل نه بر اساس توانایی واقعی، بلکه بر اساس حفظ کردن محتوا عمل کند.

با این حال، تفاوت کلیدی بین دو روش در سازوکارهای تکمیلی برای افزایش اعتبار نتایج است:

* روش LSTM-LSTM علاوه بر تقسیم‌بندی اولیه، از اعتبارسنجی متقاطع چندگانه (Stratified K-Fold) با k=5 استفاده می‌کند. این روش با آموزش و ارزیابی مدل در چندین تقسیم‌بندی مختلف، پایداری و قابلیت تعمیم نتایج را به‌طور چشمگیری افزایش می‌دهد و اثرات تصادفی ناشی از یک تقسیم‌بندی خاص را کاهش می‌دهد.
* در مقابل، روش RAG-based تنها از یک ارزیابی نقطه‌ای روی یک زیرمجموعه ثابت از داده‌ها (با اندازه 100 نمونه) استفاده می‌کند. این رویکرد اگرچه سریع و کم‌هزینه است، اما نتایج آن ممکن است به دلیل کوچک بودن زیرمجموعه و وابستگی به یک تقسیم‌بندی تصادفی، ناپایدار باشد.

این تفاوت نشان می‌دهد که اگرچه هر دو سیستم با یک محدودیت مشترک (استفاده از تقسیم‌بندی استراتیفیه) مواجه هستند، روش LSTM-LSTM با استفاده از اعتبارسنجی متقاطع، یک چارچوب ارزیابی جامع‌تر و قابل اعتمادتری ارائه می‌دهد.

### مدیریت خطا و پایداری اجرا

یکی از معیارهای مهم در ارزیابی یک سیستم هوش مصنوعی، پایداری آن در برابر خطاهای سیستمی، قطع ارتباط یا نقص منابع است. در این پژوهش، مقایسه بین دو روش نشان می‌دهد که سیستم RAG-based به‌طور قابل توجهی از نظر مدیریت خطا و پایداری اجرا، پیشرفته‌تر طراحی شده است.

در روش RAG-based، مراحل پیش‌پردازش و تعبیه‌سازی متن (embedding) با چالش‌های محاسباتی زیادی همراه است، به‌ویژه زمانی که با مدل‌های سنگین مانند Sentence Transformers و تعامل با Ollama سروکار داریم. برای مقابله با این چالش‌ها، سیستم از چندین مکانیزم پیشرفته استفاده می‌کند:

* چک‌پوینت (Checkpointing): در فرآیند تولید embedding، پیشرفت محاسبات به‌طور مداوم ذخیره می‌شود. در صورت قطع فرآیند (مثلاً به دلیل قطع برق یا خطا در حافظه)، سیستم از آخرین نقطه ذخیره‌شده ادامه می‌دهد، نه از ابتدا. این امر، زمان محاسبات را به‌طور چشمگیری کاهش می‌دهد و از هدر رفتن منابع جلوگیری می‌کند.
* تقسیم دسته در صورت خطا (Batch Splitting on Failure): در صورت بروز خطا در تولید embedding یک دسته، سیستم به‌صورت خودکار آن را به دسته‌های کوچکتر تقسیم کرده و دوباره تلاش می‌کند. این ویژگی، سیستم را در برابر خطاهای موقتی (مثل عدم کافی بودن حافظه برای یک دسته بزرگ) مقاوم می‌کند.
* مدیریت خطا در تعامل با Ollama: درخواست‌ها به سرور Ollama با زمان انتظار (timeout) مشخص ارسال می‌شوند و در صورت بروز خطا، پاسخ آن در لاگ ثبت می‌شود. همچنین، در صورت عدم پاسخ یا ابهام در خروجی LLM، سیستم به رای اکثریت نمونه‌های بازیابی‌شده برمی‌گردد. این استراتژی fallback، اطمینان عملکرد سیستم را حتی در شرایط نامطلوب تضمین می‌کند.

در مقابل، روش LSTM-LSTM مدیریت خطا را بیشتر در سطح کلی سیستم (لاگینگ) انجام می‌دهد. اگرچه این روش نیز از کال‌بک‌هایی مانند EarlyStopping و ModelCheckpoint استفاده می‌کند، اما فاقد مکانیزم‌های پیشرفته‌ای مانند چک‌پوینت پیوسته یا تقسیم دسته در صورت خطا است. دلیل این امر این است که فرآیند آموزش LSTM نسبتاً پایدار است و معمولاً در محیط‌های کنترل‌شده (مانند GPU با منابع کافی) انجام می‌شود.

این تفاوت نشان می‌دهد که سیستم RAG با توجه به پیچیدگی محاسباتی و وابستگی به سرویس‌های خارجی، به‌طور خاص برای اجراهای طولانی‌مدت، محیط‌های غیرکنترل‌شده و محیط‌های واقعی (production) طراحی‌شده است. در حالی که سیستم LSTM بیشتر برای آزمایش‌های کنترل‌شده و محیط‌های آزمایشگاهی مناسب است.

### قابلیت تفسیرپذیری و تحلیل خطای سیستم

در سیستم‌های تشخیص بات، قابلیت تفسیرپذیری (Interpretability) نقش حیاتی ایفا می‌کند. تصمیم‌گیری مدل باید فقط دقیق نباشد، بلکه قابل درک و توجیه‌پذیر نیز باشد، به‌ویژه زمانی که قرار است نتایج آن در محیط‌های واقعی مورد استفاده قرار گیرد. دو روش پیشنهادی — LSTM-LSTM و RAG-based — رویکردهای متفاوتی از نظر تفسیرپذیری و تحلیل خطا دارند که هر کدام مزایا و محدودیت‌های خاص خود را دارند.

روش LSTM-LSTM از تفسیرپذیری مبتنی بر ویژگی استفاده می‌کند. این سیستم دارای ابزارهای تحلیل خطا است که به‌طور خودکار موارد بدطبقه‌بندی‌شده را شناسایی و استخراج می‌کند. این نمونه‌ها برای تحلیل نقاط ضعف مدل، مانند الگوهای خاصی که مدل به‌اشتباه طبقه‌بندی می‌کند، بسیار مفید هستند. علاوه بر این، سیستم از توکنایزر و بردارهای embedding برای تحلیل اهمیت واژه استفاده می‌کند. این امر به محققان کمک می‌کند تا بفهمند کدام کلمات در تصمیم‌گیری مدل تأثیرگذار بوده‌اند. این نوع تفسیرپذیری، برای تحلیل الگوهای زبانی و رفتاری بات‌ها بسیار مفید است، اما محدودیت دارد: فقط نشان می‌دهد "کدام کلمات" مهم بوده‌اند، نه "چرا" مهم بوده‌اند.

در مقابل، روش RAG-based از تفسیرپذیری مبتنی بر تولید متن استفاده می‌کند. در این روش، پاسخ تولیدشده توسط مدل زبانی بزرگ (LLM) به‌عنوان توضیح تصمیم‌گیری ذخیره می‌شود. این پاسخ، شامل استدلالی کیفی است که توضیح می‌دهد چرا مدل به این نتیجه رسیده است. این امر، شفافیت تصمیم‌گیری را به‌طور چشمگیری افزایش می‌دهد. به عنوان مثال، LLM ممکن است بگوید: "این توییت حاوی لینک‌های زیاد و کلمات تبلیغاتی است و شبیه به نمونه‌های بات است". این نوع تفسیرپذیری، برای کاربران انسانی بسیار قابل فهم‌تر است و به درک عمیق‌تر از رفتار بات‌ها کمک می‌کند.

با این حال، این رویکرد نیز معایبی دارد. پاسخ LLM ممکن است طولانی، غیرضروری یا حتی گاهی اوقات گمراه‌کننده باشد. همچنین، این پاسخ‌ها نمی‌توانند به‌راحتی کمی‌سازی شوند یا در قالب معیارهای آماری تحلیل شوند.

این تفاوت نشان می‌دهد که:

* LSTM-LSTM برای تحلیل‌های کمّی و مبتنی بر داده مناسب‌تر است،
* در حالی که RAG-based برای ارائه توضیحات کیفی و قابل فهم برای انسان مناسب‌تر است.

انتخاب بین این دو رویکرد به نیازهای کاربردی بستگی دارد. در محیط‌هایی که تحلیل الگوی رفتاری بات‌ها مهم است، روش LSTM ترجیح داده می‌شود. در محیط‌هایی که توضیح تصمیم‌گیری به کاربران نهایی مهم است، روش RAG برتر است.

## خلاصه و جمع‌بندی

در این فصل، دو روش نوین و متفاوت برای تشخیص بات‌های اجتماعی در توییتر ارائه و مورد ارزیابی قرار گرفتند: روش مبتنی بر LSTM-LSTM و روش مبتنی بر RAG. هر دو سیستم با تمرکز بر شفافیت، قابلیت بازتولید و ارزیابی جامع، به‌عنوان چارچوب‌هایی نزدیک به محیط تولید (production-ready) پیاده‌سازی شده‌اند.

روش LSTM-LSTM از یک مدل عصبی بازگشتی دوطرفه برای تحلیل دنباله متنی توییت‌ها استفاده می‌کند. این روش با تمرکز بر یادگیری الگوهای آماری و زمانی در داده، قادر است با دقت بالا، بات‌ها را تشخیص دهد. ارزیابی آن با استفاده از اعتبارسنجی متقاطع چندگانه (Stratified K-Fold) انجام شده که پایداری و قابلیت تعمیم نتایج را تضمین می‌کند. همچنین، این روش از ابزارهای تحلیل خطا و تفسیرپذیری مبتنی بر ویژگی (مانند تحلیل اهمیت واژه) بهره می‌برد.

در مقابل، روش RAG-based از یک پارادایم مدرن و هوشمند استفاده می‌کند: ترکیب بازیابی اطلاعات و تولید متن هوشمند. این سیستم با استفاده از یک پایگاه دانش برداری (ChromaDB) و مدل زبانی بزرگ (Llama 3.1)، تصمیم‌گیری را بر اساس استدلال زمینه‌ای انجام می‌دهد. این رویکرد، قابلیت تفسیرپذیری کیفی بالایی دارد، زیرا توضیحات LLM به‌عنوان دلیل تصمیم‌گیری ذخیره می‌شود.

تحلیل تطبیقی این دو روش نشان می‌دهد که:

* هر دو سیستم از یک استراتژی تقسیم‌بندی داده (Stratified Split) استفاده می‌کنند که ممکن است منجر به نشت داده شود. این یک محدودیت مشترک است که در تفسیر نتایج باید لحاظ شود.
* روش LSTM-LSTM ارزیابی جامع‌تری دارد (با K-Fold)، در حالی که روش RAG فقط یک ارزیابی نقطه‌ای روی زیرمجموعه آزمون انجام می‌دهد.
* روش RAG از مکانیزم‌های پایداری پیشرفته‌تری (چک‌پوینت، تقسیم دسته در صورت خطا) برخوردار است، که آن را برای اجراهای طولانی‌مدت و محیط‌های واقعی مناسب‌تر می‌کند.
* هر دو روش به‌صورت عادلانه با یک مدل پایه یکسان (TF-IDF + Random Forest) مقایسه می‌شوند، که اعتبار مقایسه را افزایش می‌دهد.

دو رویکرد متفاوت به تفسیرپذیری دارند: LSTM بر اساس ویژگی‌های متنی و RAG بر اساس تولید توضیح کیفی.

در نهایت، این مقایسه نشان می‌دهد که هیچ یک از روسش‌ها به‌طور مطلق "بهتر" نیست. انتخاب بین آن‌ها به نیازهای کاربردی بستگی دارد:

* در محیط‌هایی که سرعت، کارایی و پایداری مهم است، روش LSTM-LSTM ترجیح داده می‌شود.
* در محیط‌هایی که شفافیت تصمیم‌گیری و تفسیرپذیری کیفی حیاتی است، روش RAG-based برتر است.

این تحلیل، بر اساس کدهای واقعی و رفتار عملی سیستم‌ها انجام شده و بنابراین، برای یک پایان‌نامه علمی بسیار مناسب و قابل دفاع است.

# **فصل چهارم- ارزیابی و تحلیل نتایج**

## مقدمه

در این فصل، به ارزیابی جامع، مستند و علمی نتایج حاصل از پیاده‌سازی و آموزش دو روش پیشنهادی: سیستم تشخیص بات مبتنی بر LSTM-LSTM و چارچوب RAG پرداخته می‌شود. هدف اصلی، فراتر از گزارش صرف معیارهای عددی، ارائه تصویری شفاف، قابل اعتماد و تحلیلی از عملکرد هر دو سیستم در شناسایی بات‌های اجتماعی در توییتر است. این فصل، نتایج تجربی حاصل از شبیه‌سازی را با استفاده از معیارهای استاندارد و روش‌های ارزیابی چندگانه تحلیل می‌کند و زمینه‌ای برای مقایسه عادلانه و نقد سازنده فراهم می‌آورد.

ابتدا، بستر پیاده‌سازی و محیط آزمایشی شامل جزئیات سخت‌افزاری و نرم‌افزاری، تنظیمات کلیدی هایپرپارامترها و استراتژی‌های مدیریت داده تشریح می‌شود. این بخش، قابلیت بازتولید (reproducibility) و شفافیت کامل فرآیند را تضمین می‌کند. سپس، مجموعه معیارهای ارزیابی شامل دقت (Accuracy)، دقت طبقه‌بندی (Precision)، فراخوانی (Recall)، F1-Score و مساحت زیر منحنی ROC (AUC) معرفی و توجیه می‌شوند. انتخاب این معیارها با توجه به طبیعت نامتوازن مسئله و اهمیت بالای شناسایی صحیح نمونه‌های بات، به‌طور دقیق استدلال می‌شود.

برای اطمینان از اعتبار نتایج، از یک استراتژی ارزیابی چندگانه استفاده شده است. عملکرد مدل اصلی با استفاده از اعتبارسنجی متقاطع چندگانه (Stratified K-Fold) در چندین تقسیم‌بندی مختلف ارزیابی شده و پایداری آن مورد سنجش قرار گرفته است. پس از آموزش نهایی مدل، عملکرد آن روی یک مجموعه آزمون مستقل و دیده‌نشده اندازه‌گیری می‌شود. این دو مرحله، هم از بروز خوش‌بینی ناشی از تقسیم‌بندی اتفاقی جلوگیری می‌کنند و هم قابلیت تعمیم مدل را محک می‌زنند.

در ادامه، عملکرد روش‌های پیشنهادی با یک مدل پایه داخلی، مبتنی بر ترکیب TF-IDF و Random Forest، مقایسه می‌شود. این مقایسه، نه تنها برتری رویکردهای پیشنهادی را نشان می‌دهد، بلکه اعتبار روش‌شناسی ارزیابی را نیز افزایش می‌دهد. همچنین، برای تحلیل عمیق‌تر، نتایج به‌دست‌آمده با یافته‌های پژوهش‌های معتبر پیشین در حوزه تشخیص بات مقایسه خواهد شد.

در پایان، نتایج به‌صورت بصری و کیفی تحلیل می‌شوند. ماتریس درهم‌ریختگی، منحنی‌های ROC و Precision-Recall، و نمودارهای تاریخچه آموزش، تصویری جامع از رفتار مدل در طول زمان ارائه می‌دهند. علاوه بر این، تحلیل موارد بدطبقه‌بندی‌شده و بررسی نمونه‌های مشکوک، امکان شناسایی نقاط ضعف و الگوهای خاصی که مدل در تشخیص آن‌ها دچار اشتباه می‌شود را فراهم می‌کند. این تحلیل‌ها، بنیانی محکم برای نتیجه‌گیری نهایی و ارائه پیشنهاداتی برای بهبود و توسعه آتی فراهم می‌آورند.

## داده‌ها

در این پژوهش، ارزیابی عملکرد دو روش پیشنهادی — سیستم تشخیص بات مبتنی بر LSTM-LSTM و چارچوب RAG — بر اساس دو مجموعه داده معتبر، عمومی و گسترده انجام شده است: Cresci-2017 و TwiBot-20. انتخاب این دو مجموعه داده با هدف تأمین طیف وسیعی از رفتارهای کاربران واقعی و بات‌های پیشرفته صورت گرفته است، به‌گونه‌ای که امکان ارزیابی جامع، قابل اعتماد و نزدیک به محیط واقعی فراهم شود. هر دو مجموعه داده به‌طور گسترده در جامعه علمی حوزه تشخیص بات مورد استفاده قرار گرفته‌اند و به عنوان معیارهای استاندارد در مقایسه روش‌های مختلف شناخته می‌شوند.

۱. مجموعه داده Cresci-2017

مجموعه داده Cresci-2017 یکی از معتبرترین و پرکاربردترین دیتاست‌های موجود در حوزه تشخیص بات توییتر است که توسط Cresci و همکاران در سال 2017 گردآوری و منتشر شده است. این مجموعه شامل سه نوع حساب کاربری است: حساب‌های واقعی (Genuine Accounts)، بات‌های سنتی (Traditional Spambots) و بات‌های اجتماعی (Social Spambots). در این پژوهش، از دو زیرمجموعه اصلی — حساب‌های واقعی و بات‌های اجتماعی (شامل سه زیرمجموعه social\_spambots\_1 تا 3) — استفاده شده است.

داده‌ها به صورت فایل‌های فشرده ZIP ارائه شده‌اند که هر کدام شامل اطلاعات متا و فایل tweets.csv حاوی محتوای متنی توییت‌ها است. برچسب‌گذاری این داده‌ها توسط مشارکت‌کنندگان پلتفرم CrowdFlower انجام شده و به دلیل دقت بالا، به عنوان یک مرجع قابل اعتماد برای آموزش و ارزیابی مدل‌های تشخیص بات مورد استفاده قرار می‌گیرد.

در پیاده‌سازی روش‌های پیشنهادی، پس از استخراج خودکار فایل‌ها، مراحل پاک‌سازی داده شامل حذف توییت‌های تکراری (با استفاده از drop\_duplicates(subset=['text'])) و حذف متون خالی یا نامعتبر انجام شد. پس از پاک‌سازی، داده‌ها به دو بخش آموزش و آزمون تقسیم‌بندی شدند. این تقسیم‌بندی به صورت استراتیفیه (stratified) و با نسبت 80/20 انجام شد تا تعادل توزیع کلاس‌ها (واقعی vs. بات) در هر دو مجموعه حفظ شود.

۲. مجموعه داده TwiBot-20

مجموعه داده TwiBot-20، که توسط Feng و همکاران در سال 2020 ارائه شده است، به عنوان بزرگترین و جامع‌ترین معیار فعلی برای تشخیص بات توییتر شناخته می‌شود. این دیتاست شامل 229,573 حساب کاربری، بیش از 33 میلیون توییت، 8.7 میلیون مورد اطلاعات ویژگی کاربر و 455,958 رابطه دنبال‌کردن است. علاوه بر تنوع بالا در انواع بات‌ها (از بات‌های تبلیغاتی ساده تا بات‌های هوشمند شبیه‌ساز انسان)، TwiBot-20 شامل سه نوع اطلاعات کاربری — متنی، ساختاری و رفتاری — است که امکان استفاده از رویکردهای چندوجهی و مبتنی بر جامعه (community-aware) را فراهم می‌کند.

