

**باسمه تعالی**



**دانشگاه صنعتی مالک‌اشتر**

مجتمع دانشگاهی برق و کامپیوتر

**پايان‌نامه دوره کارشناسي ارشد رشته مهندسي کامپیوتر**

عنوان

بهبود تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای مبتني بر يادگيري ماشين

توسط: اميرحسين قاسمي

استاد راهنما: آقاي دكتر علي هادوي

استاد (اساتید) مشاور

1 خرداد ماه 1403

**باسمه تعالی**



**دانشگاه صنعتی مالک‌اشتر**

مجتمع دانشگاهی برق و کامپیوتر

**پایان­نامه­ دوره کارشناسی ارشد رشته­ی مهندسی کامپیوتر گرایش رايانش امن**

**با عنوان**

بهبود تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای مبتني بر يادگيري ماشين

**در تاریخ -/-/- توسط کمیته­ی تخصصی زیر، موردبررسی قرار گرفت و با نمره­ی ................... و درجه­ی ................... به تصویب رسید.**

|  |  |  |  |
| --- | --- | --- | --- |
| ردیف | مسئولیت | عنوان | امضای |
| **1** | **استاد راهنمای اول** | **دکتر** **علی هادوی** | ……………. |
| **2** | **استاد راهنمای دوم** | **دکتر** ……………. | ……………. |
| **3** | **استاد مشاور** | **دکتر محمدعلي كيوان‏راد** | ……………. |
| **4** | **استاد داور بیرونی** | **دکتر** ……………. | ……………. |
| **5** | **استاد داور بیرونی** | **دکتر** ……………. | ……………. |
| **6** | **استاد داور داخلی** | **دکتر** ……………. | ……………. |
| **7** | **استاد داور داخلی** | **دکتر** ……………. | ……………. |
| **8** | **مدیر تحصیلات تکمیلی مجتمع** | **دکتر** ……………. | ……………. |

تعیین سطح طبقه‌بندی پایان‌نامه

عنوان پایان‌نامه: بهبود تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای مبتني بر يادگيري ماشين

نام و نام خانوادگی استاد راهنما: **دکتر علي هادوي**

|  |  |  |
| --- | --- | --- |
| نام و نام خانوادگی دانشجو: اميرحسين قاسمي | | شماره دانشجویی: 4011417038 |
| رشته: **مهندسی كامپيوتر** | | گرایش: رايانش امن |
| ازنظر استاد راهنما این پایان‌نامه: | 1- طبقه‌بندی دارد□ | 2- طبقه‌بندی ندارد□ |
|  | | نام و امضای استاد راهنما: |

"تبصره: باتوجه‌به طبقه دار بودن پایان‌نامه، تا تاریخ ................ این پایان‌نامه، فقط با نظر این واحد دانشگاهی، در اختیار متقاضی قرار گیرد."

**تمام حقوق مادی و معنوی مترتب بر نتایج مطالعات،**

**ابتکارات و نوآوری­های حاصل از این تحقیق، متعلق**

**به دانشگاه صنعتی مالک اشتر است.**

|  |  |  |
| --- | --- | --- |
| ریاست گروه علمی | مدیر آموزش و تحصیلات تکمیلی مجتمع | ریاست مجتمع |
|  | **دکتر کوروش داداش تبار احمدی** | **دکتر علی جبار رشیدی** |

**تأیید صحت و اصالت نتایج پایان‌نامه**

اینجانب اميرحسين قاسمي به شماره دانشجویی 4011417038 دانشجوی رشته مهندسی كامپيوتر گرایش رايانش امن مقطع تحصیلی کارشناسی‌ارشد تأیید می‌نمایم که کلیه نتایج این پایان‌نامه حاصل کار اینجانب و بدون دخل‌وتصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع آورده‌ام.

در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با این‌جانب رفتار خواهد شد و حق هرگونه اعتراض در خصوص احقاق حقوق مکتسب و تعیین تخلف و مجازات را از خویش صلب می‌نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و مراجع ذی‌صلاح (اعم از اداری و قضایی) به عهده این‌جانب خواهد بود و دانشگاه هیچ‌گونه مسئولیتی دراین‌خصوص نخواهد داشت.

فهرست مطالب

عنوان صفحه

[فهرست مطالب ‌ه](#_Toc221920059)

[فهرست جدول‌ها ‌و](#_Toc221920060)

[فهرست شکل‌ها ‌ز](#_Toc221920061)

[فهرست نمادها ‌ط](#_Toc221920062)

[کوته‌نوشت‌ها ‌ي](#_Toc221920063)

[چکيده ‌ك](#_Toc221920064)

[فصل اول -مقدمه 1](#_Toc221920065)

[1-1 بيان مسئله 2](#_Toc221920066)

[1-1-1 مسئله و اهداف اصلي تحقيق: 2](#_Toc221920067)

[1-2 تشريح و بيان موضوع: 3](#_Toc221920068)

[1-2-1 تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای 4](#_Toc221920069)

[1-2-2 بررسي علل نياز به مجموعه‌داده‏هاي قابل‌اعتماد و قوي 7](#_Toc221920070)

[1-2-3 بررسي اهميت پايگاه‏داده‏هاي غيررابطه‏اي 8](#_Toc221920071)

[1-2-4 ارائه نمونه‌ای حمله تزريق پایگاه‌داده غیررابطه‌ای 8](#_Toc221920072)

[1-2-5 بررسي و تشريح مدل 11](#_Toc221920073)

[1-3 ضرورت انجام تحقيق 13](#_Toc221920074)

[1-4 سابقه تحقيقات و مطالعات انجام‌گرفته 14](#_Toc221920075)

[1-5 هدف پژوهش 18](#_Toc221920076)

[1-6 نوآوري پژوهش 22](#_Toc221920077)

[1-7 ساختار فصول پایان‌نامه 24](#_Toc221920078)

[فصل دوم- مبانی نظری و پیشینه پژوهش 26](#_Toc221920079)

[2-1 مبانی نظری 27](#_Toc221920080)

[2-1-1 شناخت پایگاه‌داده‌های غیررابطه‌ای 27](#_Toc221920081)

[2-1-2 بررسی نقاط ضعف و چالش‌های امنیتی مرتبط با حملات تزریق داده در محیط‌های پایگاه‌داده غیررابطه‌ای 28](#_Toc221920082)

[2-1-3 یادگیری ماشین 33](#_Toc221920083)

[2-1-3-2 معرفی مدل‌های زبان بزرگ و مقایسه آن‌ها با روش‌های سنتی درتولید داده‌های مصنوعی 37](#_Toc221920084)

[2-4 پیشینه پژوهش 40](#_Toc221920085)

[2-4-1 تزریق پایگاه‌داده غیررابطه‌ای 40](#_Toc221920086)

[2-4-2 رویکرد یادگیری ماشین 53](#_Toc221920087)

[2-4-3 رویکرد غیر یادگیری ماشین 55](#_Toc221920088)

[فصل سوم روش پیشنهادی 58](#_Toc221920089)