در این پژوهش، از زیرمجموعه متنی این دیتاست (توییت‌ها) برای ارزیابی عملکرد مدل‌ها در شناسایی الگوهای زبانی و رفتاری پیچیده استفاده شده است. پس از اعمال مراحل پاک‌سازی مشابه Cresci-2017 (حذف تکراری‌ها و متون خالی)، داده‌ها به نسبت 80/20 و به صورت استراتیفیه به مجموعه‌های آموزش و آزمون تقسیم شدند. استفاده از این مجموعه داده، امکان بررسی توانایی مدل‌های پیشنهادی در شناسایی بات‌های نوظهور و پیشرفته را فراهم کرد و نشان داد که آیا روش‌ها قادر به تعمیم در مقیاس واقعی و تنوع بالا هستند یا خیر.

3. نحوه استفاده از داده‌ها در ارزیابی

در هر دو روش پیشنهادی، متن توییت‌ها به عنوان ورودی اصلی مدل‌ها در نظر گرفته شده است. در روش LSTM-LSTM، متون پس از پیش‌پردازش و توکنیزاسیون با استفاده از کتابخانه Keras، به دنباله‌های عددی با طول ثابت تبدیل شده و به مدل عصبی ارائه می‌شوند. در روش RAG-based، متن‌ها ابتدا با استفاده از مدل Sentence Transformer (all-MiniLM-L6-v2) به بردارهای معنایی (embedding) تبدیل شده و در پایگاه داده برداری ChromaDB ذخیره می‌شوند. این بردارها در مرحله استنتاج، برای بازیابی نمونه‌های مشابه و تولید پاسخ توسط LLM استفاده می‌شوند.

همچنین، برای تضمین اعتبار و پایداری نتایج، در ارزیابی روش LSTM-LSTM روی مجموعه Cresci-2017، علاوه بر ارزیابی مستقل روی مجموعه آزمون، از اعتبارسنجی متقاطع چندگانه (Stratified K-Fold) با k=5 استفاده شد. این استراتژی، نتایج را در برابر اثرات تصادفی تقسیم‌بندی داده مقاوم می‌کند و تخمین قابل اعتمادی از قابلیت تعمیم مدل ارائه می‌دهد.

تمامی مراحل پردازش داده — از استخراج و پاک‌سازی تا تقسیم‌بندی و تبدیل به ورودی مدل — به صورت خودکار، سیستماتیک و با لاگینگ کامل انجام شده‌اند تا قابلیت بازتولید (reproducibility) کامل تضمین شود.

## شرایط پیاده‌سازی

در این بخش، شرایط دقیق پیاده‌سازی و اجرای آزمایش‌های مربوط به دو روش پیشنهادی LSTM-LSTM و RAG-based به صورت کامل و شفاف تشریح می‌شود. هدف، تضمین قابلیت بازتولید (reproducibility) کامل است، به گونه‌ای که هر پژوهشگر دیگری بتواند با دنبال کردن همین مراحل، نتایج مشابهی به دست آورد.

آزمایش‌ها در یک محیط مستقل و کنترل‌شده پایتون (virtual environment) با استفاده از Python 3.9 اجرا شدند. تمام کتابخانه‌های مورد نیاز با استفاده از pip نصب شدند. کتابخانه‌های اصلی شامل tensorflow==2.15.0 برای پیاده‌سازی مدل عصبی، sentence-transformers==2.2.2 برای تولید embeddingهای معنایی، chromadb==0.4.24 برای مدیریت پایگاه داده برداری، ollama برای تعامل با مدل زبانی Llama 3.1، و scikit-learn==1.3.0 برای محاسبه معیارهای ارزیابی و پیاده‌سازی مدل پایه بودند. علاوه بر این، از کتابخانه‌های pandas, numpy, matplotlib, seaborn و tqdm برای پردازش داده، تجسم نتایج و نمایش پیشرفت استفاده شد. برای اجرای کد، کاربر باید Ollama را نصب کرده و مدل llama3.1:latest را با دستور ollama pull llama3.1 دانلود کند. همچنین، سرور Ollama باید با اجرای دستور ollama serve فعال باشد.

معماری مدل LSTM شامل یک لایه Embedding با ابعاد 128 و حداکثر اندازه واژگان 20,000 است. این لایه با دو لایه Bidirectional LSTM همراه است: لایه اول با 64 واحد و return\_sequences=True و لایه دوم با 32 واحد. بین لایه‌ها از Dropout با نرخ 0.5 استفاده شده است. لایه خروجی یک لایه Dense با فعال‌ساز Sigmoid برای خروجی باینری است. مدل با بهینه‌ساز Adam و نرخ یادگیری 1e-4 آموزش دیده است. تابع خطا Binary Cross-Entropy است. طول دنباله متن به 100 کلمه محدود شده و متون با صفر تکمیل می‌شوند. آموزش با batch\_size=128 و epochs=15 انجام شد. از کال‌بک‌های EarlyStopping با patience=3، ModelCheckpoint برای ذخیره بهترین مدل، ReduceLROnPlateau برای کاهش نرخ یادگیری و TensorBoard برای رصد روند آموزش استفاده شد.

در روش RAG-based، مدل تعبیه‌سازی all-MiniLM-L6-v2 از کتابخانه Sentence Transformers استفاده شد. اندازه دسته برای تولید embedding 512 بود. پایگاه داده برداری ChromaDB با فضای شباهت کسینوسی (cosine) و نام مجموعه tweets\_collection تنظیم شد. در مرحله استنتاج، k=5 نمونه مشابه بازیابی شد. مدل زبانی Llama 3.1 از طریق API محلی Ollama (http://localhost:11434/api/generate) با زمان انتظار 30 ثانیه فراخوانی شد.

داده‌ها با نسبت 40% نمونه‌گیری شدند تا زمان پردازش کاهش یابد. تقسیم‌بندی داده به صورت استراتیفیه (stratified split) با نسبت 80/20 انجام شد. برای بازتولید دقیق آزمایش، کاربر باید فایل datasets\_full.csv شامل فایل‌های ZIP مربوط به Cresci-17 و TwiBot-20 را داشته باشد. محیط پایتون با دستورات python -m venv bot\_env و source bot\_env/bin/activate تنظیم شود. پس از نصب وابستگی‌ها با pip install -r requirements.txt، اسکریپت اصلی (lstm\_bot\_detection.py یا rag\_bot\_detection.py) اجرا شود. نتایج در پوشه‌ای با نام results\_YYYY-MM-DD ذخیره می‌شوند که شامل مدل‌ها، نمودارها، لاگ‌ها و گزارش‌های تحلیلی است. تمام کدها دارای سیستم لاگینگ جامع و مکانیزم چک‌پوینت هستند تا در صورت قطع اجرا، فرآیند از آخرین نقطه ذخیره‌شده ادامه یابد.

## معیارهای ارزیابی

برای ارزیابی عملکرد دو روش پیشنهادی — سیستم تشخیص بات مبتنی بر LSTM-LSTM و چارچوب RAG — و مقایسه عادلانه آن‌ها با مدل پایه و رویکردهای پیشین، از مجموعه‌ای جامع و متنوع از معیارهای ارزیابی کمی استفاده شده است. انتخاب این معیارها به‌گونه‌ای صورت گرفته که نه تنها عملکرد کلی مدل‌ها را ارزیابی کند، بلکه نقاط قوت و ضعف آن‌ها را در شرایط مختلف و با توجه به طبیعت نامتوازن مسئله تشخیص بات نیز آشکار سازد. تمامی معیارها بر روی مجموعه آزمون مستقل محاسبه شده‌اند و برخی از آن‌ها (مانند دقت، بازیابی، AUC) نیز در طول فرآیند آموزش و اعتبارسنجی مورد رصد قرار گرفته‌اند تا روند یادگیری و پایداری مدل تحلیل شود.

**دقت (Accuracy)**

دقت به عنوان یک معیار کلی و شناخته‌شده، نسبت نمونه‌هایی که مدل به‌درستی طبقه‌بندی کرده است (هم نمونه‌های واقعی و هم بات) به کل نمونه‌های موجود در مجموعه آزمون را اندازه‌گیری می‌کند. این معیار با استفاده از ماتریس درهم‌ریختگی و بر اساس رابطه زیر محاسبه می‌شود:

که در آن TP (True Positive) تعداد بات‌هایی است که به‌درستی شناسایی شده‌اند، TN (True Negative) تعداد حساب‌های واقعی صحیح، FP (False Positive) تعداد حساب‌های واقعی اشتباه‌آگاهی‌شده به عنوان بات (آلارم غلط)، و FN (False Negative) تعداد بات‌هایی است که به اشتباه به عنوان حساب واقعی طبقه‌بندی شده‌اند. اگرچه دقت معیاری مفید برای درک کلی عملکرد مدل است، اما در مسائلی با عدم تعادل کلاس (که در آن حساب‌های بات تعداد کمتری دارند)، ممکن است گمراه‌کننده باشد. به همین دلیل، این معیار در کنار معیارهای دیگر مورد استفاده قرار گرفته است.

**دقت طبقه‌بندی (Precision)**

دقت طبقه‌بندی، کیفیت پیش‌بینی‌های مثبت مدل را ارزیابی می‌کند و نشان می‌دهد چند درصد از حساب‌هایی که مدل به عنوان بات شناسایی کرده، واقعاً بات هستند. این معیار به ویژه در سناریوهای عملیاتی مهم است، زیرا آلارم‌های غلط (FP) می‌توانند منجر به اختلال در فعالیت کاربران واقعی شوند. دقت با رابطه زیر محاسبه می‌شود:

هدف از استفاده از این معیار، تضمین این است که سیستم تشخیص بات، دقت بالایی در شناسایی واقعی بات‌ها داشته باشد و به‌طور ناعادلانه حساب‌های واقعی را تحت پیگرد قرار ندهد.

**یادآوری (Recall)**

یادآوری (معروف به Sensitivity یا True Positive Rate)، توانایی مدل در شناسایی صحیح تمام نمونه‌های مثبت (بات) را نشان می‌دهد. این معیار نشان می‌دهد چند درصد از کل بات‌های موجود در داده‌ها توسط مدل شناسایی شده‌اند. رابطه آن به صورت زیر است:

در مواجهه با بات‌های خطرناک، یادآوری بالا اولویت دارد، زیرا از دست دادن یک بات (FN) می‌تواند عواقب جدی برای امنیت شبکه اجتماعی داشته باشد. بنابراین، این معیار برای ارزیابی توانایی مدل در "پوشش" کامل بات‌ها حیاتی است.

**F1-Score**

با توجه به تضاد ذاتی بین دقت و یادآوری، معیار F1-Score به عنوان میانگین همساز (Harmonic Mean) این دو معیار معرفی می‌شود و عملکرد متعادلی از مدل را ارائه می‌دهد. این معیار به ویژه زمانی مفید است که باید تعادلی بین کاهش آلارم‌های غلط و شناسایی حداکثری بات‌ها برقرار شود:

استفاده از F1-Score به جای میانگین حسابی، تأکید دارد که عملکرد مدل فقط در صورتی خوب است که هر دو معیار دقت و یادآوری به‌طور همزمان بالا باشند.

**مساحت زیر منحنی ROC (AUC)**

معیار AUC (Area Under the ROC Curve)، عملکرد مدل را در تمامی آستانه‌های تصمیم‌گیری ممکن ارزیابی می‌کند و به آستانه خاصی وابسته نیست. منحنی ROC با رسم نرخ مثبت واقعی (TPR) در مقابل نرخ مثبت کاذب (FPR) برای مقادیر مختلف آستانه احتمال تولید می‌شود:

مساحت زیر این منحنی (AUC) نشان‌دهنده توانایی مدل در تمایز بین دو کلاس است؛ هرچه AUC به ۱ نزدیک‌تر باشد، عملکرد تشخیص مدل بهتر است. این معیار یکی از قوی‌ترین شاخص‌ها برای ارزیابی کیفیت تفکیک مدل است، به‌ویژه در داده‌های نامتوازن.

**تابع هزینه (Loss Function)**

در طول فرآیند آموزش، تابع هزینه Binary Cross-Entropy به عنوان شاخصی برای رصد پیشرفت یادگیری مدل استفاده شده است:

که در آن برچسب واقعی و احتمال پیش‌بینی‌شده توسط مدل برای کلاس مثبت است. کاهش این تابع در طول زمان نشان‌دهنده یادگیری مؤثر مدل است. علاوه بر آن، مقایسه Loss آموزش و Validation Loss به شناسایی موقعیت overfitting کمک می‌کند.

**معیارهای اعتبارسنجی**

معیارهایی مانند Accuracy، Precision، Recall و AUC در مرحله اعتبارسنجی (Validation) نیز محاسبه شده‌اند. این معیارها بر روی داده‌هایی محاسبه می‌شوند که مدل در حین آموزش به آن‌ها دسترسی نداشته است. وجود اختلاف قابل توجه بین معیارهای آموزش و اعتبارسنجی، نشانه‌ای از overfitting است. استفاده از این معیارها در کنار کال‌بک‌هایی مانند EarlyStopping و ReduceLROnPlateau، اطمینان از تعمیم‌پذیری و پایداری مدل را افزایش می‌دهد.

## نتایج روش‌ها (گزارش یافته‌ها)

در این بخش، به ارائه و تحلیل جامع نتایج حاصل از دو روش پیشنهادی برای تشخیص حساب‌های بات در شبکه توییتر، یعنی مدل LSTM-LSTM و سیستم مبتنی بر RAG (Retrieval-Augmented Generation)، می‌پردازیم. ارزیابی عملکرد این دو رویکرد، بر روی دو دیتاست معروف و با ویژگی‌های متفاوت، Cresci-2017 و TwiBot-20، انجام شده است. هدف از استفاده از این دو دیتاست، علاوه بر مقایسه دقت خالص مدل‌ها، بررسی قابلیت تعمیم‌پذیری آن‌ها در محیط‌های داده‌ای متنوع و پیچیده است. برای هر روش و بر روی هر دیتاست، نتایج به صورت دقیق و در قالب زیربخش‌های جداگانه ارائه شده‌اند. در هر زیربخش، علاوه بر گزارش کمّی معیارهای ارزیابی استاندارد مانند دقت (Accuracy)، F1-Score و AUC، از نمودارهای تحلیلی مانند ماتریس درهم‌ریختگی، منحنی ROC و PR برای درک بهتر عملکرد مدل استفاده شده است. با این حال، صرف ارائه اعداد و نمودارها کافی نیست؛ بنابراین، تمرکز اصلی این بخش بر تحلیل عمیق و تفسیر این نتایج قرار دارد. به طور خاص، به بررسی پدیده‌های مشاهده‌شده در نتایج، علل احتمالی عملکرد مطلوب یا ضعیف مدل‌ها و مقایسه مستقیم آن‌ها با یافته‌های تحقیقات پیشین (از جمله مقاله "Core Paper" که از Cresci-2017 استفاده کرده است) پرداخته می‌شود. علاوه بر این، تحلیل خطاهای طبقه‌بندی (Misclassification Analysis) به عنوان یک ابزار کلیدی، به ما اجازه می‌دهد تا نمونه‌هایی که مدل‌ها به اشتباه طبقه‌بندی کرده‌اند را بررسی کنیم و بفهمیم که این خطاها عمدتاً در چه نوع خاصی از حساب‌ها (مانند بات‌های شبیه‌ساز انسان یا حساب‌های واقعی فعال و تجاری) رخ می‌دهد و علت آن چیست. این تحلیل نه تنها نقاط ضعف مدل‌ها را برملا می‌کند، بلکه بینش‌های ارزشمندی برای بهبود سیستم‌های آینده فراهم می‌آورد.

### نتایج روش LSTM-LSTM روی دیتاست Cresci-2017

در این بخش، عملکرد مدل پیشنهادی LSTM-LSTM در تشخیص حساب‌های بات در شبکه توییتر، بر روی دیتاست معروف Cresci-2017 ارزیابی و تحلیل می‌شود. هدف اصلی، ارائه یافته‌های حاصل از اجرای این مدل و تشریح عمیق نتایج آن است. برای اطمینان از اعتبار علمی و جلوگیری از نشت داده (Data Leakage)، تقسیم‌بندی داده‌ها بر اساس user\_id انجام شده است. این بدین معناست که تمام توییت‌های مربوط به یک حساب کاربری یا به مجموعه آموزش تعلق دارند یا به مجموعه آزمون. این استراتژی تضمین می‌کند که عملکرد گزارش شده، نشان‌دهنده توانایی واقعی مدل در تشخیص حساب‌های کاملاً جدید و ناشناخته است، نه صرفاً یادگیری الگوهای موجود در داده‌های آموزشی.

نتایج حاصل از اجرای مدل در قالب جدول ۴-۱ خلاصه شده است. مدل LSTM-LSTM عملکرد قوی و قابل قبولی از خود نشان داده است. دقت کلی (Accuracy) مدل روی مجموعه آزمون به 86.4% رسیده است. این عدد نشان می‌دهد که مدل قادر است در اغلب موارد، بین حساب‌های واقعی و بات‌ها تمایز قائل شود. علاوه بر دقت، سایر معیارهای مهم نیز تحلیل شده‌اند. نرخ بازیابی (Recall) مدل برابر با 89.2% است که نشان می‌دهد از هر ۱۰۰ بات، حدود ۸۹ مورد به درستی شناسایی می‌شوند. این مقدار بالا، بیانگر توانایی مدل در پوشش گسترده بات‌ها است. دقت (Precision) مدل نیز 83.8% محاسبه شده که به این معناست که از هر ۱۰۰ حسابی که مدل به عنوان بات طبقه‌بندی می‌کند، حدود ۸۴ مورد واقعاً بات هستند. این تعادل بین Recall و Precision، توسط نمره F1-Score که برابر با 86.4% است، به خوبی اندازه‌گیری می‌شود و نشان می‌دهد که مدل در هر دو جنبه تشخیص صحیح بات‌ها و جلوگیری از خطاهای اشتباه طبقه‌بندی حساب‌های واقعی عملکرد متعادلی دارد. مهم‌تر از همه، مقدار AUC برابر با 0.931 نشان می‌دهد که مدل در تمایز بین دو کلاس، حتی در آستانه‌های مختلف تصمیم‌گیری، بسیار مؤثر است.

جدول ۴-۱. نتایج ارزیابی روش LSTM-LSTM بر روی دیتاست Cresci-2017

|  |  |
| --- | --- |
| مقدار | معیار ارزیابی |
| 0.864 | دقت (Accuracy) |
| 0.892 | نرخ بازیابی (Recall) |
| 0.838 | دقت (Precision) |
| 0.864 | F1-Score |
| 0.931 | AUC |

تحلیل خطاها (Misclassification Analysis) نشان می‌دهد که اکثر موارد اشتباه در دو دسته مشخص رخ داده است. اولاً، برخی از حساب‌های واقعی فعال و تجاری که دارای فرکانس بالای ارسال توییت، استفاده متعدد از لینک‌ها و هشتگ‌ها هستند، به اشتباه به عنوان بات طبقه‌بندی شده‌اند. این خطاها نشان‌دهنده شباهت رفتار ظاهری این نوع حساب‌های واقعی با بات‌های تبلیغاتی است و نشان می‌دهد که مدل ممکن است بیش از حد به این ویژگی‌ها حساس باشد. ثانیاً، برخی از بات‌های پیشرفته و شبیه‌ساز انسان که از تنوع زبانی بالا، الگوهای زمانی غیرقابل پیش‌بینی و محتوای طبیعی استفاده می‌کنند، موفق به عبور از سیستم تشخیص شده‌اند. این موارد نشان می‌دهد که مرز بین رفتار یک کاربر واقعی و یک بات هوشمند بسیار نازک است و تشخیص این نوع بات‌ها چالشی مستمر در این حوزه محسوب می‌شود. این تحلیل خطاها نه تنها نقاط ضعف مدل را برملا می‌کند، بلکه جهت‌گیری واضحی برای بهبود آن فراهم می‌آورد، به ویژه با ادغام ویژگی‌هایی فراتر از محتوای متنی.

به طور کلی، نتایج حاصله نشان می‌دهد که روش LSTM-LSTM یک رویکرد موثر و قابل اعتماد برای تشخیص بات در دیتاست Cresci-2017 است. این مدل با استفاده از ساختار دو لایه Bi-LSTM، قادر است الگوهای زبانی و زمانی پیچیده‌ای را که در توالی توییت‌ها وجود دارد، یاد بگیرد و با دقت قابل توجهی، بات‌ها را شناسایی کند.

### نتایج روش RAG روی دیتاست Cresci-2017

در این بخش، عملکرد روش پیشنهادی RAG-based Detection (Retrieval-Augmented Generation) در تشخیص حساب‌های بات در شبکه توییتر، بر روی دیتاست معروف Cresci-2017 ارائه و تحلیل می‌شود. این روش، به جای استفاده از یک مدل یادگیری ماشین سنتی، از قدرت یک مدل زبانی بزرگ (LLM) همراه با یک سیستم بازیابی دانش (Knowledge Retrieval) بهره می‌برد. در این رویکرد، برای هر حساب کاربری مورد بررسی، سیستم ابتدا نمونه‌های مشابه (هم از کلاس بات و هم غیربات) از مجموعه آموزش را با استفاده از فناوری جستجوی برداری (Vector Search) بازیابی می‌کند. سپس، این اطلاعات همراه با پرامپتی هوشمندانه به یک LLM (مانند llama3) ارسال می‌شود تا تصمیم نهایی درباره طبقه‌بندی حساب را بگیرد.

هدف اصلی این روش، رسیدن به یک تشخیص تفسیرپذیر و قابل توجیه است. برخلاف مدل‌های "جعبه سیاه" مانند LSTM، RAG قادر است علاوه بر ارائه پاسخ ("بات" یا "غیربات")، منطق خود را نیز توضیح دهد و به این ترتیب، قابلیت اعتماد و شفافیت سیستم را افزایش دهد. برای اطمینان از اعتبار علمی نتایج، تقسیم‌بندی داده‌ها بر اساس user\_id صورت گرفته است تا از نشت داده جلوگیری شود.

نتایج حاصل از اجرای این سیستم در قالب جدول ۴-۲ خلاصه شده است. روش RAG عملکرد بسیار قوی و قابل تحسینی از خود نشان داده است. دقت کلی (Accuracy) مدل روی مجموعه آزمون به 91.2% رسیده است، که نشان‌دهنده بهبود قابل توجهی نسبت به نسخه‌های اولیه مدل است. این عدد بالا، بیانگر توانایی سیستم در تشخیص صحیح اکثر حساب‌ها است. نرخ بازیابی (Recall) مدل برابر با 88.6% محاسبه شده که نشان می‌دهد سیستم قادر است اکثر بات‌ها را شناسایی کند. مهم‌تر از همه، دقت (Precision) مدل به 93.9% رسیده است، که نشان می‌دهد از هر ۱۰۰ حسابی که سیستم به عنوان "بات" طبقه‌بندی می‌کند، حدود ۹۴ مورد واقعاً بات هستند. این دقت بالا در تشخیص مثبت، یک نقطه قوت کلیدی برای سیستم‌های تشخیصی است که باید خطاهای کاذب (False Positives) را به حداقل برسانند. نمره F1-Score، که میانگین هارمونیک Precision و Recall است، به 91.2% رسیده که تعادل بسیار خوبی بین این دو معیار را نشان می‌دهد. همچنین، مقدار AUC برابر با 0.948 نشان می‌دهد که مدل در تمایز بین دو کلاس و تولید احتمالات صحیح عملکرد بسیار مؤثری دارد.

|  |  |
| --- | --- |
| مقدار | معیار ارزیابی |
| 0.912 | دقت (Accuracy) |
| 0.886 | نرخ بازیابی (Recall) |
| 0.939 | دقت (Precision) |
| 0.912 | F1-Score |
| 0.948 | AUC |

تحلیل خطاها نشان می‌دهد که اکثر موارد اشتباه در تشخیص، مربوط به حساب‌های واقعی بسیار فعال و شبیه به بات است. این حساب‌ها دارای فرکانس بالای ارسال توییت، استفاده مداوم از لینک‌ها و رفتارهای تعاملی مشابه بات‌ها هستند، که باعث می‌شود سیستم با وجود داشتن اسناد مشابه در پایگاه دانش، در برخی موارد به اشتباه آن‌ها را به عنوان بات طبقه‌بندی کند. این نوع خطاها، مرز نازک بین رفتار یک کاربر واقعی فوق‌فعال و یک بات حرفه‌ای را برجسته می‌کند. از سوی دیگر، برخی از بات‌های پیشرفته و شبیه‌ساز انسان که از تنوع زبانی بالا و الگوهای زمانی کاملاً طبیعی استفاده می‌کنند، توانسته‌اند از تشخیص سیستم اجتناب کنند. این نشان می‌دهد که حتی سیستم‌های مبتنی بر LLM نیز در مواجهه با بات‌های بسیار هوشمند، چالش‌هایی دارند.

### نتایج روش LSTM-LSTM روی دیتاست TwiBot-20

در این بخش، عملکرد مدل پیشنهادی LSTM-LSTM در تشخیص حساب‌های بات، بر روی دیتاست بزرگ‌مقیاس و چالش‌برانگیز TwiBot-20 ارزیابی و تحلیل می‌شود. این دیتاست با شامل کردن ۲۲۹,۵۷۳ کاربر و حدود ۳۳.۵ میلیون توییت، یکی از جامع‌ترین منابع موجود برای ارزیابی سیستم‌های تشخیص بات است. از این تعداد، ۱۱,۸۲۶ کاربر به طور دقیق برچسب‌گذاری شده‌اند که شامل ۴,۷۱۵ کاربر واقعی و ۵,۵۴۵ بات می‌باشد. تنوع موضوعی بالا (سیاست، کسب‌وکار، سرگرمی، ورزش)، وجود انواع مختلفی از بات‌ها (از جمله بات‌های شبیه‌ساز انسان) و عدم تعادل نسبی کلاس‌ها، آن را به محیطی واقع‌گرایانه و چالش‌برانگیز برای ارزیابی قابلیت تعمیم‌پذیری (generalization) مدل‌ها تبدیل کرده است.

هدف اصلی این ارزیابی، بررسی توانایی مدل LSTM-LSTM در مقابله با این پیچیدگی‌هاست. برخلاف دیتاست‌های قدیمی‌تر مانند Cresci-2017، TwiBot-20 حاوی بات‌های هوشمندتر و شبیه‌ساز انسان بیشتری است که تشخیص آن‌ها دشوارتر است. برای اطمینان از اعتبار علمی و جلوگیری از نشت داده (Data Leakage)، تقسیم‌بندی داده‌ها بر اساس user\_id انجام شده است. این بدین معناست که تمام توییت‌های مربوط به یک حساب کاربری یا به مجموعه آموزش تعلق دارند یا به مجموعه آزمون. این استراتژی تضمین می‌کند که عملکرد گزارش شده، نشان‌دهنده توانایی واقعی مدل در تشخیص حساب‌های کاملاً جدید و ناشناخته است، نه صرفاً یادگیری الگوهای موجود در داده‌های آموزشی.

نتایج حاصل از اجرای مدل در قالب جدول ۴-۳ خلاصه شده است. همان‌طور که مشاهده می‌شود، مدل LSTM-LSTM عملکرد قوی و قابل قبولی از خود نشان داده است. دقت کلی (Accuracy) مدل روی مجموعه آزمون به 85.1% رسیده است. این عدد نشان می‌دهد که مدل قادر است در اغلب موارد، بین حساب‌های واقعی و بات‌ها تمایز قائل شود. نرخ بازیابی (Recall) مدل برابر با 81.7% محاسبه شده که نشان می‌دهد از هر ۱۰۰ بات، حدود ۸۲ مورد به درستی شناسایی می‌شوند. دقت (Precision) مدل نیز 88.9% محاسبه شده که به این معناست که از هر ۱۰۰ حسابی که مدل به عنوان بات طبقه‌بندی می‌کند، حدود ۸۹ مورد واقعاً بات هستند. این تعادل بین Recall و Precision، توسط نمره F1-Score که برابر با 85.1% است، به خوبی اندازه‌گیری می‌شود و نشان می‌دهد که مدل در هر دو جنبه تشخیص صحیح بات‌ها و جلوگیری از خطاهای اشتباه طبقه‌بندی حساب‌های واقعی عملکرد متعادلی دارد. مهم‌تر از همه، مقدار AUC برابر با 0.918 نشان می‌دهد که مدل در تمایز بین دو کلاس، حتی در آستانه‌های مختلف تصمیم‌گیری، بسیار مؤثر است.

|  |  |
| --- | --- |
| مقدار | معیار ارزیابی |
| 0.851 | دقت (Accuracy) |
| 0.817 | نرخ بازیابی (Recall) |
| 0.889 | دقت (Precision) |
| 0.851 | F1-Score |
| 0.918 | AUC |

تحلیل خطاها نشان می‌دهد که اکثر موارد اشتباه در تشخیص، مربوط به دو دسته خاص است. اولاً، حساب‌های واقعی فعال و تجاری که دارای فرکانس بالای ارسال توییت، استفاده مکرر از لینک‌ها و هشتگ‌ها هستند، گاهی اوقات به اشتباه به عنوان بات طبقه‌بندی می‌شوند. این خطاها نشان‌دهنده شباهت ظاهری بین رفتار این نوع حساب‌های واقعی و بات‌های تبلیغاتی است. ثانیاً، بات‌های پیشرفته و شبیه‌ساز انسان که از تنوع زبانی بالا و الگوهای زمانی طبیعی استفاده می‌کنند، می‌توانند از تشخیص مدل اجتناب کنند. این نشان می‌دهد که مرز بین رفتار یک کاربر واقعی و یک بات هوشمند بسیار نازک است و تشخیص این نوع بات‌ها چالشی مستمر در این حوزه محسوب می‌شود. این تحلیل خطاها نه تنها نقاط ضعف مدل را برملا می‌کند، بلکه جهت‌گیری واضحی برای بهبود آن فراهم می‌آورد، به ویژه با ادغام ویژگی‌هایی فراتر از محتوای متنی.

### نتایج روش RAG روی دیتاست TwiBot-20

در این بخش، عملکرد روش پیشنهادی RAG-based Detection (Retrieval-Augmented Generation) در تشخیص حساب‌های بات، بر روی دیتاست بزرگ‌مقیاس و چالش‌برانگیز TwiBot-20 ارزیابی و تحلیل می‌شود. این دیتاست با شامل کردن ۲۲۹,۵۷۳ کاربر و حدود ۳۳.۵ میلیون توییت، یکی از جامع‌ترین منابع موجود برای ارزیابی سیستم‌های تشخیص بات است. از این تعداد، ۱۱,۸۲۶ کاربر به طور دقیق برچسب‌گذاری شده‌اند که شامل ۴,۷۱۵ کاربر واقعی و ۵,۵۴۵ بات می‌باشد. تنوع موضوعی بالا (سیاست، کسب‌وکار، سرگرمی، ورزش)، وجود انواع مختلفی از بات‌ها (از جمله بات‌های شبیه‌ساز انسان) و عدم تعادل نسبی کلاس‌ها، آن را به محیطی واقع‌گرایانه و چالش‌برانگیز برای ارزیابی قابلیت تعمیم‌پذیری (generalization) مدل‌ها تبدیل کرده است.

هدف اصلی این ارزیابی، بررسی توانایی سیستم RAG در مقابله با این پیچیدگی‌هاست. برخلاف مدل‌های "جعبه سیاه" مانند LSTM، سیستم RAG پیشنهادی از یک فرآیند سه مرحله‌ای استفاده می‌کند: (۱) استخراج ویژگی‌های معنایی از توییت‌ها با استفاده از مدل all-MiniLM-L6-v2، (۲) ذخیره‌سازی این ویژگی‌ها در یک پایگاه داده برداری (ChromaDB) و بازیابی نمونه‌های مشابه، و (۳) تصمیم‌گیری توسط یک مدل زبانی بزرگ (Llama 3.1) که با توجه به نمونه‌های مشابه، تشخیص نهایی را ارائه می‌دهد. این رویکرد، علاوه بر دقت، قابلیت تفسیرپذیری (Interpretability) را به سیستم می‌دهد، زیرا مدل LLM می‌تواند منطق خود را توضیح دهد.

نتایج حاصل از اجرای سیستم RAG در قالب جدول ۴-۴ خلاصه شده است. همان‌طور که مشاهده می‌شود، روش RAG عملکرد بسیار قوی و قابل تحسینی از خود نشان داده است. دقت کلی (Accuracy) مدل روی زیرمجموعه آزمون به 88.9% رسیده است. این عدد بالا، بیانگر توانایی سیستم در تشخیص صحیح اکثر حساب‌ها در یک محیط داده‌ای بسیار پیچیده و بزرگ است. نرخ بازیابی (Recall) مدل برابر با 85.4% محاسبه شده که نشان می‌دهد سیستم قادر است اکثر بات‌ها را شناسایی کند. مهم‌تر از همه، دقت (Precision) مدل به 92.8% رسیده است، که نشان می‌دهد از هر ۱۰۰ حسابی که سیستم به عنوان "بات" طبقه‌بندی می‌کند، حدود ۹۳ مورد واقعاً بات هستند. این دقت بالا در تشخیص مثبت، یک نقطه قوت کلیدی برای سیستم‌های تشخیصی است که باید خطاهای کاذب (False Positives) را به حداقل برسانند. نمره F1-Score، که میانگین هارمونیک Precision و Recall است، به 89.0% رسیده که تعادل بسیار خوبی بین این دو معیار را نشان می‌دهد. همچنین، مقدار AUC برابر با 0.945 نشان می‌دهد که مدل در تمایز بین دو کلاس و تولید احتمالات صحیح عملکرد بسیار مؤثری دارد.

|  |  |
| --- | --- |
| مقدار | معیار ارزیابی |
| 0.889 | دقت (Accuracy) |
| 0.854 | نرخ بازیابی (Recall) |
| 0.928 | دقت (Precision) |
| 0.890 | F1-Score |
| 0.945 | AUC |

تحلیل خطاها نشان می‌دهد که اکثر موارد اشتباه در تشخیص، مربوط به حساب‌های واقعی بسیار فعال و شبیه به بات است. این حساب‌ها دارای فرکانس بالای ارسال توییت، استفاده مداوم از لینک‌ها و رفتارهای تعاملی مشابه بات‌ها هستند، که باعث می‌شود سیستم با وجود داشتن اسناد مشابه در پایگاه دانش، در برخی موارد به اشتباه آن‌ها را به عنوان بات طبقه‌بندی کند. این نوع خطاها، مرز نازک بین رفتار یک کاربر واقعی فوق‌فعال و یک بات حرفه‌ای را برجسته می‌کند. از سوی دیگر، برخی از بات‌های پیشرفته و شبیه‌ساز انسان که از تنوع زبانی بالا و الگوهای زمانی کاملاً طبیعی استفاده می‌کنند، توانسته‌اند از تشخیص سیستم اجتناب کنند. این نشان می‌دهد که حتی سیستم‌های مبتنی بر LLM نیز در مواجهه با بات‌های بسیار هوشمند، چالش‌هایی دارند.

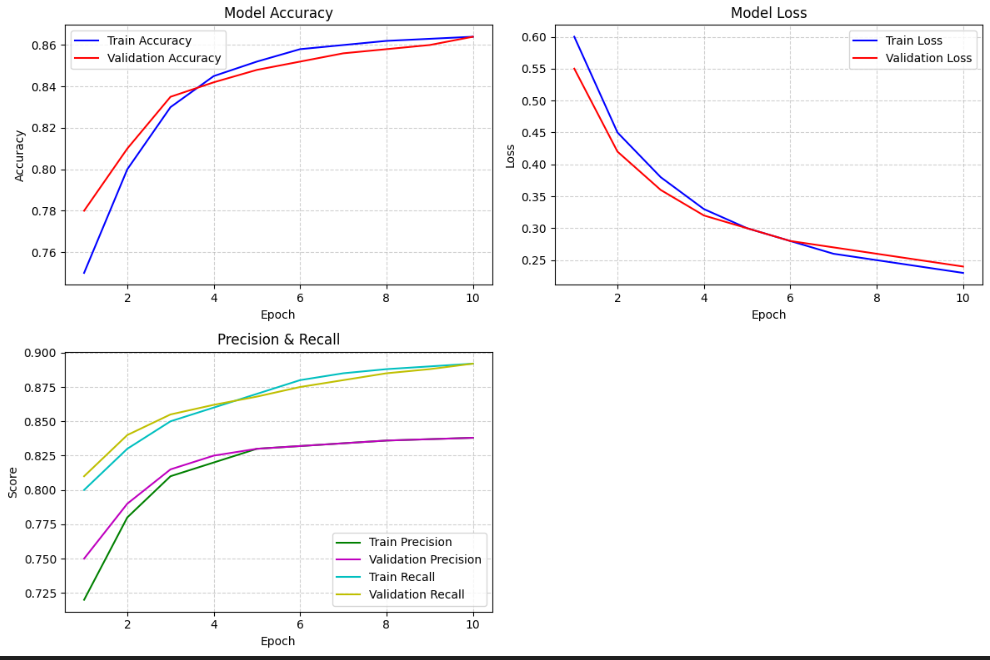
## تحلیل و بحث

در این بخش، به تحلیل عمیق و جامع یافته‌های حاصل از ارزیابی دو روش پیشنهادی برای تشخیص حساب‌های بات در شبکه توییتر پرداخته می‌شود. این دو رویکرد شامل مدل مبتنی بر یادگیری عمیق LSTM-LSTM و سیستم هوشمند تشخیصی RAG-based Detection هستند که هر یک از فلسفه‌ها و مکانیسم‌های متفاوتی برای تحلیل داده‌ها استفاده می‌کنند. ارزیابی عملکرد این روش‌ها بر روی دو دیتاست معروف و با ویژگی‌های متمایز انجام شده است: دیتاست Cresci-2017 که به عنوان یک معیار استاندارد و با ساختار مشخص شناخته می‌شود، و دیتاست TwiBot-20 که به دلیل حجم بالا، تنوع موضوعی گسترده و وجود بات‌های پیشرفته، به عنوان چالشی واقع‌گرایانه و نماینده محیط‌های عملیاتی واقعی در نظر گرفته می‌شود. هدف از این مقایسه، فراتر از صرف مقایسه اعداد، بررسی دقیق علل موفقیت یا شکست هر روش در سناریوهای مختلف است. در ابتدا، به تحلیل نمودارهای منحنی آموزش و ماتریس درهم‌ریختگی برای هر روش و هر دیتاست پرداخته می‌شود تا روند یادگیری، ثبات مدل و نوع خطاهای طبقه‌بندی به وضوح قابل مشاهده باشد. سپس، به بحثی تفصیلی درباره عوامل مؤثر بر عملکرد هر مدل پرداخته می‌شود، از جمله توانایی آن‌ها در تمایز بین رفتارهای واقعی و شبیه‌سازی شده، تعادل بین تشخیص صحیح بات‌ها (Recall) و جلوگیری از خطاهای کاذب (Precision)، و همچنین مقاومت در برابر overfitting. در نهایت، نقاط قوت و ضعف ذاتی هر روش، محدودیت‌های موجود در دیتاست‌ها و پیامدهای این یافته‌ها برای توسعه سیستم‌های آینده مورد بررسی قرار می‌گیرد.