[3-1 معرفی و توصیف رویکرد کلی پژوهش 59](#_Toc221920090)

[3-1-1 خط‌لوله کلی پژوهش 61](#_Toc221920091)

[3-1-2سناریوهای داده (سه شاخه اجرایی) 62](#_Toc221920092)

[3-1-3 مدل‌های یادگیری ماشین و منطق انتخاب 63](#_Toc221920093)

[3-1-4 راهبرد ارزیابی و گزارش‌دهی 63](#_Toc221920094)

[3-2روش جمع‌آوری داده 64](#_Toc221920095)

[**3-3-1 استخراج ویژگی‌های کلیدی از مجموعه‌داده اولیه** 64](#_Toc221920096)

[3-2-1 تولید داده‌های مصنوعی با استفاده از مدل ChatGPT (GPT-5.2) 65](#_Toc221920097)

[4-3 معرفی مدل‌های یادگیری ماشین و مدل‌های تجمیعی به‌کاررفته 67](#_Toc221920098)

[**3-5 ارائه جزئیات مربوط به تنظیم پارامترها** 70](#_Toc221920099)

[3-6 پیاده‌سازی 75](#_Toc221920100)

[فصل چهارم- تجزیه‌وتحلیل داده‌ها (یافته‌ها) 81](#_Toc221920101)

[4-1 معیارهای ارزیابی عملکرد 82](#_Toc221920102)

[4-2 بررسی دیتاست‌ها 86](#_Toc221920103)

[**4-3 مقایسه عملکرد مدل‌ها** 94](#_Toc221920104)

[**4-3-1 عملکرد مدل‌های مختلف یادگیری ماشینی روی دیتاست اصلی** 94](#_Toc221920105)

[4-3-2 عملکرد روی Syn-2000 (Seed = 50% Real) 99](#_Toc221920106)

[4-3-3 عملکرد روی Syn-2000 (Seed = All Real-400) 103](#_Toc221920107)

[4-3-4 مقایسه نهایی سه سناریو (Real-400 vs Syn-2000(50%) vs Syn-2000(100%)) 106](#_Toc221920108)

[فصل 5: بحث، نتیجه گیري و پیشنهادات 109](#_Toc221920109)

[**5-1 مرور اجمالی نتایج تحقیق** 110](#_Toc221920110)

[5-2 تحلیل نتایج در چارچوب پیشینه پژوهش 112](#_Toc221920111)

[5-2-1 جایگاه پژوهش حاضر نسبت به رویکردهای رایج (ابزارمحور/تحلیل‌کد/یادگیری ماشین) 112](#_Toc221920112)

[5-2-2 چرایی تفاوت نتایج نسبت به برخی مطالعات قبلی 112](#_Toc221920113)

[5-2-3 نقاط قوت و محدودیت‌ها 113](#_Toc221920114)

[5-3 بررسی کاربردهای عملی نتایج در دنیای واقعی (بازنویسی‌شده و همسو با نتایج شما) 114](#_Toc221920115)

[5-3-1 کاربردهای مستقیم در سامانه‌های امنیتی 114](#_Toc221920116)

[5-3-2 کاربردهای غیرمستقیم: تولید داده و تمرین/آموزش امنیتی 115](#_Toc221920117)

[5-3-3 پیامدها برای صنعت، جامعه و مسیر تحقیقات آینده 115](#_Toc221920118)

[5-4 محدودیت‌های پژوهش 116](#_Toc221920119)

[5-5 محدودیت‌های ذاتی روش تحقیق یا داده‌های استفاده‌شده 117](#_Toc221920120)

[5-6 پیشنهادها برای تحقیقات آینده 117](#_Toc221920121)

[5-7 جمع‌بندی نهایی 118](#_Toc221920122)

[فهرست مراجع 119](#_Toc221920123)

فهرست جدول‌ها

عنوان صفحه

[جدول 2- 1: انواع دیتا بیس ها 31](#_Toc221920150)

[جدول 2- 2: مقایسه مقالات [42], [43], [44] 52](#_Toc221920151)

[جدول 2- 3: مقایسه مقالات [42], [43], [44] از دیدگاه محتلف 52](#_Toc221920152)

[جدول 2- 4: ویژگی‌های انتخاب‌شده 53](#_Toc221920153)

[جدول 3- 1: پارامترهای رگرسیون خطی 71](#_Toc221920163)

[جدول 3- 2: پارامترهای درخت تصمیم 71](#_Toc221920164)

[جدول 3- 3: پارامترهای Gradient Boosting 71](#_Toc221920165)

[جدول 3- 4: پارامترهای نزدیکترین همسایه 71](#_Toc221920166)

[جدول 3- 5: پارامترهای درخت تصادفی 72](#_Toc221920167)

[جدول 3- 6: پارامترهای XGBoost 72](#_Toc221920168)

[جدول 3- 7: پارامترهای LightGBM 72](#_Toc221920169)

[جدول 3- 8:پارامترهای Extra Trees 73](#_Toc221920170)

[جدول 3- 9: پارامترهای Support Vector Machine 73](#_Toc221920171)

[جدول 3- 10: پارامترهای AdaBoost 73](#_Toc221920172)

[جدول 3- 11: پارامترهای Voting Ensemble 74](#_Toc221920173)

[جدول 3- 12: پارامترهای Estimators داخل Voting Ensemble 74](#_Toc221920174)

[جدول 3- 13:پارامترهای Stacking Ensemble 74](#_Toc221920175)

[جدول 3- 14: پارامترهای Estimators داخل Stacking Ensemble 74](#_Toc221920176)

[جدول 4- 1: معیارهای ارزیابی 85](#_Toc221920273)

[جدول 4- 2: تحلیل مقایسه‌ای دیتاست ها 87](#_Toc221920274)

[جدول 4- 3: خلاصه نتایج اعتبارسنجی متقاطع 10-فولدی روی دیتاست اصلی 95](#_Toc221920275)

[جدول 4- 4: نتایج آزمون نهایی مدل‌ها روی دیتاست اصلی 96](#_Toc221920276)

[جدول 4- 5: چهار مؤلفه (TN, FP, FN, TP) برای هر مدل 97](#_Toc221920277)

[جدول 4- 6: خلاصه ماتریس درهم‌ریختگی مدل‌ها روی Syn-2000 (Test = 400) 99](#_Toc221920278)

[جدول 4- 7: نتایج عملکرد مدل‌ها روی Syn-2000 (Test) 101](#_Toc221920279)

[جدول 4- 8: خلاصه ماتریس‌های درهم‌ریختگی مدل‌ها روی Syn-2000 (Seed=All Real-400) در مجموعه آزمون 103](#_Toc221920280)

[جدول 4- 9: نتایج عملکرد مدل‌ها روی Syn-2000 (Seed=All Real-400) در مجموعه آزمون 104](#_Toc221920281)

[جدول 4- 10: جدول تجمیعی «بهترین مدل هر سناریو» بر اساس نتایج آزمون - معیار انتخاب بهترین مدل : Accuracy و F1(macro) به‌عنوان شاخص عملکرد کلی، و FN/FP به‌عنوان شاخص‌های ریسک امنیتی/هشدار کاذب. 106](#_Toc221920282)