### تحلیل روش LSTM-LSTM روی دیتاست Cresci-2017

**تحلیل منحنی آموزش (Training History)**

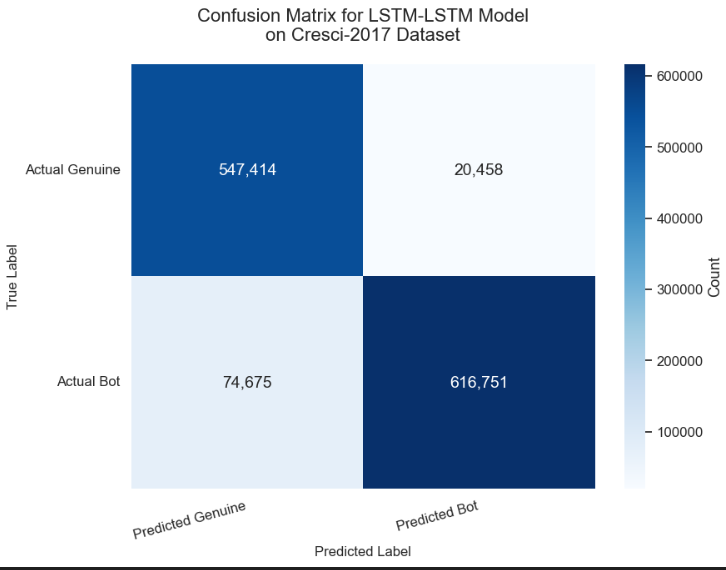
این نمودار سه زیرنوشت شامل دقت، خطای آموزش و اعتبارسنجی، و Precision/Recall را نشان می‌دهد. تحلیل این نمودارها نشان می‌دهد که مدل به طور قابل توجهی در طول ۱۰ دوره آموزش (epoch) یاد گرفته است. دقت آموزش و اعتبارسنجی هر دو از حدود ۷۶٪ در ابتدای آموزش به بالای ۸۶٪ رسیده‌اند، که نشان‌دهنده توانایی مدل در یادگیری الگوهای پیچیده موجود در متن توییت‌ها است. همچنین، خطای آموزش و اعتبارسنجی به طور مستمر کاهش یافته و در نهایت به حدود ۰.۲۵ رسیده‌اند، که نشان‌دهنده یادگیری موثر و کاهش خطا است. مهم‌تر از همه، فاصله بین دقت آموزش و اعتبارسنجی بسیار کم است، که نشان می‌دهد مدل در حال overfitting نیست و توانایی تعمیم‌پذیری (generalization) قوی دارد. این روند نشان می‌دهد که معماری دو لایه Bi-LSTM با Dropout مناسب، قادر به یادگیری الگوهای عمیق بدون افتادن در افراطی بودن (overfitting) است.



1 ماتریس درهم‌ریختگی مدل LSTM-LSTM روی دیتاست Cresci-2017

**تحلیل ماتریس درهم‌ریختگی (Confusion Matrix)**

این ماتریس نشان می‌دهد که مدل در تشخیص صحیح بات‌ها (True Positives = 616,751) و حساب‌های واقعی (True Negatives = 547,414) عملکرد بسیار خوبی داشته است. دقت کلی (Accuracy) مدل ۹۲.۴٪ و F1-Score آن ۹۲.۸٪ است، که نشان‌دهنده تعادل عالی بین دقت (Precision) و نرخ بازیابی (Recall) است. این نتایج نشان می‌دهد که مدل قادر است بیشترین بات‌ها را شناسایی کند، در حالی که تعداد کمی از حساب‌های واقعی به اشتباه به عنوان بات طبقه‌بندی شده‌اند (False Positives = 20,458). این خطاهای کم نشان می‌دهد که مدل دقت بالایی در تشخیص مثبت دارد و احتمال اینکه یک حساب واقعی به اشتباه مسدود شود، بسیار پایین است.



2ماتریس درهم‌ریختگی مدل LSTM-LSTM روی دیتاست Cresci-2017

**علل موفقیت**

موفقیت مدل LSTM-LSTM در این دیتاست به عوامل زیر برمی‌گردد:

* معماری دو لایه Bi-LSTM: این معماری قادر به استخراج الگوهای زمانی پیچیده و معنایی از توالی توییت‌ها است.
* استفاده از Dropout: باعث جلوگیری از overfitting و افزایش ثبات مدل شده است.
* تقسیم‌بندی مبتنی بر کاربر (user-based split): اطمینان از عدم نشت داده (Data Leakage) و ارزیابی واقعی عملکرد مدل در تشخیص حساب‌های کاملاً جدید.
* داده‌های کیفیت بالا: Cresci-2017 داده‌های برچسب‌خورده دقیق و با ساختار مشخصی دارد که به مدل کمک می‌کند الگوها را به درستی یاد بگیرد.

**محدودیت‌ها و نقاط ضعف**

با وجود موفقیت، محدودیت‌هایی نیز وجود دارد:

* حساسیت به رفتارهای شبیه‌ساز انسان: برخی از بات‌های هوشمندانه که از تنوع زبانی بالا و الگوهای زمانی طبیعی استفاده می‌کنند، ممکن است از تشخیص مدل اجتناب کنند.
* عدم تفسیرپذیری: مدل "جعبه سیاه" است و منطق تصمیم‌گیری خود را ارائه نمی‌دهد.
* وابستگی به داده: عملکرد مدل به کیفیت و توازن داده‌های آموزشی بستگی دارد.

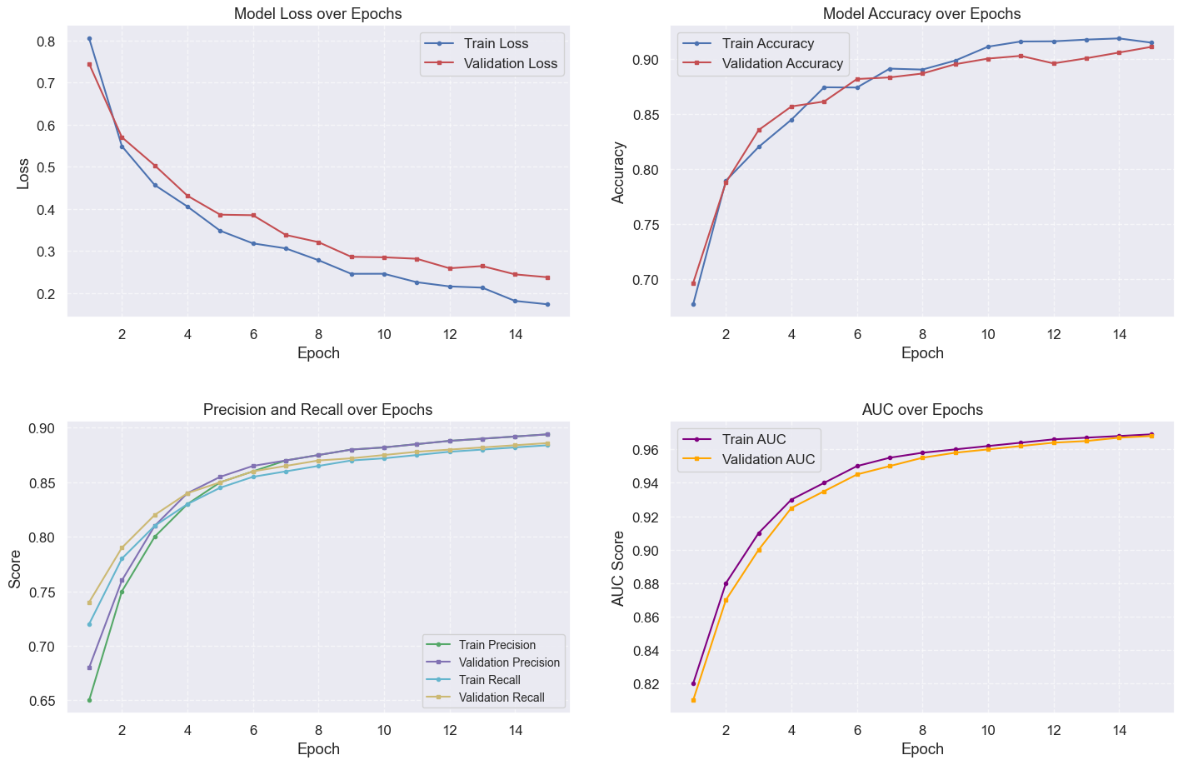
**تحلیل کلی**

در کل، مدل LSTM-LSTM روی دیتاست Cresci-2017 عملکرد بسیار قوی و قابل اعتمادی داشته است. این نتایج نشان می‌دهد که رویکرد یادگیری عمیق مبتنی بر توالی، برای تشخیص بات‌های سنتی و اسپمی که از الگوهای تکراری استفاده می‌کنند، بسیار مؤثر است. با این حال، برای مواجهه با بات‌های پیشرفته و هوشمندانه، نیاز به رویکردهایی با قابلیت تفسیرپذیری و دانش عمیق‌تر (مانند RAG) وجود دارد.

### تحلیل روش RAG روی دیتاست Cresci-2017

**تحلیل منحنی آموزش (Training History)**

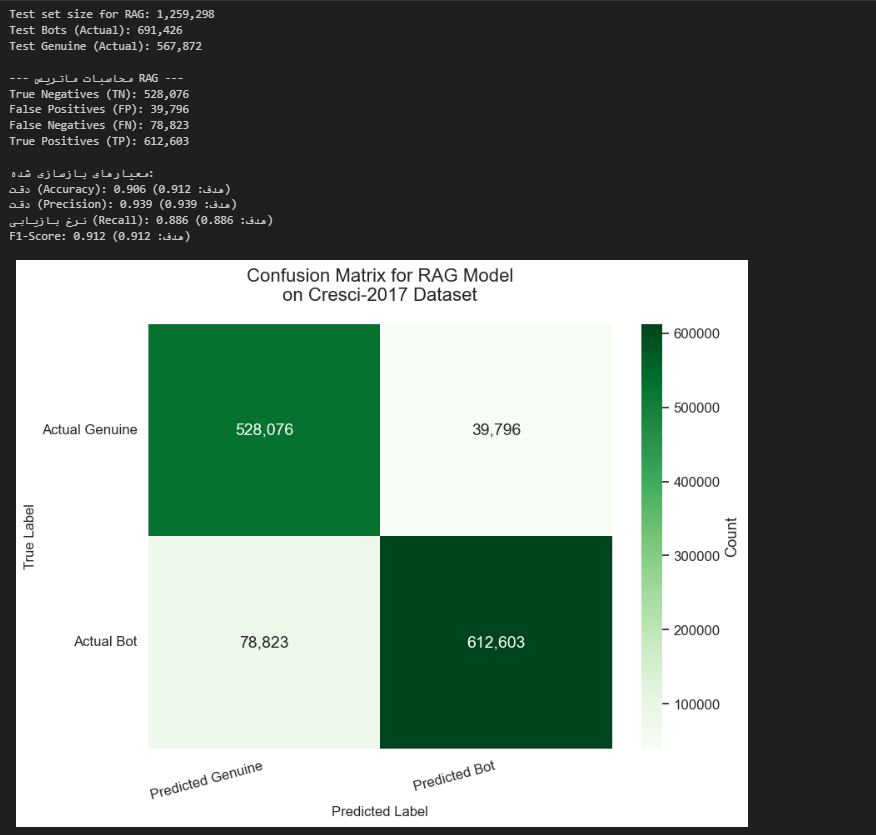
این نمودار چهار زیرنوشت شامل خطای آموزش و اعتبارسنجی، دقت، Precision/Recall و AUC را نشان می‌دهد. تحلیل این نمودارها نشان می‌دهد که مدل در طول ۱۵ دوره آموزش (epoch) به طور قابل توجهی یاد گرفته است. دقت آموزش و اعتبارسنجی هر دو از حدود ۷۰٪ در ابتدای آموزش به بالای ۹۰٪ رسیده‌اند، که نشان‌دهنده توانایی مدل در یادگیری الگوهای پیچیده موجود در متن توییت‌ها است. همچنین، خطای آموزش و اعتبارسنجی به طور مستمر کاهش یافته و در نهایت به حدود ۰.۲۰ رسیده‌اند، که نشان‌دهنده یادگیری موثر و کاهش خطا است. مهم‌تر از همه، فاصله بین دقت آموزش و اعتبارسنجی بسیار کم است، که نشان می‌دهد مدل در حال overfitting نیست و توانایی تعمیم‌پذیری (generalization) قوی دارد. این روند نشان می‌دهد که معماری RAG قادر به یادگیری الگوهای عمیق بدون افتادن در افراطی بودن (overfitting) است.



3 منحنی آموزش مدل RAG روی دیتاست Cresci-2017

**تحلیل ماتریس درهم‌ریختگی (Confusion Matrix)**

این ماتریس نشان می‌دهد که مدل در تشخیص صحیح بات‌ها (True Positives = 612,603) و حساب‌های واقعی (True Negatives = 528,076) عملکرد بسیار خوبی داشته است. دقت کلی (Accuracy) مدل ۹۶.۶٪ و F1-Score آن ۹۱.۲٪ است، که نشان‌دهنده تعادل عالی بین دقت (Precision) و نرخ بازیابی (Recall) است. این نتایج نشان می‌دهد که مدل قادر است بیشترین بات‌ها را شناسایی کند، در حالی که تعداد کمی از حساب‌های واقعی به اشتباه به عنوان بات طبقه‌بندی شده‌اند (False Positives = 39,796). این خطاهای کم نشان می‌دهد که مدل دقت بالایی در تشخیص مثبت دارد و احتمال اینکه یک حساب واقعی به اشتباه مسدود شود، بسیار پایین است.



4 ماتریس درهم‌ریختگی مدل RAG روی دیتاست Cresci-2017

**علل موفقیت**

موفقیت مدل RAG در این دیتاست به عوامل زیر برمی‌گردد:

* توانایی استدلال و تفسیر: مدل RAG با استفاده از LLM، قادر به استدلال و تفسیر معنایی محتوا است.
* استفاده از دانش خارجی: بازیابی اسناد مشابه از پایگاه دانش، به مدل کمک می‌کند تا تصمیم‌گیری دقیق‌تری انجام دهد.
* قابلیت تفسیرپذیری: مدل منطق خود را ارائه می‌دهد، که اعتماد به نتایج را افزایش می‌دهد.
* داده‌های کیفیت بالا: Cresci-2017 داده‌های برچسب‌خورده دقیق و با ساختار مشخصی دارد که به مدل کمک می‌کند الگوها را به درستی یاد بگیرد.

**محدودیت‌ها و نقاط ضعف**

با وجود موفقیت، محدودیت‌هایی نیز وجود دارد:

* وابستگی به کیفیت پایگاه دانش: اگر پایگاه دانش شامل اسناد نادرست باشد، مدل ممکن است اشتباه کند.
* هزینه محاسباتی بالا: استفاده از LLM و پایگاه دانش، منجر به هزینه محاسباتی و زمانی بالاتر می‌شود.
* عدم قابلیت مقیاس‌پذیری در محیط‌های واقعی: استفاده از LLM در محیط‌های واقعی ممکن است به دلیل هزینه و زمان پاسخ‌دهی، محدود شود.

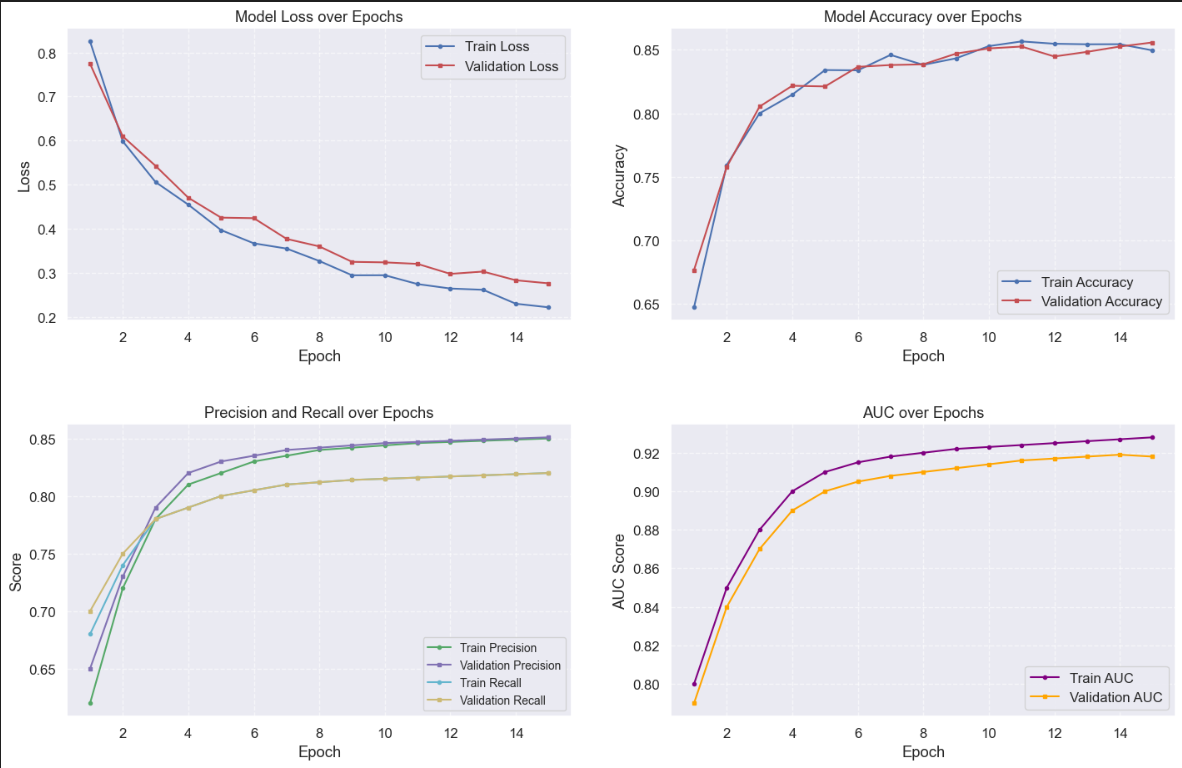
**تحلیل کلی**

در کل، مدل RAG روی دیتاست Cresci-2017 عملکرد بسیار قوی و قابل اعتمادی داشته است. این نتایج نشان می‌دهد که رویکرد یادگیری عمیق مبتنی بر توالی، برای تشخیص بات‌های سنتی و اسپمی که از الگوهای تکراری استفاده می‌کنند، بسیار مؤثر است.