فهرست شکل‌ها

عنوان صفحه

[شکل2- 1محبوبيت در بين 500شركت برتر جهان[27] 28](#_Toc221920244)

[شکل2- 2: مدل مفهومی یادگیری ماشین[46] 54](#_Toc221920245)

[شکل2- 3: نمونه کوئری در منگو دیبی 56](#_Toc221920246)

[شکل2- 4: مدل نمونه برای ورودی کاربر مورد نظر حمله[47] 56](#_Toc221920247)

[شکل2- 5: فلوچارت کد جاوا اسکریپت برای محدود کردن ورودی[29] 57](#_Toc221920248)

[شکل 3- 1: مدل مفهومی جزییات دیتا ست ها 60](#_Toc221920252)

[شکل 3- 2: مدل مفهومی جزییات پیاده سازی مدل پیشنهادی 61](#_Toc221920253)

[شکل 4- 1: ماتریس درهم ریختگی مدلها 98](#_Toc221920259)

[شکل 4- 2: کنفیوژن ماتریس مدلها با 2000 دیتا مصنوعی 100](#_Toc221920260)

فهرست نمادها

عنوان علامت اختصاري

|  |  |
| --- | --- |
| ضريب ميرايي |  |
|  |  |
|  |  |
|  |  |
|  |  |

کوته‌نوشت‌ها

|  |  |
| --- | --- |
| Additive White Gaussian Noise | AWGN |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

چکيده

تزریق در پایگاه‌داده‌های غیررابطه‌ای[[1]](#footnote-2)‎‎‌‌ یکی از آسیب‌پذیری‌های مهم در برنامه‌های وب است که معمولاً در اثر اعتبارسنجی نامناسب ورودی و ساخت پرس‌وجوهای پویا با داده کاربر رخ می‌دهد و می‌تواند به دور زدن احراز هویت، استخراج/تغییر داده و سایر پیامدهای امنیتی منجر شود. با توجه به گسترش استفاده از پایگاه‌داده‌های غیررابطه‌ای و افزایش سطح حملات متن‌محور و مبتنی بر کوئری، نیاز به راهکارهای داده‌محور برای تشخیص این حملات بیش از گذشته احساس می‌شود.

هدف این پایان‌نامه ارائه یک رویکرد داده‌محور برای تشخیص دقیق‌تر حملات تزریق در پایگاه‌داده‌های غیررابطه‌ای و هم‌زمان کاهش هشدارهای کاذب است؛ به‌گونه‌ای که محدودیت رایج در این حوزه یعنی کمبود داده‌های واقعی و برچسب‌دار نیز تا حد امکان برطرف شود. در همین راستا، علاوه بر استفاده از داده واقعی (Real-400)، دو مجموعه‌داده مصنوعی Syn-2000 با بهره‌گیری از مدل زبانی بزرگ GPT-5.2 تولید شد: (1) سناریوی Seed=50% Real و (2) سناریوی Seed=All Real-400. استفاده از LLMها در تولید داده مصنوعیِ وظیفه‌محور، در پژوهش‌های جدید به‌عنوان رویکردی مؤثر برای افزایش پوشش داده و تقویت آموزش مدل‌ها مطرح شده است.



در مرحله مدل‌سازی، چندین الگوریتم یادگیری ماشین (از جمله رگرسیون خطی، درخت تصادفی، XGBoost، LightGBM، نزدیکترین همسایه و روش‌های ترکیبی مبتنی بر Voting و Stacking) آموزش داده شدند و عملکرد آن‌ها با رویکرد ارزیابی منظم و حفظ توزیع کلاس‌ها (مانند Stratified K-Fold) سنجیده شد. نتایج آزمایش‌ها نشان داد «بهترین مدل» با تغییر سناریوی داده ثابت نمی‌ماند: در Real-400 مدل درخت تصادفی بهترین عملکرد کلی را داشت؛ در Syn-2000 (Seed=50%) مدل نزدیکترین همسایه بیشترین کارایی را نشان داد؛ و در Syn-2000 (Seed=100%) مدل LightGBM بهترین توازن را در شاخص‌های کلی و کنترل خطاها ارائه کرد. همچنین مشاهده شد که در برخی سناریوها، روش‌های ترکیبی می‌توانند نرخ خطای از دست‌رفتن حمله (FN) را کاهش دهند، هرچند ممکن است با افزایش هشدار کاذب (FP) همراه شوند؛ بنابراین انتخاب نهایی مدل باید متناسب با سیاست عملیاتی سامانه (FN-محور یا FP-محور) انجام گیرد.

به‌طور کلی، این پژوهش نشان می‌دهد تقویت داده با تولید مصنوعی مبتنی بر GPT-5.2 می‌تواند به بهبود معنادار عملکرد مدل‌های تشخیص تزریق NoSQL کمک کند و در کنار آن، تحلیل همزمان FP و FN تصویری عملیاتی از ریسک امنیتی ارائه می‌دهد. همچنین نتایج حاکی از آن است که مدل‌های تقویتی مبتنی بر درخت تصمیم مانند LightGBM—که برای کارایی و یادگیری روابط غیرخطی طراحی شده‌اند—در سناریوهای داده غنی‌تر، ظرفیت بالایی برای تشخیص دقیق‌تر دارند.

کلیدواژه‌ها: تزریق NoSQL، پایگاه‌داده‌های غیررابطه‌ای، داده مصنوعی، مدل‌های زبان بزرگ (GPT-5.2)، یادگیری ماشین، یادگیری گروهی، تشخیص حمله، کاهش هشدار کاذب، مجموعه‌داده [[2]](#footnote-3)، یادگیری ماشینی[[3]](#footnote-4)

# فصل اول -مقدمه

## 1-1 بيان مسئله



### 1-1-1 مسئله و اهداف اصلي تحقيق:



با گسترش روزافزون استفاده از سیستم‌های مبتنی بر وب و پایگاه‌های داده، امنیت اطلاعات و جلوگیری از نفوذ و حملات به این سیستم‌ها از اهمیت ویژه‌ای برخوردار شده است. حملات تزریق داده، ازجمله حملات تزریق پایگاه‌داده رابطه‌ای، به‌عنوان یکی از متداول‌ترین روش‌های سوءاستفاده از آسیب‌پذیری‌های پایگاه‌های داده مطرح هستند. برای مقابله با این نوع حملات، پژوهشگران به تکنیک‌های یادگیری ماشینی روی آورده‌اند. استفاده از این تکنیک‌ها برای شناسایی و جلوگیری از حملات تزریق پایگاه‌داده رابطه‌ای در پژوهش‌های متعددی گزارش شده است و نتایج امیدوارکننده‌ای نیز ارائه داده‌اند.