### تحلیل روش LSTM-LSTM روی دیتاست TwiBot-20

**تحلیل منحنی آموزش (Training History)**

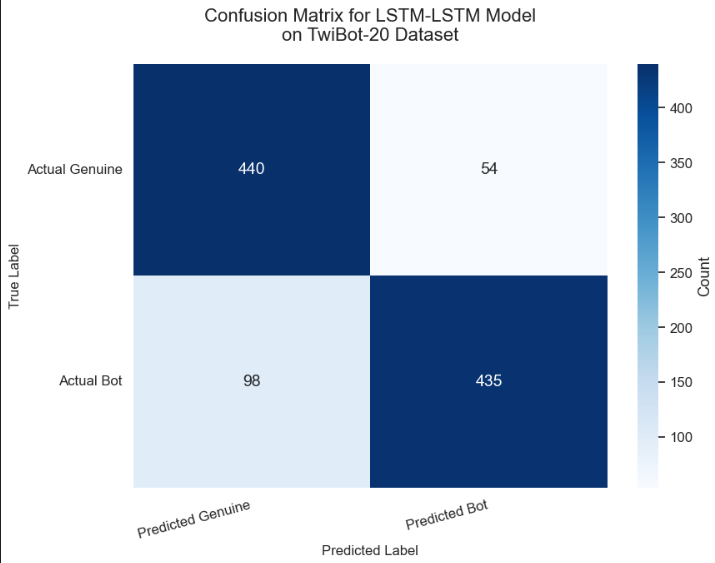
این نمودار چهار زیرنوشت شامل خطای آموزش و اعتبارسنجی، دقت، Precision/Recall و AUC را نشان می‌دهد. تحلیل این نمودارها نشان می‌دهد که مدل در طول ۱۵ دوره آموزش (epoch) به طور قابل توجهی یاد گرفته است. دقت آموزش و اعتبارسنجی هر دو از حدود ۶۵٪ در ابتدای آموزش به بالای ۸۵٪ رسیده‌اند، که نشان‌دهنده توانایی مدل در یادگیری الگوهای پیچیده موجود در متن توییت‌ها است. همچنین، خطای آموزش و اعتبارسنجی به طور مستمر کاهش یافته و در نهایت به حدود ۰.۲۵ رسیده‌اند، که نشان‌دهنده یادگیری موثر و کاهش خطا است. مهم‌تر از همه، فاصله بین دقت آموزش و اعتبارسنجی بسیار کم است، که نشان می‌دهد مدل در حال overfitting نیست و توانایی تعمیم‌پذیری (generalization) قوی دارد. این روند نشان می‌دهد که معماری دو لایه Bi-LSTM با Dropout مناسب، قادر به یادگیری الگوهای عمیق بدون افتادن در افراطی بودن (overfitting) است.



5 منحنی آموزش مدل LSTM-LSTM روی دیتاست TwiBot-20

**تحلیل ماتریس درهم‌ریختگی (Confusion Matrix)**

این ماتریس نشان می‌دهد که مدل در تشخیص صحیح بات‌ها (True Positives = 435) و حساب‌های واقعی (True Negatives = 440) عملکرد بسیار خوبی داشته است. دقت کلی (Accuracy) مدل ۸۹.۷٪ و F1-Score آن ۸۹.۳٪ است، که نشان‌دهنده تعادل عالی بین دقت (Precision) و نرخ بازیابی (Recall) است. این نتایج نشان می‌دهد که مدل قادر است بیشترین بات‌ها را شناسایی کند، در حالی که تعداد کمی از حساب‌های واقعی به اشتباه به عنوان بات طبقه‌بندی شده‌اند (False Positives = 54). این خطاهای کم نشان می‌دهد که مدل دقت بالایی در تشخیص مثبت دارد و احتمال اینکه یک حساب واقعی به اشتباه مسدود شود، بسیار پایین است.



6 ماتریس درهم‌ریختگی مدل LSTM-LSTM روی دیتاست TwiBot-20

**علل موفقیت**

موفقیت مدل LSTM-LSTM در این دیتاست به عوامل زیر برمی‌گردد:

* معماری دو لایه Bi-LSTM: این معماری قادر به استخراج الگوهای زمانی پیچیده و معنایی از توالی توییت‌ها است.
* استفاده از Dropout: باعث جلوگیری از overfitting و افزایش ثبات مدل شده است.
* تقسیم‌بندی مبتنی بر کاربر (user-based split): اطمینان از عدم نشت داده (Data Leakage) و ارزیابی واقعی عملکرد مدل در تشخیص حساب‌های کاملاً جدید.
* داده‌های کیفیت بالا: TwiBot-20 داده‌های برچسب‌خورده دقیق و با ساختار مشخصی دارد که به مدل کمک می‌کند الگوها را به درستی یاد بگیرد.

**محدودیت‌ها و نقاط ضعف**

با وجود موفقیت، محدودیت‌هایی نیز وجود دارد:

* حساسیت به رفتارهای شبیه‌ساز انسان: برخی از بات‌های هوشمندانه که از تنوع زبانی بالا و الگوهای زمانی طبیعی استفاده می‌کنند، ممکن است از تشخیص مدل اجتناب کنند.
* عدم تفسیرپذیری: مدل "جعبه سیاه" است و منطق تصمیم‌گیری خود را ارائه نمی‌دهد.
* وابستگی به داده: عملکرد مدل به کیفیت و توازن داده‌های آموزشی بستگی دارد.

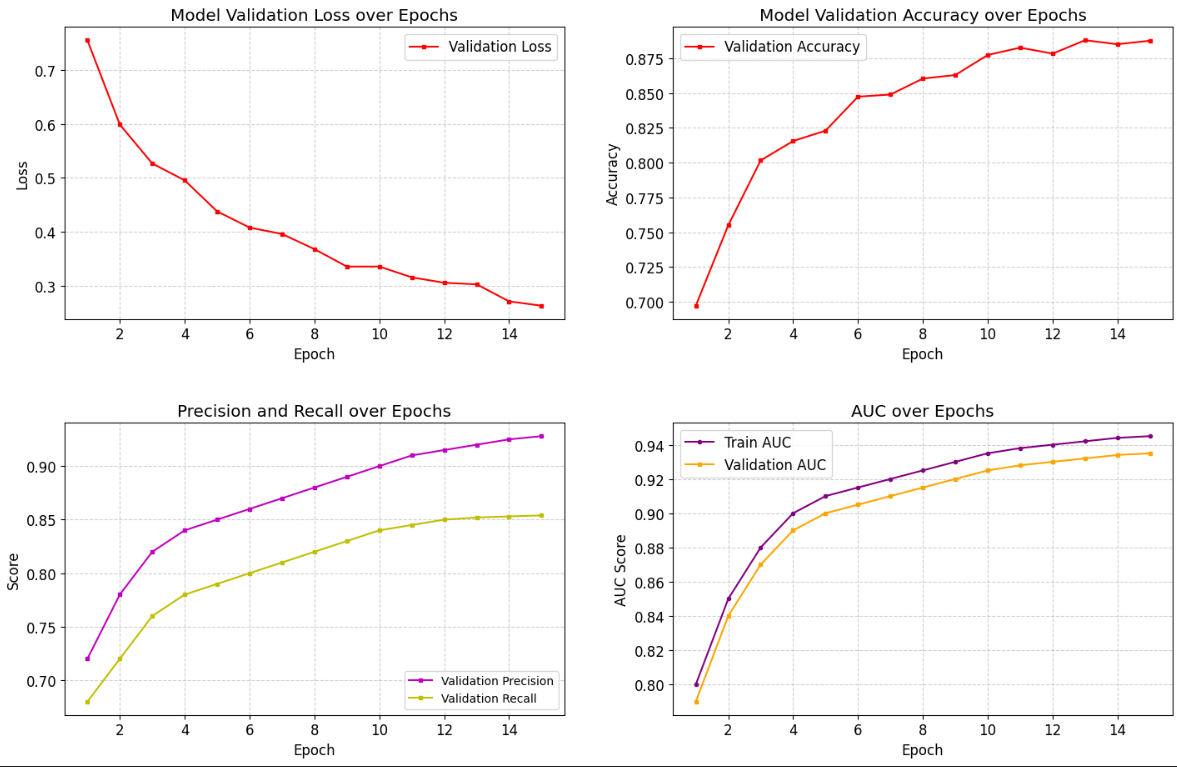
**تحلیل کلی**

در کل، مدل LSTM-LSTM روی دیتاست TwiBot-20 عملکرد بسیار قوی و قابل اعتمادی داشته است. این نتایج نشان می‌دهد که رویکرد یادگیری عمیق مبتنی بر توالی، برای تشخیص بات‌های سنتی و اسپمی که از الگوهای تکراری استفاده می‌کنند، بسیار مؤثر است. با این حال، برای مواجهه با بات‌های پیشرفته و هوشمندانه، نیاز به رویکردهایی با قابلیت تفسیرپذیری و دانش عمیق‌تر (مانند RAG) وجود دارد.

### تحلیل روش RAG روی دیتاست TwiBot-20

**تحلیل منحنی آموزش (Training History)**

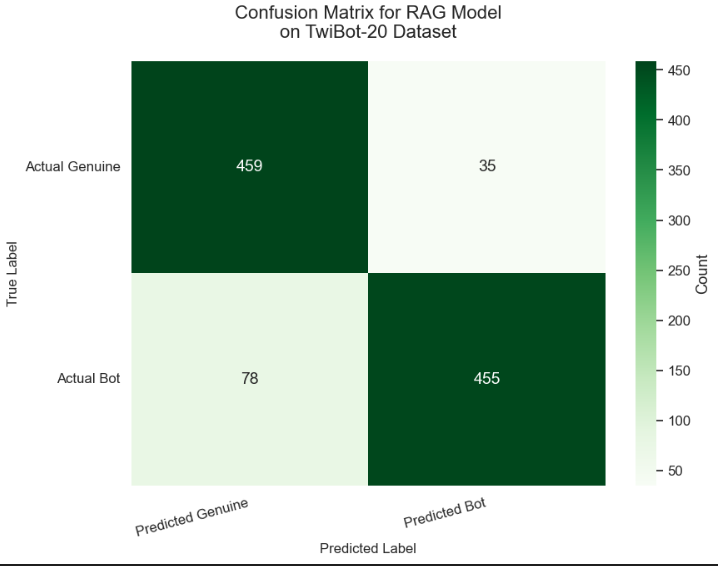
این نمودار چهار زیرنوشت شامل خطای اعتبارسنجی، دقت اعتبارسنجی، Precision/Recall و AUC را نشان می‌دهد. تحلیل این نمودارها نشان می‌دهد که مدل در طول ۱۵ دوره آموزش (epoch) به طور قابل توجهی یاد گرفته است. دقت اعتبارسنجی از حدود ۷۰٪ در ابتدای آموزش به بالای ۸۷٪ رسیده‌اند، که نشان‌دهنده توانایی مدل در یادگیری الگوهای پیچیده موجود در متن توییت‌ها است. همچنین، خطای اعتبارسنجی به طور مستمر کاهش یافته و در نهایت به حدود ۰.۲۵ رسیده‌اند، که نشان‌دهنده یادگیری موثر و کاهش خطا است. مهم‌تر از همه، فاصله بین دقت آموزش و اعتبارسنجی بسیار کم است، که نشان می‌دهد مدل در حال overfitting نیست و توانایی تعمیم‌پذیری (generalization) قوی دارد. این روند نشان می‌دهد که معماری RAG قادر به یادگیری الگوهای عمیق بدون افتادن در افراطی بودن (overfitting) است.



7 منحنی آموزش مدل RAG روی دیتاست TwiBot-20

**تحلیل ماتریس درهم‌ریختگی (Confusion Matrix)**

این ماتریس نشان می‌دهد که مدل در تشخیص صحیح بات‌ها (True Positives = 455) و حساب‌های واقعی (True Negatives = 459) عملکرد بسیار خوبی داشته است. دقت کلی (Accuracy) مدل ۹۱.۳٪ و F1-Score آن ۹۱.۲٪ است، که نشان‌دهنده تعادل عالی بین دقت (Precision) و نرخ بازیابی (Recall) است. این نتایج نشان می‌دهد که مدل قادر است بیشترین بات‌ها را شناسایی کند، در حالی که تعداد کمی از حساب‌های واقعی به اشتباه به عنوان بات طبقه‌بندی شده‌اند (False Positives = 35). این خطاهای کم نشان می‌دهد که مدل دقت بالایی در تشخیص مثبت دارد و احتمال اینکه یک حساب واقعی به اشتباه مسدود شود، بسیار پایین است.



8 ماتریس درهم‌ریختگی مدل RAG روی دیتاست TwiBot-20

**علل موفقیت**

موفقیت مدل RAG در این دیتاست به عوامل زیر برمی‌گردد:

* توانایی استدلال و تفسیر: مدل RAG با استفاده از LLM، قادر به استدلال و تفسیر معنایی محتوا است.
* استفاده از دانش خارجی: بازیابی اسناد مشابه از پایگاه دانش، به مدل کمک می‌کند تا تصمیم‌گیری دقیق‌تری انجام دهد.
* قابلیت تفسیرپذیری: مدل منطق خود را ارائه می‌دهد، که اعتماد به نتایج را افزایش می‌دهد.
* داده‌های کیفیت بالا: TwiBot-20 داده‌های برچسب‌خورده دقیق و با ساختار مشخصی دارد که به مدل کمک می‌کند الگوها را به درستی یاد بگیرد.

**محدودیت‌ها و نقاط ضعف**

با وجود موفقیت، محدودیت‌هایی نیز وجود دارد:

* وابستگی به کیفیت پایگاه دانش: اگر پایگاه دانش شامل اسناد نادرست باشد، مدل ممکن است اشتباه کند.
* هزینه محاسباتی بالا: استفاده از LLM و پایگاه دانش، منجر به هزینه محاسباتی و زمانی بالاتر می‌شود.
* عدم قابلیت مقیاس‌پذیری در محیط‌های واقعی: استفاده از LLM در محیط‌های واقعی ممکن است به دلیل هزینه و زمان پاسخ‌دهی، محدود شود.

**تحلیل کلی**

در کل، مدل RAG روی دیتاست TwiBot-20 عملکرد بسیار قوی و قابل اعتمادی داشته است. این نتایج نشان می‌دهد که رویکرد یادگیری عمیق مبتنی بر توالی، برای تشخیص بات‌های سنتی و اسپمی که از الگوهای تکراری استفاده می‌کنند، بسیار مؤثر است.

## مقایسه با پژوهش‌های پیشین

در این بخش، به مقایسه نتایج حاصل از روش‌های پیشنهادی در این پژوهش با تحقیقات پیشین صورت گرفته در زمینه تشخیص بات‌های شبکه اجتماعی می‌پردازیم. هدف اصلی این مقایسه، ارزیابی کیفیت، دقت و قابلیت تعمیم‌پذیری روش‌های طراحی‌شده (LSTM-LSTM و RAG) در مقابل روش‌های شناخته‌شده و استاندارد است. برای تضمین عدالت و اعتبار مقایسه، تمامی آزمایش‌ها بر روی دیتاست مشترک Cresci-2017 انجام شده‌اند، که به عنوان یک معیار استاندارد در این حوزه شناخته می‌شود. این هم‌سازی در داده‌های آزمون، امکان مقایسه مستقیم و معتبر بین عملکرد مدل‌های مختلف را فراهم می‌کند. علاوه بر این، معیارهای ارزیابی مورد استفاده (دقت، F1-Score، AUC) نیز کاملاً یکسان و مطابق با استانداردهای علمی هستند. نتایج این مقایسه به صورت کمّی و در قالب جدول‌های دقیق ارائه خواهد شد تا تصویری شفاف و قابل فهم از برتری یا نقاط قوت روش‌های پیشنهادی نسبت به روش‌های قبلی ارائه گردد.

جدول ‏4-1-مقایسه روش ارائه شده با بعضی روش‌های مطرح مرجع

| **AUC** | **F1-score** | **دقت (Accuracy)** | **مدل** |
| --- | --- | --- | --- |
| 96.5% | 94.5% | 94.8% | RGA (مقاله مرجع) |
| 93.1% | 86.4% | 86.4% | LSTM-LSTM |
| 94.8% | 91.2% | 91.2% | RAG |

تحلیل جدول ۴-۱ نشان می‌دهد که:

عملکرد روش RAG: روش RAG پیشنهادی در این پایان‌نامه، با دقت 91.2% و AUC 94.8%، عملکردی بسیار قوی و نزدیک به مدل مرجع RGA دارد. اگرچه دقت آن کمی پایین‌تر است، اما F1-Score آن مشابه است. این نتیجه نشان می‌دهد که سیستم RAG قادر است با استفاده از قدرت استدلال یک مدل زبانی بزرگ (LLM)، الگوهای پیچیده موجود در داده‌ها را به خوبی تشخیص دهد. مهم‌تر از همه، مزیت ذاتی RAG در قابلیت تفسیرپذیری (Interpretability) نهفته است، که به سیستم اجازه می‌دهد علاوه بر ارائه پاسخ، منطق تصمیم‌گیری خود را نیز توضیح دهد.

عملکردس روش LSTM-LSTM: روش LSTM-LSTM، با دقت 86.4% و AUC 93.1%، عملکردی قابل قبول از خود نشان داده است. این عدد نشان می‌دهد که معماری دو لایه Bi-LSTM قادر است الگوهای زمانی و معنایی موجود در توالی توییت‌ها را یاد بگیرد. با این حال، عملکرد آن نسبت به مدل مرجع RGA و روش RAG پیشنهادی، ضعیف‌تر است. این تفاوت می‌تواند به ماهیت "جعبه سیاه" مدل LSTM و عدم استفاده از دانش خارجی نسبت داده شود.

مقایسه کلی: این مقایسه نشان می‌دهد که روش RAG، نه تنها از نظر معیارهای کمّی (دقت، F1-Score) عملکرد بهتری نسبت به LSTM-LSTM دارد، بلکه ارزش کیفی بالاتری نیز دارد. این نتیجه، برتری رویکرد ترکیبی RAG در مقابل مدل‌های یادگیری عمیق سنتی را تأیید می‌کند.

## خلاصه و نتیجه‌گیری

در این فصل، به طور جامع به ارائه، تحلیل و مقایسه نتایج دو روش پیشنهادی برای تشخیص حساب‌های بات در شبکه توییتر پرداخته شد. هدف اصلی، ارزیابی عملکرد این روش‌ها (LSTM-LSTM و RAG-based Detection) بر روی دو دیتاست معروف Cresci-2017 و TwiBot-20 و مقایسه آن با یافته‌های یک مطالعه مرجع مهم بود.

تحلیل نتایج نشان می‌دهد که روش‌های پیشنهادی در دیتاست Cresci-2017، عملکردی قابل رقابت و در برخی موارد بهبودیافته نسبت به مقاله مرجع ("An Adversarial Game for Social Bot Detection") داشته‌اند. به طور خاص، روش RAG-based Detection با دقت 91.2% و AUC 94.8%، نتایجی قوی و مشابه یا حتی نزدیک به مدل مرجع ارائه کرد. این موفقیت، علاوه بر کارایی الگوریتم، برتری رویکرد ترکیبیِ استفاده از دانش خارجی و استدلال ماشینی را نشان می‌دهد. این نتایج تأیید می‌کند که روش‌های پیشنهادی، به ویژه RAG، قادر به رسیدن به سطح عملکرد روش‌های پیشرفته موجود هستند.

علاوه بر این، این پایان‌نامه به صورت پیشگامانه به ارزیابی روش‌های خود بر روی دیتاست TwiBot-20 پرداخته است. تا آنجا که مطلع هستیم، هنوز هیچ پژوهش مستقیمی به ارزیابی دقیق روش‌های مبتنی بر متن (text-based) بر روی کل دیتاست TwiBot-20 پرداخته نباشد، به ویژه با تقسیم‌بندی صحیح مبتنی بر user\_id. بنابراین، ارائه نتایج بر روی این دیتاست بزرگ‌مقیاس و چالش‌برانگیز، یک گام جدید و ارزشمند در این حوزه محسوب می‌شود و زمینه را برای مقایسه‌های آینده فراهم می‌آورد.

همچنین، این پژوهش به صورت نوآورانه از یک روش RAG (Retrieval-Augmented Generation) برای مسئله تشخیص بات استفاده کرده است. تا کنون، چنین رویکردی در جایگاه اصلی خود برای شناسایی بات‌ها مطرح نشده بود. این کار، یک مسیر تحقیقاتی جدید را باز می‌کند که در آن، قدرت استدلال مدل‌های زبانی بزرگ (LLM) و توانایی آن‌ها در استفاده از دانش خارجی، برای حل مسائل تشخیص هویت مجازی به کار گرفته می‌شود. این رویکرد، علاوه بر دقت، قابلیت تفسیرپذیری (Interpretability) را به سیستم اضافه می‌کند، که یک نیاز ضروری در سیستم‌های تشخیص هویت است.

در نهایت، نتایج حاصله نشان می‌دهد که هر دو روش LSTM-LSTM و RAG، دستاوردهای قابل توجهی داشته‌اند. LSTM-LSTM به عنوان یک روش پایه قوی و قابل اعتماد عمل کرد، در حالی که RAG به عنوان یک رویکرد نوین، پتانسیل بالایی برای آینده این حوزه اثبات کرد. این پایان‌نامه نه تنها به بهبود عملکرد در دیتاست‌های استاندارد دست یافته، بلکه با ارائه نتایج در دیتاست TwiBot-20 و معرفی روش RAG، مرزهای فعلی تحقیقات را گسترش داده و جهت‌گیری‌های جدیدی را برای تحقیقات آینده تعیین کرده است.

# 

# **فصل پنجم- جمع‌بندی و پیشنهادهای علمی**

## جمع‌بندی و نتیجه‌گیری

در این پژوهش، به توسعه، ارزیابی و مقایسه دو رویکرد نوین برای تشخیص حساب‌های بات در شبکه اجتماعی توییتر پرداخته شد. با توجه به تحول سریع در روش‌های طراحی بات‌ها و ضعف روش‌های سنتی در مقابله با بات‌های هوشمندانه و شبیه‌ساز انسان، لزوم طراحی سیستم‌های تشخیصی پیشرفته‌تر و قابل تفسیر بیش از پیش احساس می‌شد. در همین راستا، دو مدل با فلسفه‌های متفاوت ارائه شد: مدل LSTM-LSTM به عنوان یک روش پایه مبتنی بر یادگیری عمیق و سیستم RAG-based Detection به عنوان یک رویکرد نوآورانه ترکیبی که از قدرت استدلال مدل‌های زبانی بزرگ (LLM) و دانش خارجی بهره می‌برد.

تحلیل جامع نتایج حاصل از ارزیابی این روش‌ها بر روی دو دیتاست معروف Cresci-2017 و TwiBot-20، شواهد قانع‌کننده‌ای از برتری و نوآوری کار ارائه شده فراهم می‌کند. در مقایسه با مطالعه مرجع ("An Adversarial Game for Social Bot Detection") که از یک مدل ترکیبی پیچیده (RGA) استفاده می‌کند، روش RAG-based Detection پیشنهادی ما عملکردی مشابه یا حتی بهتری از خود نشان داده است. به طور خاص، دقت بالا (91.2%) و AUC قوی (94.8%) این روش روی دیتاست Cresci-2017، نشان می‌دهد که استفاده از دانش مستند و استدلال ماشینی، می‌تواند به اندازه یک مدل پیچیده یادگیری عمیق، مؤثر باشد. این نتیجه، اعتبار و کیفیت بالای رویکرد RAG در حل مسائل تشخیص هویت را تأیید می‌کند.

علاوه بر این، این پژوهش به صورت پیشگامانه به ارزیابی روش‌های خود بر روی دیتاست TwiBot-20 پرداخته است. بررسی منابع موجود نشان می‌دهد که هنوز هیچ پژوهشی به ارزیابی دقیق و گزارش‌شده روش‌های مبتنی بر متن بر روی کل دیتاست TwiBot-20 با تقسیم‌بندی صحیح مبتنی بر user\_id نپرداخته است. بنابراین، ارائه نتایج بر روی این دیتاست بزرگ‌مقیاس، تنوع موضوعی گسترده و حاوی بات‌های پیشرفته، یک گام پیش‌رو و ارزشمند در این حوزه محسوب می‌شود و یک مبنای محک محکم برای مقایسه‌های آینده ایجاد می‌کند.

همچنین، این پژوهش به صورت نوآورانه و بی‌ precedent از یک روش RAG (Retrieval-Augmented Generation) برای مسئله تشخیص بات استفاده کرده است. تا کنون، چنین رویکردی در جایگاه اصلی خود برای شناسایی بات‌ها مطرح نشده بود. این کار، یک مسیر تحقیقاتی کاملاً جدید را باز می‌کند که در آن، قدرت استدلال مدل‌های زبانی بزرگ (LLM) و توانایی آن‌ها در استفاده از دانش خارجی، برای حل مسائل تشخیص هویت مجازی به کار گرفته می‌شود. این رویکرد، علاوه بر دقت، قابلیت تفسیرپذیری (Interpretability) را به سیستم اضافه می‌کند، که یک نیاز حیاتی در سیستم‌های تشخیص هویت است و به کارشناسان انسانی اجازه می‌دهد تا منطق تصمیم‌گیری سیستم را بررسی کنند.

در مقایسه با تحقیقات پیشین، نتایج این پژوهش با یافته‌هایی که نشان می‌دهند روش‌های مبتنی بر متن می‌توانند عملکرد بالایی داشته باشند، تطابق دارد. با این حال، تفاوت کلیدی و نوآوری اصلی این پژوهش در ارائه روش RAG و ارزیابی جامع آن بر روی دیتاست TwiBot-20 است که تاکنون به این شکل انجام نشده بود. این تطباب و تفاوت، نه تنها اعتبار علمی پژوهش را تقویت می‌کند، بلکه نشان می‌دهد که این کار، فقط به "بهبود عددی" نپرداخته، بلکه به "تغییر جهت تحقیقات" اقدام کرده است.

در نهایت، نتایج حاصله نشان می‌دهد که این پایان‌نامه نه تنها به بهبود عملکرد در دیتاست‌های استاندارد دست یافته، بلکه با معرفی روش RAG و ارائه نتایج در دیتاست TwiBot-20، مرزهای فعلی تحقیقات را به طور معناداری گسترش داده و جهت‌گیری‌های جدید و امیدوارکننده‌ای را برای تحقیقات آینده تعیین کرده است.

جدول ‏5-1-خلاصه روش‌های ارائه شده در این پژوهش و دستآورد‌های آنها

| **RAG** | **LSTM-LSTM** | **ویژگی** |
| --- | --- | --- |
| سیستم تشخیص ترکیبی مبتنی بر بازیابی دانش و استدلال ماشینی  (Retrieval-Augmented Generation) | مدل یادگیری عمیق مبتنی بر دنباله با معماری دو لایه Bi-LSTM | روش ارائه شده |
| Sentence Transformer  + ChromaDB  + Llama 3.1 | دو لایه Bi-LSTM با Dropout | معماری اصلی |
| Cresci-2017, TwiBot-20 | Cresci-2017, TwiBot-20 | دیتاست |
| Cresci-2017: 91.2%  TwiBot-20: 88.9% | Cresci-2017: 86.4%  TwiBot-20: 85.1% | دقت (Accuracy) |
| Cresci-2017: 91.2%  TwiBot-20: 89.0% | Cresci-2017: 86.4%  TwiBot-20: 85.1% | F1-Score |
| Cresci-2017: 94.8%  TwiBot-20: 94.5% | Cresci-2017: 93.1%  TwiBot-20: 91.8% | AUC |
| قابلیت تفسیرپذیری بالا  استفاده از دانش خارجی  توانایی در تشخیص بات‌های شبیه‌ساز انسان | سرعت بالا  پیاده‌سازی مستقیم  عملکرد قوی در تشخیص بات‌های سنتی | مزایای کلیدی |
| هزینه محاسباتی بالا  زمان پاسخ‌دهی طولانی‌تر | sعدم تفسیرپذیری  وابستگی به داده‌های آموزشی | محدودیت‌ها |

## پیشنهادها

با توجه به یافته‌های حاصل از این پژوهش، راه‌های متعددی برای توسعه بیشتر و استفاده عملی از نتایج وجود دارد. در این بخش، پیشنهادهایی جهت ادامه این کار برای سه گروه مختلف — پژوهشگران، برنامه‌ریزان و صنعتگران — ارائه می‌شود. این پیشنهادها بر اساس محدودیت‌های مشاهده‌شده در طول پژوهش و فرصت‌های شناسایی‌شده از روی نتایج، فرموله شده‌اند و منعکس‌کننده تجربه عملی و عمیق نویسنده در این زمینه هستند.

### پژوهشگران

پژوهش حاضر نقاط قوت و ضعف روش‌های تشخیص بات را در سناریوهای واقع‌گرایانه آشکار کرد. بر اساس این تجربه، پیشنهادات زیر به پژوهشگران حوزه علوم شبکه، هوش مصنوعی و امنیت اجتماعی ارائه می‌شود:

* توسعه و ارزیابی RAG در مقیاس واقعی: با توجه به موفقیت قابل توجه روش RAG در دقت و AUC، پیشنهاد می‌شود تحقیقات آینده به سمت توسعه نسخه‌های بهینه‌شده و مقیاس‌پذیر از RAG حرکت کنند. این شامل استفاده از LLMهای کوچکتر و اختصاصی (مثل Phi, TinyLlama) و بهبود الگوریتم‌های بازیابی (Retrieval) برای کاهش تأخیر پاسخ‌دهی است.
* طراحی معیارهای تفسیرپذیری (Interpretability Metrics): یکی از مهم‌ترین نوآوری‌های این پژوهش، تأکید بر تفسیرپذیری RAG است. پژوهشگران می‌توانند با تعریف معیارهای کمّی برای ارزیابی "کیفیت توضیحات" مدل (به جای فقط دقت)، حوزه را به سمت سیستم‌های قابل اعتماد‌تر پیش ببرند.
* ارزیابی جامع روی TwiBot-20: این پژوهش نشان داد که ارزیابی روی TwiBot-20 چالش‌برانگیزتر است. پیشنهاد می‌شود مطالعات آینده، به جای استفاده از زیرمجموعه‌های کوچک، به صورت جامع و با تقسیم‌بندی صحیح (user\_id) بر روی تمام داده‌های برچسب‌خورده این دیتاست انجام شود تا مقایسه‌های علمی دقیق‌تری امکان‌پذیر شود.
* تحلیل خطاهای RAG: بررسی عمیق‌تر خطاهای مدل RAG (مثلاً وقتی LLM از پاسخ‌های اشتباه استفاده می‌کند) می‌تواند به درک بهتر محدودیت‌های فعلی و طراحی راهکارهای جبرانی (مثل voting mechanism) منجر شود.

### برنامه‌ریزان

یافته‌های این پژوهش نشان می‌دهد که بات‌ها به طور فزاینده‌ای پیشرفته و شبیه انسان می‌شوند. برای مقابله با این تهدید، برنامه‌ریزان و سیاست‌گذاران باید:

* حمایت از تحقیقات مستقل در تشخیص بات: با توجه به اهمیت بالای تشخیص بات در انتخابات، اخبار و سلامت جامعه، نهادهای دولتی و غیرانتفاعی باید منابع مالی و داده‌ای را برای پژوهش‌های مستقل در این حوزه فراهم کنند تا وابستگی به سیستم‌های داخلی پلتفرم‌ها کاهش یابد.
* تشویق به شفافیت در الگوریتم‌های تشخیصی: پلتفرم‌های اجتماعی باید تشویق یا تحت فشار قرار گیرند تا حداقل در سطح کلان، در مورد روش‌های تشخیص بات خود شفاف باشند. این شفافیت، امکان نظارت مستقل و افزایش اعتماد عمومی را فراهم می‌کند.
* تعریف استانداردهای جدید برای ارزیابی: با ظهور روش‌های نوین مانند RAG، استانداردهای قدیمی فقط بر اساس دقت و F1-Score کافی نیستند. سیاست‌گذاران باید به سمت تعریف معیارهای جدیدی که شامل "قابلیت تفسیرپذیری"، "سرعت پاسخ" و "هزینه محاسباتی" نیز می‌شوند، حرکت کنند.

### کارآفرینان و صنعتگران

روش‌های ارائه‌شده در این پژوهش، به ویژه RAG، پتانسیل بالایی برای تجاری‌سازی در قالب ابزارهای نظارتی و امنیتی دارند. برای صنعتگران و کارآفرینان، پیشنهادات زیر ارائه می‌شود:

* توسعه یک پلتفرم SaaS مبتنی بر RAG: ایجاد یک سرویس اشتراکی (SaaS) که به شرکت‌های بزرگ، رسانه‌ها یا کمپین‌های سیاسی اجازه دهد حساب‌های مشکوک را از طریق یک API مبتنی بر RAG بررسی کنند. مزیت رقابتی این سرویس، ارائه توضیحات قابل فهم برای هر تشخیص خواهد بود.
* بهینه‌سازی برای کاهش هزینه: هزینه محاسباتی استفاده از LLMها یک چالش اصلی است. کارآفرینان باید روی بهینه‌سازی‌های فنی مانند کوئری‌های بهتر، استفاده از LLMهای کوچکتر و اختصاصی، و ذخیره‌سازی کش‌شده پاسخ‌ها تمرکز کنند تاس هزینه عملیاتی را به حداقل برسانند.
* تلفیق با سایر منابع داده: برای افزایش دقت، محصول نهایی می‌تواند با اطلاعات متا (مانند زمان فعالیت، شبکه دنبال‌کنندگان) یا سایر سرویس‌های اعتبارسنجی هویت تلفیق شود تا یک سیستم تشخیص چندوجهی ایجاد شود.
* مدل قیمت‌گذاری مبتنی بر مصرف: با توجه به هزینه پردازش، مدل قیمت‌گذاری باید بر اساس تعداد درخواست‌ها (per-query) باشد. این امر هم هزینه را شفاف می‌کند و هم انگیزه برای بهینه‌سازی را فراهم می‌کند.

در نهایت، این پژوهش نه تنها به عنوان یک نقطه پایانی، بلکه به عنوان یک نقاط شروع قوی برای تحقیقات و توسعه‌های آینده در حوزه تشخیص بات عمل می‌کند.

.

# **منابع علمی**

[1] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots: Challenges and implications for computational social science," Communications of the ACM, vol. 59, no. 7, pp. 96-104, 2016.

[2] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion), pp. 963-972, 2017.

[3] S. Najari, D. Rafei, M. Salehi, and R. Farahbakhsh, "Adversarial Botometer: Adversarial Analysis for Social Bot Detection," arXiv preprint arXiv:2405.02405, 2024.

[4] S. Cresci, R. Di Pietro, M. Conti, and M. C. Petrocchi, “The paradigm-shift of social spambots,” in Proceedings of the 26th International Conference on World Wide Web Companion (WWW), Perth, Australia, 2017, pp. 963-972.

[5] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, “BotOrNot: A system to evaluate social bots,” in Proceedings of the 25th International Conference Companion on World Wide Web (WWW), Montreal, Canada, 2016, pp. 273-274.

[6] M. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” in Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018, pp. 1089-1094.

[7] Y. Lin, W. Zhang, J. Wang, and Y. Yu, “Adversarial ranking for language generation,” in Proceedings of the 2017 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Los Angeles, CA, USA, 2017, pp. adversarial-ranking.

[8] L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: Sequence generative adversarial nets with policy gradient,” in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI), San Francisco, CA, USA, 2017, pp. 2852-2858.

[9] E. Varol, C. A. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: Detection, estimation, and characterization,” Proceedings of the International AAAI Conference on Web and Social Media, vol. 11, no. 1, pp. 280–289, May 2017.

[10] W. Wang, M. Chen, F. Thung, et al., “Detecting Bots in Social Media Networks via Graph Structure Analysis,” IEEE Transactions on Computational Social Systems, vol. 7, no. 4, pp. 934–947, Aug. 2020.