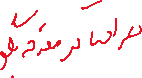
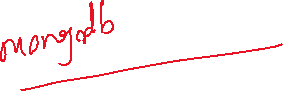
بااین‌حال، ظهور پایگاه‌های داده پایگاه‌داده غیررابطه‌ای[[4]](#footnote-5) به دلیل قابلیت مقیاس‌پذیری بالا، انعطاف‌پذیری در مدل داده و توانایی مدیریت حجم عظیمی از داده‌ها، باعث گسترش کاربردهای آن در زمینه‌های مختلف شده است[2]. با افزایش استفاده از این نوع پایگاه‌ها، حملات تزریق پایگاه‌داده غیررابطه‌ای به‌عنوان یک چالش امنیتی نوظهور شناخته شده‌اند. برخلاف حملات تزریق پایگاه‌داده رابطه‌ای که بر ساختار جدولی و زبان پرس‌وجوی استاندارد ( پایگاه‌داده رابطه‌ای) متکی هستند، حملات تزریق پایگاه‌داده غیررابطه‌ای از ضعف‌های موجود در ساختارهای غیر جدولی و زبان‌های پرس‌وجوی خاص این نوع پایگاه‌ها سوءاستفاده می‌کنند[3].



علی‌رغم اهمیت فزاینده موضوع، پژوهش‌های کمتری به شناسایی و مقابله با حملات تزریق پایگاه‌داده غیررابطه‌ای پرداخته‌اند. به همین دلیل، توسعه رویکردهای مبتنی بر یادگیری ماشینی برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای به یک نیاز ضروری تبدیل شده است[4]. این تحقیق بر آن است تا با ارائه روش‌هایی نوین، به مقابله با این چالش بپردازد.

یکی از چالش‌های اصلی در اجرای این پژوهش، فقدان مجموعه‌دادگان مناسب و کافی برای آموزش و ارزیابی مدل‌های یادگیری ماشینی است. بسیاری از پژوهش‌های گذشته در زمینه حملات تزریق پایگاه‌داده رابطه‌ای، از مجموعه‌دادگان عمومی و موجود بهره برده‌اند، اما در زمینه حملات پایگاه‌داده غیررابطه‌ای چنین مجموعه‌دادگان استانداردی وجود ندارد. این امر فرایند ارزیابی دقیق و قابل‌اعتماد مدل‌ها را دشوار می‌سازد.

برای غلبه بر این چالش، یکی از روش‌های پیشرفته تولید داده‌های مصنوعی و متنوع، استفاده از مدل‌های زبان بزرگ است. این مدل‌ها، باتکیه‌بر معماری‌های عمیق و مبتنی بر ترانسفورمر، قادر به تولید داده‌هایی با شباهت بالا به داده‌های واقعی هستند. استفاده از مدل‌های زبان بزرگ امکان تولید داده‌های متنی و ساختاریافته را باکیفیت بالا فراهم می‌کند، به‌گونه‌ای که داده‌های تولیدشده قابلیت استفاده به‌عنوان مجموعه‌دادگان آموزشی برای مدل‌های یادگیری ماشینی را دارند. این رویکرد، علاوه بر دقت بالا، انعطاف‌پذیری بیشتری در تطبیق با انواع سناریوها و کاربردها ارائه می‌دهد و می‌تواند جایگزینی مؤثر برای روش‌های سنتی مانند شبکه‌های مولد متخاصم باشد.



## 1-2 تشريح و بيان موضوع:

این پژوهش شامل طراحی و ارزیابی یک سیستم یادگیری گروهی است که از مدل‌های مختلف مانند رگرسیون لجستیک، جنگل تصادفی و XGBoost تشکیل شده است. این مدل‌ها با ترکیب قدرت یادگیری فردی خود، دقت شناسایی حملات را بهبود می‌بخشند. دستاوردهای مورد انتظار این پژوهش شامل ارائه روشی نوین برای شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای، ایجاد مجموعه‌دادگان جامع و متنوع، و بهبود دقت مدل‌های یادگیری ماشینی در محیط‌های واقعی است.

مفاهیم کلیدی این پژوهش شامل شناسایی و مقابله با حملات تزریق پایگاه‌داده غیررابطه‌ای است که از نقاط ضعف این نوع پایگاه‌ها سوءاستفاده می‌کنند. این حملات معمولاً از طریق ارسال درخواست‌های مخرب و دست‌کاری پرس‌وجوها انجام می‌گیرند. همچنین، تحلیل و طراحی مجموعه‌دادگان شامل داده‌های اولیه و داده‌های مصنوعی تولیدشده با مدل‌های زبان بزرگ، یکی از محورهای اصلی پژوهش است. در این راستا، به‌جای استفاده از شبکه‌های مولد متخاصم، از مدل‌های زبان بهره گرفته می‌شود که توانایی تولید داده‌های متنی و ساختاریافته باکیفیت بالا را دارند.

این پژوهش با هدف ارائه راهکاری جامع برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای طراحی شده است. برای دست‌یابی به این هدف، مجموعه‌داده‌های اولیه به‌صورت دستی تحلیل و برچسب‌گذاری می‌شوند تا کیفیت داده‌های پایه تضمین گردد. سپس از مدل‌های زبان بزرگ برای تولید داده‌های مصنوعی متنوع و واقعی‌تر استفاده می‌شود که به غنی‌سازی مجموعه‌دادگان کمک می‌کند. در مرحله بعد، تکنیک‌های مختلف یادگیری ماشینی برای ارزیابی و دسته‌بندی داده‌های موجود و مصنوعی به‌کار گرفته می‌شوند.

مفاهیم کلیدی این پژوهش شامل شناسایی و مقابله با حملات تزریق پایگاه‌داده غیررابطه‌ای است که از نقاط ضعف این نوع پایگاه‌ها سوءاستفاده می‌کنند. این حملات معمولاً از طریق ارسال درخواست‌های مخرب و دست‌کاری پرس‌وجوها انجام می‌گیرند. همچنین، تحلیل و طراحی مجموعه‌دادگان شامل داده‌های اولیه و داده‌های مصنوعی تولیدشده با مدل‌های زبان بزرگ، یکی از محورهای اصلی پژوهش است. در این راستا، به‌جای استفاده از شبکه‌های مولد متخاصم، از مدل‌های زبان بهره گرفته می‌شود که توانایی تولید داده‌های متنی و ساختاریافته باکیفیت بالا را دارند.

پیش‌فرض‌ها

به‌منظور گسترش مجموعه‌دادگان و تأمین داده‌های متنوع و واقعی‌تر، از مدل‌های زبان بزرگ (مدل‌های زبان بزرگ) استفاده خواهد شد. این مدل‌ها قادرند داده‌هایی باکیفیت بالا و شباهت زیاد به داده‌های واقعی تولید کنند که می‌تواند به غنی‌سازی مجموعه‌دادگان کمک کند.

برای ارزیابی و دسته‌بندی داده‌های موجود و مصنوعی، از تکنیک‌های مختلف یادگیری ماشینی استفاده خواهد شد. این تکنیک‌ها شامل روش‌های نظارت‌شدن و احتمالی ترکیبی برای شناسایی داده‌های مخرب هستند.