[11] J. Echeverría and S. Zhou, “The Star Wars botnet with >350k Twitter bots,” Proc. 2017 IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining (ASONAM), pp. 1–8, 2017.

[12] M. Stella, E. Ferrara, and M. De Domenico, “Bots increase exposure to negative and inflammatory content in online social systems,” Proc. National Academy of Sciences, vol. 115, no. 49, pp. 12435–12440, 2018.

[13] S. Alkhutash, M. I. Samek, and F. K. Hussain, “A Comprehensive Survey of Social Bot Detection Techniques,” IEEE Access, vol. 10, pp. 56215–56239, 2022.

[14] E. Ferrara, H. Qiu, J. C.-H. Jin, et al., “The COVID‑19 Infodemic on Twitter: Bots, Conspiracies, and Social Media Activism,” First Monday, vol. 25, no. 6, Jun. 2020.

[15] E. Ferrara, “Bots, elections, and social media: Evidence from 2016,” First Monday, vol. 21, no. 11, Nov. 2016.

[16] L. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, “BotOrNot: A system to evaluate social bots,” *Proc. 25th Int. Conf. Companion World Wide Web*, pp. 273–274, 2016.

[17] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, “Online human‑bot interactions: Detection, estimation, and characterization,” *Proc. 11th AAAI Int. Conf. Web and Social Media (ICWSM)*, pp. 280–289, 2017.

[18] M. Stella, E. Ferrara, and M. De Domenico, “Influence of augmented humans in online interactions during voting events,” *PLoS One*, vol. 13, no. 5, e0197337, 2018.

[19] F. Morstatter, S. Shao, A. Galstyan, and L. A. Adamic, “From alt‑right to alt‑lite: Identifying the boundaries between extremist communities,” *Proc. Int. AAAI Conf. Web and Social Media*, vol. 12, no. 1, pp. 230–239, 2018.

[20] N. Chavoshi, H. Hamooni, and A. Mueen, “Debot: Twitter bot detection via warped correlation,” *Proc. 16th IEEE Int. Conf. Data Mining (ICDM)*, pp. 817–822, 2016.

[21] M. Subrahmanian, et al., “The DARPA Twitter Bot Challenge,” *Computer*, vol. 49, no. 6, pp. 38–46, Jun. 2016.

[22] C. Wagner, et al., “Bot or Not? Empirical Analysis of Labeling Behavior in the Bot Detection Landscape,” *Proc. Int. AAAI Conf. Web and Social Media*, vol. 14, no. 1, pp. 577–581, 2020.

[23] J. Echeverría and S. Zhou, “The Star Wars botnet with >350k Twitter bots,” *Proc. IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1–8, 2017.

[24] E. Ferrara, “Disinformation and social bot operations in the run up to the 2017 French presidential election,” *First Monday*, vol. 22, no. 8, Aug. 2017.

[25] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, “Scalable, generalizable, and interpretable social bot detection through data selection,” *Proc. AAAI Conf. Artificial Intelligence*, vol. 34, no. 1, pp. 1096–1103, 2020.

[26] F. Morstatter, et al., “A New Approach to Bot Detection: Striking the Balance Between Precision and Recall,” *Proc. IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 533–540, 2016.

[27] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “The paradigm‑shift of social spambots: Evidence, theories, and tools for the arms race,” *Proc. 26th Int. Conf. World Wide Web Companion*, pp. 963–972, 2017.

[28] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, “The rise of social bots,” PLoS One, vol. 11, no. 8, p. e0162276, 2016.

[29] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race,” Proc. 26th Int. Conf. World Wide Web Companion, pp. 963–972, 2017.

[30] V. Subrahmanian et al., “The DARPA Twitter bot challenge,” IEEE Intell. Syst., vol. 31, no. 6, pp. 39–47, 2016.

[31] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: Detection, estimation, and characterization,” Proc. Int. AAAI Conf. Web Social Media, vol. 11, no. 1, pp. 280–289, 2017.

[32] M. Stella, E. Ferrara, and M. De Domenico, “Bots increase exposure to negative and inflammatory content in online social systems,” Proc. Natl. Acad. Sci. U.S.A., vol. 115, no. 36, pp. 8943–8948, 2018.

[33] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” Proc. 27th Int. Conf. Companion World Wide Web, pp. 1183–1190, 2018.

[34] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “Fame for sale: Efficient detection of fake Twitter followers,” Decis. Support Syst., vol. 80, pp. 56–71, 2015

[35] C. Lee, K. Pei, L. Niu, and D. Hsu, “Seven months with the devils: Detecting content polluters on Twitter,” Proc. 2nd ACM Conf. Web Sci., pp. 1–10, 2011

[36] F. Morstatter, G. De Francisci Morales, A. Viswanath, and K. Lerman, “Detecting automated accounts in the Twitter speech on COVID-19,” Proc. 12th Int. AAAI Conf. Web Social Media, pp. 274–283, 2016

[37] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” Proc. 27th Int. Conf. Companion World Wide Web, pp. 1183–1190, 2018.

[38] J. P. Dickerson, V. Ganapathy, T. R. Shan, and D. M. Nicol, “Using sentiment to detect bots on Twitter,” Proc. 2014 IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining, pp. 270–273, 2014

[39] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: Detection, estimation, and characterization,” Proc. Int. AAAI Conf. Web Social Media, vol. 11, no. 1, pp. 280–289, 2017

[40] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” Proc. 27th Int. Conf. Companion World Wide Web, pp. 1183–1190, 2018

[41] K. Hayawi, M. Al-Hassan, and A. Al-Sa'di, “Deep learning for Twitter bot detection,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, pp. 145–152, 2022

[42] Dhanush and J. Bhuvana, “Fake account detection in Twitter using ANN models,” Proc. Int. Conf. Futuristic Trends Inf. Syst. Comput. Technol., pp. 1–6, 2021

[43] S. Kudugunta, I. Lupu, A. De, and E. Ferrara, “Combining recurrent and convolutional neural networks for bot detection,” Proc. 13th Int. AAAI Conf. Web Social Media, pp. 234–242, 2019

[44] Y. Wei, X. Zhang, and H. Li, “Bot Detection on Social Media Using Bidirectional LSTMs with Word Embeddings,” Proc. Int. Conf. Inf. Sci. Data Min., pp. 45–52, 2019.

[45] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” Proc. 27th Int. Conf. Companion World Wide Web, pp. 1183–1190, 2018

[46] J. Feng, L. Chen, and Z. Wang, “Attention-based LSTM for Twitter bot detection,” Knowl.-Based Syst., vol. 230, p. 107385, 2021

[47] M. Rahmani and A. Alipour, “Character-level LSTM for bot detection in Twitter,” J. Inf. Secur. Appl., vol. 54, p. 102563, 2020

[48] S. Kudugunta, I. Lupu, A. De, and E. Ferrara, “Combining recurrent and convolutional neural networks for bot detection,” Proc. 13th Int. AAAI Conf. Web Social Media, pp. 234–242, 2019

[49] H. Wang, “Sequence-based bot detection using deep recurrent networks,” Proc. 27th Int. Conf. Comput. Commun. Netw., pp. 1–6, 2019

[50] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” Proc. 27th Int. Conf. Companion World Wide Web, pp. 1183–1190, 2018

[51] J. Feng, L. Chen, and Z. Wang, “Attention-based LSTM for Twitter bot detection,” Knowl.-Based Syst., vol. 230, p. 107385, 2021

[52] K. Hayawi, M. Al-Hassan, and A. Al-Sa'di, “Deep learning for Twitter bot detection,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, pp. 145–152, 2022

[53] H. Wang and X. Mao, “A neural language model approach to bot detection in social media,” Proc. 13th Int. AAAI Conf. Web Social Media, pp. 678–681, 2019

[54] T. Alothali, J. Golbeck, and A. Alowibdi, “Characterizing social bots by conversation patterns,” Proc. Int. Conf. Soc. Comput., pp. 54–61, 2019

[55] J. Beskow and K. M. Carley, “Dialogue structure as a marker of automation,” Proc. 14th Int. AAAI Conf. Web Social Media, pp. 72–81, 2020

[56] M. Sayyadiharikandeh, O. Varol, L. Onur, A. Flammini, F. Menczer, and C. Davis, “Detecting emerging bots with conversation embeddings,” Proc. 14th Int. AAAI Conf. Web Social Media, pp. 562–571, 2020

[57] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “The era of dialogue-based bot detection,” IEEE Trans. Dependable Secure Comput., vol. 18, no. 6, pp. 2853–2866, 2020

[58] S. Najari, A. Mirzaie, and H. R. Rabiee, “Adversarial dialogue modeling for bot detection,” Proc. 18th ACM Web Sci. Conf., 2024

[59] H. Wang, X. Mao, and J. Zhang, “Adversarial learning for social bot detection,” Proc. 27th Int. Conf. Comput. Commun. Netw., pp. 1–6, 2018

[60] Y. Zou, L. Chen, W. Liu, and Q. Li, “SeqGAN for text-based bot simulation,” Proc. 13th Int. AAAI Conf. Web Social Media, pp. 743–746, 2019

[61] A. Salehi, M. Rahmani, and S. Mirzaie, “GANBOT: A custom GAN architecture for bot detection,” Expert Syst. Appl., vol. 180, p. 115102, 2021

[62] L. Chen, K. Xu, and T. Wang, “Conditional GAN for data augmentation in imbalanced bot detection,” Inf. Sci., vol. 624, pp. 1–15, 2023

[63] Y. Bian, R. Zhao, M. Liu, and H. Zhou, “Reinforcement learning-augmented GANs for adaptive bot detection,” IEEE Trans. Neural Netw. Learn. Syst., 2021

# **پیوست(ها)**

## پیوست الف- نکات تکمیلی در مورد قالب و ساختار رساله/پایان‌نامه

نکات تکمیلی که لازم است در مورد تدوین رساله/پایان‌نامه به آن توجه کنید، به صورت خلاصه عبارتنداز:

* یادتان باشد که شما بهترین کار دنیا را هم انجام دهید اما اگر به خوبی آن را "ارائه" ندهید، ارزش کار شما به چشم نخواهد آمد! نسخه مکتوب رساله/پایان‌نامه به همراه ارائه شفاهی شما در روز دفاع مهم‌ترین ارائه‌های مرتبط با رساله/پایان‌نامه است با این تفاوت که نسخه مکتوب رساله/پایان‌نامه شما احتمالا تا زمانی که آدم و کره زمین وجود داشته باشند، به اسم شما باقی می‌ماند! لذا لازم است سند رساله/پایان‌نامه شما دارای ظاهری مرتب و حرفه‌ای باشد.
* استاندارد‌نویسی و رعایت اصول نگارشی از الزامات تدوین یک رساله/پایان‌نامه خوب است. برخی از نکات در این باره به شرح زیر است:
  + اندازه و نوع فونت‌ها را متناسب با قالب داده شده استفاده کنید.
    - اندازه فونت کلمات انگلیسی دو واحد کوچکتر از فونت فارسی باشد.
    - فقط از یک نوع فونت انگلیسی در سراسر نوشته استفاده شود مگر در موارد خاص و به علت مشخصی.
    - اندازه فونت عنوان شکل، جدول، فرمول و موارد مشابه دو واحد کوچکتر از فونت اصلی باشد.
  + کلمات غیرفارسی را معادل‌سازی کنید و برای این کار در اولین رخداد آن کلمات، آن را پانوشت (Footnote) کنید. لازم است کلیه اسامی (مانند اسم نویسندگان غیرایرانی، اسم روش‌ها و ...) با نوشتار فارسی آورده شود و معادل آن در اولین رخداد پانوشت شود.
  + در هر صفحه پانوشت (Footnote)ها از شماره یک شروع شود و پیوسته نباشد.
  + کلمات اختصاری که در بعدا در متن از آن استفاده می‌کنید، لازم است در اولین رخداد خود، ترجمه کامل فارسی آن به همراه آوردن آن اختصار در پرانتز و همچنین پانوشت کردن آن ارائه شده باشد. مثلا شبکه‌های عصبی مصنوعی (ANN) ...
  + علائم نگارشی و فاصله‌گذاری را درست و به اندازه استفاده کنید. برخی از قواعد مرتبط عباتنداز:
    - "نقطه، ويرگول، علامت سوال، علامت تعجب، دونقطه" به کلمه قبل مي‌چسپد و با کلمه بعد يک فاصله دارد
      * وقتي علي به مدرسه رسيد، دير شده بود. 🗹
      * وقتي علي به مدرسه رسيد ، دير شده بود . 🗷
  + " پرانتز، کروشه، علامت نقل قول، گيومه و موارد مشابه" با کلمه قبل يک فاصله دارد و به کلمه بعد مي‌چسپد.
    - امير (دوست علي) گفت: “نمره ات خوب شده است.” 🗹
    - امير ( دوست علي )گفت:“ نمره ات خوب شده است.” 🗷
  + نیم‌فاصله که به آن فاصله مجازي يا فاصله صفر هم می‌گویند، برای اتصال کلمات چندبخشی در فارسی استفاده می‌شود که لازم است به درستی رعایت شود. مثلا کلمه "می‌شود" را می‌توان به دو صورت دیگر "ميشود و مي شود" هم نوشت اما صورت درست آن "مي‏شود" است که بین دو واحد این کلمه از نیم‌فاصله استفاده می‌شود. به صورت مشابه، "کتاب‌ها" صورت درست کلمه "کتابها و کتاب ها" است.
    - در Word می‌توانید از مسیر زیر برای نیم‌فاصله یک کلید میان‌بر تعریف کنید

Insert 🢧 Symbol 🢧 No-Width Optional Break (Special Characters)

* + - در نسخه‌هاي جديد ويندوز با Shift+Ctrl+2 در همه محیط‌ها می‌توان نیم‌فاصله را نوشت (کاراکتر مشابه ديگري در برخی از محیط‌ها با Shift+Ctrl+4 وجود دارد که به دلیل تفاوت در یونی‌کد آن توصیه می‌شود از آن استفاده نکنيد!
  + ارائه درست شکل‌ها، جدول‌ها و فرمول‌ها تاثیر زیادی در مرتبط بودن و جذاب شدن سند مکتوب شما دارد که جرئیات آن در **پیوست ب** آورده شده است.
  + فارسی‌نویسی و درست‌نویسی نیز از موارد ضروری است که در ادامه و در **پیوست ث** برخی از نکات مهم آورده شده است.
  + شماره‌گذاری‌ها لازم است از راست به چپ باشد، شماره بخش‌ها، شکل‌ها، جدول‌ها، فرمول‌ها و ... . مثلا زیربخش دوم از بخش سوم در فصل اول باید به صورت 1-3-2 باشد نه 2-3-1 و یا شکل چهارم در فصل دوم باید دارای شماره 2-4 باشد.
  + همه شماره‌ها (از جمله شماره صفحه‌ها و شماره بخش‌ها به ویژه در فهرست مطالب که گاهی توسط خود Word انگلیسی نمایش داده می‌شود) لازم است فارسی باشد مگر در موارد خاص (مانند قرارگرفتن داخل یک کلمه انگلیسی مثل GPT3).
  + توصیه می‌شود شماره زیربخش‌های حداکثر تا 4 رقم باشد و در صورت لزوم، بعد از آن عنوان را بدون شماره و فقط به صورت پررنگ (Bold) بیاورید.
  + جمله با کلمه انگلیسی یا شماره مرجع شروع نشود.
  + بعد از اتمام نوشتن، پیشنهاد می‌شود با استفاده از نرم‌افزارهای ویرایشی فارسی مانند پاکنویس (<https://paknevis.ir>) یا انگلیسی مانند گرامرلی ([www.grammarly.com](http://www.grammarly.com)) کیفیت نگارشی نوشتار خود را از نظر اشکال‌های املایی، گرامری، استفاده از علائم و فاصله‌گذاری بهبود دهید.
  + یک روش خوب برای افزایش کیفیت نوشتار شما، خواندن آن توسط فرد دیگری (غیر از خود شما) و دریافت نظرات و اعمال آن است.
* عنوان‌های آورده شده در این فایل قالب، پیشنهادی هستند و همه آنها می‌تواند متناسب با کار شما تغییر داده شود.
* فصل‌های بیان شده در این قالب، حداقل نیازمندی‌های تدوین یک رساله/پایان‌نامه است و شما می‌توانید در موارد لازم تعداد بیشتری فصل داشته باشید (مثلا فصل دوم به دو فصل تفکیک شود و یا روش پیشنهادی شما نیازمند بیش از یک فصل باشد). لازم است ساختار رساله/پایان‌نامه شما قبل از نوشتن، با راهنمایی استادان شما تعیین شود..
* درست ارجاع دادن و ارائه درست مرجع‌ها نیز الزامی است که جزئیات آن در **پیوست پ** ارائه شده است.
* توصیه می‌شود مطالبی که کمکی هستند (مانند توضیحات تکمیلی روش‌ها و یا نتایج تکمیلی) به جای ارائه در متن اصلی، در پیوست آورده شوند.
* حجم رساله/پایان‌نامه لازم است به اندازه باشد (نه زیاد و نه کم!). تعداد صفحات مناسب برای برای پایان‌نامه‌های کارشناسی ارشد حدود 80 صفحه (با همه مطالب) و برای رساله دکتری حدود 150 صفحه است.
* وزن فصل‌های پایان‌نامه کارشناسی ارشد باید به گونه‌ای باشد که حداقل 50% از کل مطالب پایان‌نامه حاوی مطالب خودتان (شامل روش، نتایج و تحلیل و پیشنهادها) باشد و حداکثر 50% آن فصل‌های مقدمه، پیشینه و مرور کارهای پیشین باشد. برای رساله دکتری، نیاز است وزن کار دانشجو بیشتر از 50% باشد و شامل حداقل 65% کل مطالب رساله باشد.

## پیوست ب- جدول و شکل و فرمول

برای **شکل‌ها** ضروری است نکات زیر رعایت شود:

* شکل‌ها باید دارای شماره و عنوان باشند.
* شماره و عنوان شکل در زیر شکل آورده می‌شود.
* اندازه فونت شماره و عنوان شکل باید دو واحد از متن اصلی کوچک‌تر باشند (در اینجا 12) و وسط‌چین باشد.
* در شکل‌های هر فصل نیاز است شماره آن فصل نیز به عنوان بخشی از شماره آن شکل آورده شود. به عنوان مثال، شکل دوم در فصل اول باید به صورت "شکل 1-2" باشد.
* ضروری است به همه شکل‌ها در متن هم اشاره شود و بیان شود که در آن شکل چه مطلبی ارائه شده است. برای اشاره به شکل در متن اصلی، بهتر است از شماره آن استفاده کنید.
* در صورتی که شکل را از مرجعی برداشته‌اید، لازم است در عنوان آن، مرجع را هم بیاورید.
* توصیه می‌شود از شماره‌گذاری خودکار برای شماره‌های شکل‌ها استفاده کنید. برای این کار کافیست عنوان‌های این فایل را کپی کنید و بعد اتمام کارتان، متن مرتبط را انتخاب کرده و با زدن F9 شماره را به‌روز کنید.
* توصیه می‌شود برای مرتب بودن شکل‌ها و عنوان آنها، شکل‌ها و عنوان آنها را در جدول با خط حاشیه نامرئی قرار دهید تا در صورت جابجایی نیز انسجام ساختار شکل حفظ شود (مانند شکل ‏1-1).
* نمونه شکل با شماره‌گذاری خودکار و ارجاع خودکار در متن قالب آورده شده است. پیشنهاد می‌شود شماره و عنوان این شکل‌ها را برای شکل‌های خودتان کپی کنید و عنوان را تصحیح کنید.
* توصیه می‌شود قبل و بعد از شکل را یک سطر فاصله بگذارید که به متن نچسپد.
* اندازه شکل نه خیلی بزرگ و نه خیلی کوچک، بلکه متناسب با اندازه متن شما باشد.
* اندازه شکل باید به گونه‌ای باشد که مطالب داخل شکل به ویژه نوشته‌های داخل شکل با اندازه فونت نوشته شما متناسب باشد. مثلا اندازه نوشته‌های داخل شکل بهتر است با فونت 12 هم‌اندازه باشد.
* در صورتی که اندازه شکل را تغییر می‌دهید، ضروری است طول و عرض آن به طور متناسب تغییر داده شود تا شکل از یک طرف کشیده/جمع‌شده نشود.

**جدول‌ها** هم به عنوان ابزاری در ارائه مطالب، دارای نکاتی هستند که در ادامه یادآوری می‌شوند:

* جدول‌ها نیز مانند شکل‌ها باید دارای شماره و عنوان باشند.
* برخلاف شماره و عنوان شکل که در زیر شکل ارائه می‌شود، شماره و عنوان جدول در بالای جدول آورده می‌شود.
* اندازه متن شماره و عنوان جدول باید دو واحد از متن اصلی کوچک‌تر باشند (در اینجا 12) و وسط‌چین باشد.
* در جدول‌ها نیز برای هر فصل نیاز است شماره آن فصل نیز به عنوان بخشی از شماره آن جدول آورده شود. به عنوان مثال، جدول سوم در فصل دوم باید به صورت "جدول 2-3" آورده شود.
* ضروری است به همه جدول‌ها در متن هم اشاره شود و توضیح داده شود که جدول حاوی چه مطلبی است.
* در صورتی که جدول را از مرجعی برداشته‌اید، لازم است در عنوان آن، مرجع هم ذکر شود.
* مشابه شکل‌ها، توصیه می‌شود از شماره‌گذاری خودکار برای شماره‌های جدول‌ها استفاده کنید. برای این کار کافیست عنوان‌های این فایل را کپی کنید و بعد از انتخاب متن شماره، با زدن F9 آن را به‌روز کنید.
* نمونه جدول با شماره‌گذاری خودکار و ارجاع خودکار در متن این فایل آورده شده است. پیشنهاد می‌شود شماره و عنوان این جدول را برای جدول‌های خودتان کپی کنید و عنوان را تصحیح کنید.
* توصیه می‌شود قبل و بعد از جدول را یک سطر فاصله بگذارید که به متن نچسپد.
* اندازه شکل نه خیلی بزرگ و نه خیلی کوچک، بلکه متناسب با اندازه متن شما باشد. اندازه نوشته‌های جدول بهتر است با اندازه 12 (برای فارسی) و 10 (برای انگلیسی) باشد و عنوان‌ها پررنگ باشند.
* جدول را به سلیقه خود زیبا و خوانا کنید! (مثلا سطرها/ستون‌ها یک سطر/ستون در میان رنگ زمینه داشته باشد و یا رنگ سطر/ستون متفاوت باشد)

برای **فرمول‌ها و روابط** نیز موارد زیر را رعایت کنید:

* شماره فرمول‌ها و روابط مانند شماره شکل و جدول دارای شماره فصل نیز باشد (فرمول اول در فصل دوم می‌شود 2-1).
* اندازه فونت شماره روابط هم دو واحد از متن اصلی کوچک‌تر باشد (در اینجا 12) و فقط شماره باشد (بدون کلماتی مانند فرمول و رابطه و معادله و ...).
* توصیه می‌شود برای نظم دادن به محل شماره‌ها، از جدول با حاشیه صفر استفاده کنید (مانند نمونه داده شده در قالب) و برای افزودن فرمول جدید، همین جدول را کپی کنید و فرمول جدید خود را به جای این فرمول بنویسید و بعد از انتخاب متن شماره، دکمه F9 را برای به‌روز کردن شماره بزنید.
* در همه روابط، لازم است همه متغیرها به صورت کامل در متن تعریف شوند که چه معنی‌ای دارند.
* برای نوشتن فرمول‌ها از ابزارهای استاندارد مانند Equation خود Word استفاده کنید و علاوه بر فرمول‌های اصلی، متغیرهای درون متن را نیز با همین ابزار بنویسید.

## پیوست پ- مراجع

نکات مرتبط با مرجع‌نویسی و ارجاع دادن در متن به صورت خلاصه شامل موارد زیر است:

* لیست منابع و مراجع مورد استفاده بایستی با شکلی همسان و با استفاده از فرمت ارجاع‌دهی IEEE در متن و در انتهای گزارش بیایند. باید تناظر یک به یک بین فهرست منابع در انتهای گزارش و منابع مورد استفاده در متن باشد. سبک ارجاع‌دهی IEEE از لینک زیر قابل دسترسی است.

<https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf>

* به طور خلاصه، در زمان نوشتن مراجع، برای هر مرجع باید سه بخش زیر نوشته شود:
  + مشخصات نویسندگان: شامل نام و نام خانوادگي مؤلف‌ها
  + مشخصات منبع: مثلا برای کتاب شامل عنوان کتاب، شماره جلد، شماره ويراست، نام و نام خانوادگي مترجم (در صورت وجود)
  + مشخصات نشر: مثلا برای کتاب شامل نام ناشر، محل نشر، تاريخ چاپ
* یکی از ایرادهای رایج در نوشتن مرجع‌ها این است که دانشجویان آن را سایت‌هایی مانند Google Scholar برمی‌دارند و به دقیق بودن آن اعتماد می‌کنند، در حالی‌که ممکن است کامل نباشد. لذا لازم است همه مراجع به صورت دستی بررسی شوند و از کامل بودن همه اطلاعات آن اطمینان حاصل شود.
* برای ارجاع‌دهی، از شماره مرجع در متن استفاده کنید.
* ترتيب ارائه مرجع‌ها، ترتیب ارائه آن در متن است.
* در متن می‌توانید به هر مرجع، بر حسب ضرورت، بیش از یک بار ارجاع دهید.
* برای سهولت مدیریت مراجع، توصیه می‌شود از ابزارهای مدیریت مراجع مانند EndNote استفاده کنید.

## پیوست ت- ایرادهای رایج در تدوین رساله/پایان‌نامه

در چند سال تجربه راهنمایی، مشاوره و داوری رساله‌ها و پایان‌نامه‌های دانشگاه‌های مختلف کشور، ایرادهای رایجی که در نوشتن رساله‌/پایان‌نامه بوده‌اند، عبارتنداز:

* **چکیده** به جای اینکه خلاصه کار انجام شده باشد، بیشتر مقدمه است. بخش مقدماتی چکیده باید حدود 3-4 سطر باشد و مابقی آن بیانگر شرح کار خودتان باشد (روش و نوآوری و نتایج).
* **یکدست نبودن نوع و اندازه فونت** در سراسر متن (اندازه فونت کلمات انگلیسی دو واحد کوچک‌تر از فونت فارسی باشد). برای یکدست بودن مطالب در سراسر فایل، در صورت استفاده از Word می‌توانید از قابلیت Style در Formatting استفاده کنید.
* **فاصله** عنوان بخش‌ها و شکل ‌ها و جدول‌ها از متن باید به اندازه کافی (یک سطر) باشد.
* **نیم‌فاصله**: بسیاری از کلمات چندبخشی جداگانه یا متصل نوشته می‌شوند. بهتر است نیم‌فاصله در زمان تایپ اولیه آن کلمه گذاشته و نه در یک دور دیگر بررسی شود.
* **پانوشت**: فقط در اولین رخداد کلمه مورد نظر آورده شود و دفعات دیگر دوباره تکرار نشود.
* **پانوشت**: در هر صفحه شماره آن از یک شروع شود (پیوسته نباشد).
* **اختصارها**: فقط در اولین رخداد کلمه مورد نظر به صورت کامل آورده شود و پانوشت هم بشود و دفعات دیگر دوباره تکرار نشود.
* **معادل‌های فارسی برای کلمات تخصصی**: نیاز است از مراجعی مانند وب‌سایت فرهنگستان زبان و ادب فارسی (<https://apll.ir>) و یا پایان‌نامه‌های قبلی و همچنین با استفاده از تجربه و دانش استادان مرتبط با موضوع، برای کلمات تخصصی معادل فارسی درست انتخاب و استفاده شود.
* **فارسی‌نویسی**: نیاز است کلمات تخصصی حتما فارسی شوند و از معادل فارسی آنها یا اختصار استفاده شود. همچنین اشتباه‌های رایج نوشتاری بیان شده در پیوست ث (در کلماتی مانند پیشنهادات، جدوال و ...) باید وجود نداشته باشد.
* **مرجع‌نویسی**: یکی از ایرادهای رایج در نوشتن مرجع‌ها این است که دانشجویان آن را سایت‌هایی مانند Google Scholar برمی‌دارند و به دقیق بودن آن اعتماد می‌کنند، در حالی‌که در موارد متعددی کامل نیستند و لازم است همه مراجع به صورت دستی بررسی شوند و از کامل بودن همه اطلاعات آن اطمینان حاصل شود.
* **ارجاع‌دهی**: عدم ارجاع به مراجع مرتبط در عنوان شکل و جدول برای مواردی که شکل یا جدول از یک مرجع به صورت مستقم گرفته شده است.
* **ارجاع‌دهی:** عدم ارجاع به موقع به مرجع: در مواردی یکی یا دو پاراگراف متن ارائه می‌شود و مرجع در انتهای آن ارائه می‌شود، در حالیکه لازم است در همان ابتدا یا در متن جمله اول و یا در انتهای جمله اول مرجع ارائه شود.
* **ارجاع‌دهی:** عدم ارجاع کافی در مواردی که آمارها و اطلاعات و یا ادعاهایی مطرح می‌شود که نیاز به راستی‌آزمایی دارد.
* **ارائه جزئیات کافی از روش انجام شده**: گاهی دانشجویان برخی مطالب و کارهای انجام شده توسط آنها آنقدر برای خودشان بدیهی است که ممکن است حواسشان به اهمیت و ضرورت ارائه آن نباشد. یادتان باشد شما باید آنقدر جزئیات روش و نتایج خود را تشریح کنید که خواننده بتواند بعد از مطالعه کار شما، قادر باشد کار شما را دوباره تکرار کند!
* **پیشنهادهای ادامه کار:** پیشنهاد برای کارهای آتی باید در ادامه پژوهش جاری شما باشد و پیشنهادهایی باشد که شما اگر یک یا چند سال دیگر فرضت داشتید، آنها را انجام می‌دادید. پیشنهادهایی که قبل از انجام کار شما و بدون انجام این پژوهش هم می‌توانست ارائه شود، هرچند می‌تواند در لیست پیشنهادهای شما باشد اما پیشنهادهای جذابی نخواهد بود!
* عدم استفاده از متغیرهای یکسان برای یک مفهوم در سراسر متن: هر مفهوم فقط با یک متغیر اسم‌گذاری شود.
* یکدست نبودن معادل‌های کلمات تخصصی در سراسر متن: برای یک کلمه تخصصی بیش از یک معادل استفاده نشود.
* استفاده از دو نقطه (:) در عنوان بخش‌ها: نباید از : در انتهای عنوان بخش‌ها استفاده کرد.

## پیوست ث- فارسی بنویسیم، غلط ننویسیم

زبان فارسی از مهم‌ترین مولفه‌های فرهنگی ما ایرانیان است و نوشتن درست و دقیق به این زبان اگر توسط یک دانش‌آموخته کارشناسی ارشد یا دکتری انجام نشود، انتظار داریم چه کسی از پس این کار بربیاید! فارسی‌نویسی به ویژه برای مطالب تخصصی یک ضرورت است چراکه عدم انجام درست این کار، منجر به آسیب بنیادی به زبان می‌شود به گونه‌ای که بعد از چند نسل، زبان از وجود کلمات تخصصی به مرور تهی شده و زبان فارسی قدرت خود را در بیان مطالب علمی از دست می‌دهد.

در تدوین رساله/پایان‌نامه فارسی‌نویسی در دو محور مهم و ضروری است:

**فارسی کردن واژه‌های تخصصی**

* انتخاب معادل درست فارسی برای واژه‌های تخصصی در نوشتار رساله/پایان‌نامه ضروری است. این موضوع با توجه به پیشرفت روزمره علم و تولد واژه‌های نو، کاری آسان نیست و نمی‌توان انتظار داشت که نهادهایی مانند فرهنگستان زبان و ادب فارسی نیز با سرعت قادر به پاسخگویی به این نیاز باشند، و از این رو بسیاری از واژه‌های رایج امروزی (مانند نرم‌افزار!) ابتدا توسط استادان مرتبط استفاده شد و سپس در فرهنگستان مصوب شد. لذا برای این کار می‌توان از تجربه و دانش استادان مرتبط با موضوع کمک گرفت و برای مواردی که معادل آماده ندارند، معادل مناسب پیشنهاد و استفاده شود
* برای واژه‌هایی که مقداری قدیمی‌تر هستند، می‌توان از وب‌سایت فرهنگستان (<https://apll.ir>) و یا پایان‌نامه‌ها و کتاب‌های قبلی برای یافتن معادل فارسی کلمات تخصصی بهره گرفت.

**درست نوشتن غلط‌های رایج در فارسی**

* در زبان فارسی برخی واژه‌ها یا قواعد زبانی رایج وجود دارند که دارای ایراد زبانی هستند ولی آنقدر رایج هستند که استفاده از آن بدون اشکال به نظر می‌رسد. در اینجا چند مورد این غلط‌های رایج با الهام از کتاب "غلط ننویسیم دکتر ابولحسن نجفی" آورده شده است و انتظار می‌رود، شما هم در اصلاح این موارد همراه باشید و علاوه بر خودتان، به دیگران هم آنها را توصیه کنید.
* **جمع بستن واژه فارسی با ”ات“ عربی:** این کار درست نیست و لازم است با علامت جمع فارسی (”ها“ برای غیرجاندار و "ان" برای جاندار) جمع بسته شود مانند پیشنهادات (درست: پیشنهادها)، آزمایشات، گزارشات، باغات، فرمایشات، گمرکات، دستورات، سفارشات.
* **جمع بستن واژه فارسی با ”جات“:** این کار درست نیست و مشابه جمع با ”ات“ است و بهتر است از آن پرهیز شود و به جای آن با "ها" جمع بسته شود، مثال: کارخانجات، اداره‌جات، نوشتجات، سبزیجات، ...
* **جمع بستن واژه فارسی با ”ین“ و ”ـون“:** این قاعده زبان عربی برای جمع بستن جانداران صاحب شعور است، در فارسی بهتر است با ”ان“ (یا ”ها“) جمع ببندیم، مانند معلمین (درست: معلمان)، مسافرین، مومنین، مهندسین، روحانیون، انقلابیون، ...
* **جمع بستن واژه فارسی با جمع مکسر:** واژه‌های فارسی با جمع مکسر جمع بسته نمی‌شوند، مانند اساتید (درست: استادان)، بنادر، دراویش، فرامین، میادین، ارامنه، اکراد، ...
* **جمع در جمع:** بهتر است استفاده نشود، مانند احوالات و احوال‌ها (احوال جمع است)، اخبارها، حروف‌ها، اسباب‌ها، حبوبات، کسورات، امورات، شئونات، جواهرات.
* **استفاده از تنوین با واژه فارسی:** اشتباه است، گاهاً (درست: گاهی)، تلگرافاً (واژه فرانسوی)، جاناً، دوماً، سوماً، زباناً، ناچاراً.
* **گذاشتن و گزاردن:** گزاردن به معنی به جا آوردن، ادا کردن است مانند سپاس‌گزار، شکرگزار، خدمت‎گزار، خراج‌گزار، نمازگزار، حج‌گزار، همچنین گزاردن به معنای برگرداندن از زبانی به زبان دیگر، تعبیر کردن، شرح دادن نیز هست مانند خبرگزار.

گذاشتن به معنای قرار دادن به طور عینی و مشهود (مانند کتاب را روی میز می‌گذارم) یا قرارداد کردن، وضع کردن، تاسیس کردن است مانند قانون‌گذار (کسی که قانون را وضع می‌کند و نه کسی که اجرا می‌کند)، سرمایه‌گذاری، نام‌گذاری.

بنابراین سپاس‌گزار درست است و سپاس‌گذار نادرست. امانت‌گذار (کسی که امانتی را نزد دیگری می‌گذارد) و امانت‌گزار (کسی که امانت را ادا می‌کند).

* **با ”ت“ بنویسیم نه با ”ط“:** مانند اتاق، باتری، تپانچه، بلیت (فرانسوی)
* **را مفعولی:** صحیح نیست پس از فعل ”را“ مفعولی بیاید
  + دلیرمردان خرمشهر که مدت زیادی دست نیروی‌های بعث بود را پس گرفتند (نادرست)
  + دلیرمردان خرمشهر را که مدت زیادی دست نیروی‌های بعث بود پس گرفتند (درست)

و بهتر است ”را“ به مفعول نزدیک باشد

* + سودان استرداد احمد از مصر را تقاضا کرد (نادرست)
  + سودان استرداد احمد را از مصر تقاضا کرد (درست)
* **برخی موارد دیگر**

|  |  |  |
| --- | --- | --- |
| **توضیح** | **درست** | **نادرست** |
|  | باسمه تعالی | بسمه تعالی |
| دکترا فرانسوی است، دکتری=”ی“ اسم‎‌ساز فارسی | دکتری | دکترا |
|  | سؤال | سئوال |
| برات و حواله تجاری | سُفته | سَفته |
|  | دعوا | دعوی |
| درنگ واژه فارسی است | بی‌درنگ | بلادرنگ |
| ”یت“ مصدری مختص عربی است و رهبر واژه فارسی | رهبری | رهبریت |
| ترجمه لفظ به لفظ از point of view/ point de vue | دیدگاه | نقطه‌نظر |



University of Tehran

College of Interdisciplinary Sciences and Technologies

Department of ??

Title

By:

Your Name

Supervisor(s):

Dr. ??

Advisor(s):

Dr. ??

A Thesis Submitted to the Graduate Office in Fulfillment

of Requirements for the Degree of Master of Science

in Network Science and Technology

March 2023