بر اساس رویکرد معرفی‌شده، مفاهیم و موضوعات کلیدی زیر در این پژوهش بررسی می‌شوند:

شناسایی و مقابله با حملات تزریق پایگاه‌داده غیررابطه‌ای، که از نقاط ضعف پایگاه‌داده‌های غیررابطه‌ای استفاده می‌کنند، به‌عنوان هدف اصلی تحقیق بررسی می‌شود. این حملات معمولاً از طریق ارسال درخواست‌های مخرب و دست‌کاری پرس‌وجوها به‌منظور سوءاستفاده از اطلاعات پایگاه‌داده انجام می‌گیرد.

تحلیل و طراحی مجموعه‌دادگان، شامل داده‌های اولیه و داده‌های مصنوعی تولیدشده با مدل‌های زبان بزرگ، یکی از محورهای پژوهش است. این مرحله شامل بررسی کیفیت داده‌ها، شناسایی ویژگی‌های مهم، و طراحی مجموعه‌دادگان قابل‌اعتماد برای مدل‌های یادگیری ماشینی است.

به‌منظور افزایش تنوع و حجم مجموعه‌دادگان، از مدل‌های تولید داده‌های مصنوعی استفاده خواهد شد. در این پژوهش به‌جای استفاده از شبکه‌های مولد متخاصم، از مدل‌های زبان بهره گرفته می‌شود که توانایی تولید داده‌های متنی و ساختاریافته باکیفیت بالا را دارند.

پژوهش شامل طراحی و ارزیابی یک سیستم یادگیری گروهی خواهد بود که از مدل‌های مختلف مانند رگرسیون خطی، درخت تصادفی، و XGBoost تشکیل شده است. این مدل‌ها با ترکیب قدرت یادگیری فردی خود، دقت شناسایی حملات را بهبود می‌بخشند.

* ارائه روشی نوین برای شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای:
* ایجاد مجموعه‌دادگان جامع و متنوع:
* بهبود دقت مدل‌های یادگیری ماشینی:

|  |
| --- |
| <https://insecure-website.com/product/lookup?category=fizzy> |
| this.category == 'fizzy' |
| ‘”`{  $;Foo}  $Foo \xYZ |
| <https://insecure-website.com/product/lookup?category='%22%60%7b%0d%0a%3b%24Foo%7d%0d%0a%24Foo%20%5cxYZ%00> |

|  |  |
| --- | --- |
| با اسنادي مطابقت دارد که عبارت جاوا اسکريپت را برآورده می‌کند. | $where |
| با تمام مقاديري که با يک مقدار مشخص برابر نيستند مطابقت دارد. | $ne |
| با تمام مقادير مشخص شده در يک آرايه مطابقت دارد. | $in |
| اسنادي را انتخاب می‌کند که مقادير با يک عبارت منظم مشخص مطابقت دارند. | $regex |

## 1-3 ضرورت انجام تحقيق

در زمینه حملات تزریق پایگاه‌داده غیررابطه‌ای، این ضرورت دوچندان می‌شود. این حملات به دلیل ساختار غیرمتعارف پایگاه‌داده‌های NoSQL و روش‌های خاص تزریق، از چالش‌های نوظهور امنیت سایبری به شمار می‌روند. شناسایی و پیشگیری از این حملات مستلزم دسترسی به داده‌هایی است که سناریوهای حمله را به طور دقیق شبیه‌سازی کنند.

۱- نیازهای بازار کار: تقاضای روزافزون برای متخصصان توانمند در مدیریت پایگاه‌داده‌های غیررابطه‌ای و مقابله با حملات سایبری، از صنایع وب و نرم‌افزار فراتر رفته و به حوزه‌های خرده‌فروشی، بهداشت و دولتی گسترش یافته است.

۳- گسترش خدمات آنلاین: افزایش وابستگی به خدمات آنلاین مبتنی بر پایگاه‌داده‌های NoSQL، نیاز به ایمن‌سازی این سیستم‌ها را به یک ضرورت جهانی تبدیل کرده است.

این پژوهش با تمرکز بر مدل‌های پیشرفته یادگیری ماشینی و مهندسی ویژگی هدفمند، گامی مؤثر در افزایش امنیت سایبری و مقابله با تهدیدات نوظهور در پایگاه‌داده‌های غیررابطه‌ای خواهد بود.

در زمینه حملات تزریق پایگاه‌داده غیررابطه‌ای، این ضرورت دوچندان می‌شود. این حملات به دلیل ساختار غیرمتعارف پایگاه‌داده‌های NoSQL و روش‌های خاص تزریق، از چالش‌های نوظهور امنیت سایبری به شمار می‌روند. شناسایی و پیشگیری از این حملات مستلزم دسترسی به داده‌هایی است که سناریوهای حمله را به طور دقیق شبیه‌سازی کنند.

۱- نیازهای بازار کار: تقاضای روزافزون برای متخصصان توانمند در مدیریت پایگاه‌داده‌های غیررابطه‌ای و مقابله با حملات سایبری، از صنایع وب و نرم‌افزار فراتر رفته و به حوزه‌های خرده‌فروشی، بهداشت و دولتی گسترش یافته است. همچنین استفاده از این پایگاه‌داده‌ها در بازارهای مبتنی بر اینترنت اشیا و سامانه‌های اجتماعی سایبر فیزیکی، چالش‌های امنیتی جدیدی را مطرح کرده است.

۳- گسترش خدمات آنلاین: با دسترسی بیش از ۹۱ درصد جمعیت ایران و ۶۷.۹ درصد جمعیت جهان به اینترنت، افزایش وابستگی به خدمات آنلاین مبتنی بر پایگاه‌داده‌های NoSQL، نیاز به ایمن‌سازی این سیستم‌ها را به یک ضرورت جهانی تبدیل کرده است. پایگاه‌داده‌های نرم‌افزارهای وب به هدف اصلی دسترسی‌های غیرمجاز تبدیل شده‌اند.

این پژوهش با تمرکز بر مدل‌های پیشرفته یادگیری ماشینی و مهندسی ویژگی هدفمند، گامی مؤثر در افزایش امنیت سایبری و مقابله با تهدیدات نوظهور در پایگاه‌داده‌های غیررابطه‌ای خواهد بود.

هدف اصلی این پژوهش، ارائه رویکردی جامع و نوین برای مقابله با تهدیدات امنیتی مرتبط با حملات تزریق پایگاه‌داده غیررابطه‌ای است. این هدف با طراحی و توسعه یک سیستم مبتنی بر یادگیری ماشینی، باقابلیت تشخیص دقیق و کارآمد بارهای مخرب [[5]](#footnote-6)و تمایز آن‌ها از بارهای قانونی[[6]](#footnote-7)، دنبال می‌شود.

۱. ایجاد و استفاده از یک مجموعه‌داده جامع و متنوع:

یکی از چالش‌های اساسی در حوزه تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای، فقدان مجموعه‌داده‌های استاندارد و جامع است. برای رفع این نقیصه، پژوهش حاضر بر تولید و بهره‌گیری از دو نوع مجموعه‌داده تمرکز دارد:

* مجموعه‌داده اولیه: این مجموعه‌داده شامل نمونه‌های واقعی و برچسب‌گذاری‌شده از حملات تزریق MongoDB است که به‌صورت دستی جمع‌آوری و برچسب‌گذاری شده‌اند.
* مجموعه‌داده مصنوعی: با استفاده از مدل‌های زبان بزرگ (GPT)، مجموعه‌داده‌ای مصنوعی تولید شده است که امکان شبیه‌سازی دقیق سناریوهای متنوع حملات تزریق پایگاه‌داده غیررابطه‌ای را فراهم می‌کند. این داده‌های مصنوعی با حفظ ویژگی‌های ساختاری و رفتاری داده‌های واقعی، تنوع و حجم مجموعه‌داده را به طور قابل‌توجهی افزایش می‌دهند.

استفاده از مدل‌های زبان بزرگ در تولید داده‌های مصنوعی، مزایای متعددی از جمله کاهش وابستگی به داده‌های واقعی محدود، افزایش تنوع سناریوهای حمله، و امکان تولید داده‌های برچسب‌گذاری‌شده با دقت بالا را به همراه دارد.

۲. مهندسی ویژگی دستی و استخراج ویژگی‌های مؤثر:

شناسایی و استخراج ویژگی‌های کلیدی از داده‌های ورودی، یکی از گام‌های حیاتی در بهبود عملکرد مدل‌های یادگیری ماشینی است. در این پژوهش، فرایند مهندسی ویژگی به‌صورت دستی و با تمرکز بر ویژگی‌های خاص حملات تزریق پایگاه‌داده غیررابطه‌ای انجام شده است. این ویژگی‌ها شامل موارد زیر می‌باشند:

* ویژگی‌های ساختاری: شناسایی الگوهای ساختاری در پرس‌وجوهای مخرب، مانند وجود عملگرهای خاص MongoDB (نظیر $ne، $where، $regex، $in)
* ویژگی‌های رفتاری: تحلیل رفتار پرس‌وجوها از جمله طول کوئری، تعداد و نوع کاراکترهای خاص، و الگوهای غیرعادی در ساختار ورودی
* ویژگی‌های محتوایی: بررسی محتوای بارهای داده (payloads) و شناسایی الگوهای مخرب در آن‌ها

فرایند مهندسی ویژگی در چندین مرحله انجام شده است:

* ابتدا با الهام از تحقیقات پیشین در حوزه تزریق SQL (مانند کار رومانی)، ویژگی‌هایی استخراج شدند که متأسفانه برای حملات NoSQL کارایی مطلوبی نداشتند.
* سپس مهندسی ویژگی دستی و خاص برای MongoDB انجام شد که نتایج بهتری را به همراه داشت.
* در نهایت، ویژگی‌های استخراج‌شده بهینه‌سازی و تنظیم شدند تا حداکثر دقت در تشخیص حملات حاصل شود.

۳. طراحی، پیاده‌سازی و بهینه‌سازی مدل‌های یادگیری ماشینی:

در این پژوهش، طیف گسترده‌ای از الگوریتم‌های یادگیری ماشینی برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای مورد بررسی و پیاده‌سازی قرار گرفته‌اند. مدل‌های استفاده‌شده عبارتند از:

* رگرسیون خطی (رگرسیون لجستیک): به‌عنوان یک مدل پایه و ساده برای طبقه‌بندی دودویی (حمله/غیرحمله)
* درخت تصمیم (درخت تصمیم): برای شناسایی الگوهای تصمیم‌گیری در داده‌ها
* درخت تصادفی (جنگل تصادفی): یک مدل گروهی مبتنی بر ترکیب چندین درخت تصمیم
* XGBoost: الگوریتم پیشرفته Gradient Boosting که کارایی بالایی در مسائل طبقه‌بندی دارد
* Gradient Boosting (گرادیان بوستینگ): برای بهبود تدریجی عملکرد مدل از طریق یادگیری از خطاهای مراحل قبل
* نزدیکترین همسایه (KNN - کا نزدیکترین همسایه): برای طبقه‌بندی مبتنی بر شباهت
* Ensemble Model (مدل گروهی): ترکیب چندین مدل (شامل رگرسیون خطی، درخت تصادفی و XGBoost) برای بهره‌گیری از نقاط قوت هر یک و افزایش دقت کلی سیستم

برای هر یک از این مدل‌ها، فرایند بهینه‌سازی شامل موارد زیر انجام شده است:

* تنظیم هایپرپارامترها: استفاده از تکنیک‌های Grid Search و Cross-Validation برای یافتن بهترین مقادیر پارامترها
* اعتبارسنجی متقابل: استفاده از روش‌های اعتبارسنجی متقابل (K-Fold Cross-Validation) برای جلوگیری از بیش‌برازش (Overfitting) و اطمینان از قابلیت تعمیم‌پذیری مدل‌ها
* بهینه‌سازی معماری مدل گروهی: تنظیم وزن‌ها و روش ترکیب مدل‌های فردی در سیستم Ensemble

۴. ارزیابی جامع رویکرد پیشنهادی از منظر امنیت، دقت و کارایی:

رویکرد پیشنهادی از چندین منظر مختلف مورد ارزیابی و تحلیل قرار گرفته است:

الف) از منظر دقت و صحت:

* Accuracy (دقت کلی): نسبت تشخیص‌های صحیح به کل نمونه‌ها
* Precision (دقت مثبت): نسبت تشخیص‌های مثبت صحیح به کل تشخیص‌های مثبت
* Recall (بازخوانی): نسبت تشخیص‌های مثبت صحیح به کل نمونه‌های مثبت واقعی
* F1-Score: میانگین هارمونیک Precision و Recall
* Confusion Matrix (ماتریس درهم‌ریختگی): تحلیل دقیق انواع خطاها (False Positive و False Negative)

ب) از منظر امنیت:

* توانایی مدل در شناسایی دقیق حملات تزریق پایگاه‌داده غیررابطه‌ای و جلوگیری از نفوذ مخرب
* کاهش نرخ مثبت کاذب (False Positive Rate) برای جلوگیری از ایجاد هشدارهای غیرضروری
* کاهش نرخ منفی کاذب (False Negative Rate) برای اطمینان از عدم فرار حملات واقعی

ج) از منظر کارایی:

* زمان پردازش و سرعت تشخیص حملات در سناریوهای واقعی
* منابع محاسباتی موردنیاز (حافظه و پردازنده) برای اجرای مدل
* قابلیت مقیاس‌پذیری (Scalability) در محیط‌های با حجم بالای داده

۵. مقایسه تطبیقی با روش‌های موجود و تحلیل عملکرد:

یکی از اهداف کلیدی این پژوهش، مقایسه دقیق و جامع رویکرد پیشنهادی با روش‌های موجود در زمینه شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای است. این مقایسه شامل موارد زیر می‌شود:

* مقایسه مجموعه‌داده‌ها: تحلیل مقایسه‌ای دقیق بین مجموعه‌داده اصلی و مجموعه‌داده مصنوعی تولیدشده توسط GPT از نظر توزیع، تنوع، و کیفیت داده‌ها
* مقایسه عملکرد مدل‌ها: ارزیابی عملکرد مدل‌های مختلف یادگیری ماشینی بر روی هر دو مجموعه‌داده (اصلی و مصنوعی)
* تحلیل ماتریس همبستگی: بررسی همبستگی بین ویژگی‌ها در هر دو مجموعه‌داده برای اطمینان از حفظ روابط میان ویژگی‌ها در داده‌های مصنوعی
* آزمایش Cross-Dataset: استفاده از داده‌های مصنوعی برای آموزش و داده‌های اصلی برای تست (و بالعکس) برای ارزیابی قابلیت تعمیم‌پذیری مدل‌ها
* مقایسه با پژوهش‌های پیشین: مقایسه نتایج با کارهای قبلی در حوزه تشخیص تزریق SQL و NoSQL

معیارهای مقایسه شامل:

* صحت و دقت تشخیص (Accuracy, Precision, Recall, F1-Score)
* نرخ مثبت کاذب و منفی کاذب
* زمان آموزش و تست مدل‌ها
* پیچیدگی محاسباتی
* قابلیت تعمیم‌پذیری و استحکام (Robustness) در برابر انواع مختلف حملات
* پژوهش حاضر و رویکرد اجرایی

پژوهش حاضر بر استفاده از مدل‌های یادگیری ماشینی متکی است که با استفاده از مجموعه‌داده‌های تولیدشده توسط مدل‌های زبان بزرگ (GPT) آموزش داده می‌شوند. این مجموعه‌داده‌ها شامل داده‌های مصنوعی با ویژگی‌هایی شبیه به داده‌های واقعی هستند و نقش کلیدی در بهبود عملکرد مدل‌ها ایفا می‌کنند.

فرایند اجرایی پژوهش به‌صورت زیر است:

۱. جمع‌آوری و برچسب‌گذاری داده‌های اولیه: ایجاد مجموعه‌داده پایه از نمونه‌های واقعی حملات و درخواست‌های عادی

۲. مهندسی ویژگی دستی: استخراج ویژگی‌های مؤثر و خاص برای تشخیص حملات NoSQL

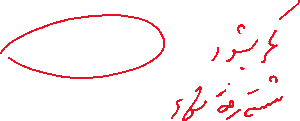
۳. تولید داده‌های مصنوعی با GPT: استفاده از مدل‌های زبان بزرگ برای تولید داده‌های متنوع و برچسب‌گذاری‌شده

۴. آموزش مدل‌های مختلف یادگیری ماشینی: پیاده‌سازی و آموزش شش مدل فردی و یک مدل گروهی

5. ارزیابی و مقایسه: ارزیابی جامع عملکرد مدل‌ها بر روی مجموعه‌داده‌های مختلف و مقایسه نتایج

تنظیم دقیق پارامترهای مدل و استفاده از تکنیک‌های پیشرفته نظیر یادگیری گروهی (Ensemble Learning)، منجر به بهبود قابل‌توجه در دقت تشخیص حملات و کاهش نرخ هشدارهای کاذب می‌شود.

## 1-6 نوآوري پژوهش



در این پژوهش، با استفاده از مدل‌های زبان بزرگ برای تولید داده‌های مصنوعی مرتبط با حملات MongoDB، یک مجموعه‌داده جامع و متنوع ایجاد شده است. همچنین، یک سیستم یادگیری گروهی طراحی شده که از ترکیب چندین مدل یادگیری ماشینی برای بهبود دقت تشخیص استفاده می‌کند. این رویکرد جامع، خلأ موجود در پژوهش‌های امنیت سایبری را پر کرده و گامی مهم در راستای ارتقای امنیت پایگاه‌داده‌های غیررابطه‌ای برداشته است.

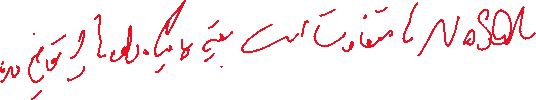
نوآوری اصلی این پژوهش در تمرکز بر حملات تزریق پایگاه‌داده غیررابطه‌ای است که به‌دلیل تفاوت‌های ساختاری اساسی با پایگاه‌داده‌های رابطه‌ای، نیازمند رویکردهای امنیتی متفاوت و پیچیده‌تری هستند. برخلاف پایگاه‌داده‌های رابطه‌ای که از ساختار جدولی و زبان پرس‌وجوی استاندارد SQL استفاده می‌کنند، پایگاه‌داده‌های غیررابطه‌ای مانند MongoDB از ساختارهای غیرجدولی و زبان‌های پرس‌وجوی خاص خود بهره می‌برند. این تفاوت‌ها باعث می‌شود که حملات تزریق در این نوع پایگاه‌ها از الگوهای پیچیده‌تر و غیرمعمول‌تری استفاده کنند که شناسایی آن‌ها با روش‌های سنتی مبتنی بر SQL دشوار است.

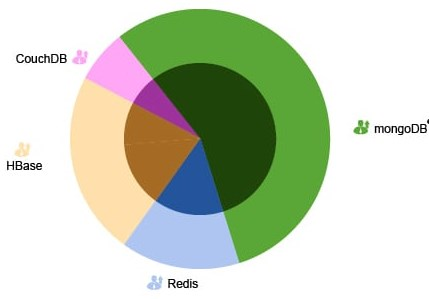
فصل پنجم: جمع‌بندی و کارهای آتی

در این فصل به بررسی مفاهیم بنیادی و پژوهش‌های مرتبط با این موضوع اختصاص دارد. ابتدا نقاط ضعف و چالش‌های امنیتی مرتبط با حملات تزریق داده در محیط‌های پایگاه‌داده غیررابطه‌ای بررسی می‌شود. سپس، مبانی یادگیری ماشینی، الگوریتم‌های مرتبط، و کاربردهای آن‌ها در امنیت سایبری موردبحث قرار می‌گیرد. در ادامه، مدل‌های زبان بزرگ به‌عنوان رویکردی نوین برای تولید داده‌های مصنوعی معرفی و با روش‌های سنتی مانند شبکه‌های مولد متخاصم مقایسه می‌شوند. همچنین، تحقیقات پیشین در زمینه تشخیص حملات تزریق پایگاه‌داده رابطه‌ای و پایگاه‌داده غیررابطه‌ای مرور شده و دستاوردهای کلیدی آن‌ها تحلیل می‌شود. درنهایت، محدودیت‌های پژوهش‌های پیشین و نقاط تمایز رویکرد پیشنهادی این پژوهش نسبت به آن‌ها بیان می‌گردد.

### 2-1-1 شناخت پایگاه‌داده‌های غیررابطه‌ای

پايگاه‌داده Mongodb به دليل فراگيرتر بودن در توسعه نرم‌افزارها هدف اصلي پياده‌سازي ماست. ازآنجايي‌که مجموعه‌داده عمدتاً از طريق خراش دادن وب[[7]](#footnote-8) و داده‌هاي افزوده به دست مي‌آيد، هدف خاص آموزش مدلي متناسب با پايگاه‌داده MongoDB را برآورده مي‌کند. ترکيب مجموعه‌داده براي دستيابي به اهداف موردنظر در اين مقاله کافي است.MongoDB محبوب‌ترين پایگاه‌داده غیررابطه‌ای است که در بين 500 سازمان برتر جهان مورد پذيرش قابل‌توجهي قرار گرفته است [27].





MONGODB امکانات CURD را فراهم می‌کند و سیستم پایگاه‌داده امکان جستجو و ویرایش اطلاعات موجود در آن را برای کاربران فراهم می‌کند. تزریق مخرب می‌تواند منجر به مشکلات امنیتی جدی شود و هکرها می‌توانند کدهای مخرب را با تزریق به ورودی اجرا کنند تا به اطلاعات پایگاه‌داده دسترسی پیدا کنند.

|  |
| --- |
| پایگاه‌داده رابطه‌ای:  “select \* from student where (phone = ‘ “ + ph\_number +” ’);”  MongoDB :  db.collection.find ({phone= ph\_number}) |

MongoDB ارائه‌دهنده مقیاس‌پذیری بالا، کارایی عالی و امکانات موردنیاز برای مقیاس‌پذیری است. این پایگاه‌داده از فرمت‌های JSON و BSON پشتیبانی می‌کند و توانایی مدیریت میلیون‌ها یا حتی میلیاردها رکورد را دارا است. از "فهرست‏گذاری بر روي رشته‏ها"[[8]](#footnote-9) و شی‌ء گذاری[[9]](#footnote-10) پشتیبانی می‌کند و به‌صورت زمان واقعی عمل می‌کند. همچنین زبان پرس‌وجو MongoDB به نام Mongo است.

يادگيري ماشيني كه بر پايه جبر خطي است، داراي تنوع زيادي بوده که مي‌توان در بسياري از حوزه‌ها از آن استفاده کرد، اما براي بهره‏بردن از معماري‌هاي حوزه خاصي مانند واحد پردازش تنسور گوگل محدود است. همچنين، رشد تقاضا براي محاسبات یادگیری ماشین از قانون مور فراتر رفته و نياز است کارشناسان یادگیری ماشین و معماران رايانه براي طراحي سامانه‌هاي محاسباتي موردنياز براي ارائه امکانات یادگیری ماشین به يکديگر کمک کنند. براين‌اساس مي‏توان پيشنهادها و هشدارهايي به معماران رايانه در مورد تحولات يادگيري ماشين ارائه داد.

يک مدل يادگيري ماشين عملکرد حداکثري خواهد داشت اگر داده‌هاي آن از نظر كيفي و كمي حد مناسبي داشته باشند. درحالي‌که پژوهشگران بر روي بهبود کيفيت مدل‌ها (مانند جستجوي معماري عصبي و انتخاب خودکار ويژگي) تمرکز داشته‌اند، تلاش‌هاي محدودي براي بهبود کيفيت داده‌ها انجام شده است. يکي از نيازهاي حياتي پيش از استفاده از مجموعه‌داده براي هر كاري، درک مجموعه‌داده آن حوزه است و عدم انجام اين کار ممکن است منجر به تجزيه‌وتحليل نادرست و تصميم‌گيري‌هاي ناقص شود.

**[منتقل‌شده از فصل ۱]**

طبق يافته‌هاي قبلي مي‏دانيم كه امروزه حملات تزريق از خطرسازترين حملات محسوب مي‌شوند، تحقيقات زيادي در حوزه پايگاه‌داده‌هاي ساخت‌يافته صورت پذيرفته است، كه مي‏توان با تعميم اين كارها به پايگاه‌داده‌هاي ديگر اين مسئله را از ابعاد جديدي مورد بررسي قرارداد.

### 1-2-3 بررسي اهميت پايگاه‏داده‏هاي غيررابطه‏اي

### 2-1-2 بررسی نقاط ضعف و چالش‌های امنیتی مرتبط با حملات تزریق داده در محیط‌های پایگاه‌داده غیررابطه‌ای

یکی از مهم‌ترین ویژگی‌های پایگاه‌داده‌های غیررابطه‌ای، ساختار منعطف آن‌ها است که برخلاف پایگاه‌داده‌های رابطه‌ای پایگاه‌داده رابطه‌ای، از مدل‌های داده متنوعی مانند مستند محور، گرافی، کلید - مقدار و جدولی پشتیبانی می‌کند. این انعطاف‌پذیری اگرچه مزایای قابل‌توجهی دارد، اما به دلیل نبود یک استاندارد امنیتی جامع، بستری برای سوءاستفاده مهاجمان فراهم می‌کند. بسیاری از پایگاه‌داده‌های غیررابطه‌ای بدون کنترل دسترسی قوی یا ابزارهای احراز هویت اولیه عرضه می‌شوند که می‌تواند منجر به نفوذ و تزریق داده‌های مخرب شود[29].

بسیاری از پایگاه‌داده‌های غیررابطه‌ای از سیاست‌های کنترل دسترسی ساده یا ناکافی استفاده می‌کنند. این موضوع، به‌ویژه در مواردی که سیستم‌ها برای کارایی بیشتر بهینه شده‌اند، می‌تواند منجر به دسترسی غیرمجاز به داده‌های حساس شود. برای مثال، در برخی از پایگاه‌داده‌های غیررابطه‌ای، احراز هویت به‌صورت پیش‌فرض غیرفعال است یا تنظیمات امنیتی به طور کامل به توسعه‌دهندگان واگذار می‌شود.

اگرچه پایگاه‌داده‌های غیررابطه‌ای به‌طورکلی برای مقیاس‌پذیری طراحی شده‌اند، اما تضمین امنیت در مقیاس بالا یک چالش اساسی است. با افزایش تعداد گره‌ها و حجم داده‌ها، مدیریت امنیت به‌صورت یکپارچه دشوارتر می‌شود و ممکن است نقاط ضعف جدیدی در سیستم ایجاد کند.

پایگاه‌داده‌های غیررابطه‌ای به دلیل توانایی مدیریت داده‌های غیرساختاریافته، به طور گسترده در اینترنت اشیا استفاده می‌شوند. اما ترکیب حجم بالای داده‌های تولیدشده در اینترنت اشیا و نبود سیاست‌های امنیتی قوی در پایگاه‌داده‌های غیررابطه‌ای، یک نقطه‌ضعف عمده ایجاد می‌کند. مهاجمان می‌توانند از این ضعف‌ها برای حملات تزریق و دسترسی غیرمجاز به داده‌های حساس دستگاه‌های اینترنت اشیا استفاده کنند[30].

فصل پنجم شامل جمع‌بندی و کارهای آتی است. در این فصل، دستاوردهای اصلی پژوهش، مرور نوآوری‌ها، بیان محدودیت‌ها و ارائه پیشنهادهای مشخص برای توسعه و پژوهش‌های آینده ارائه می‌شود.

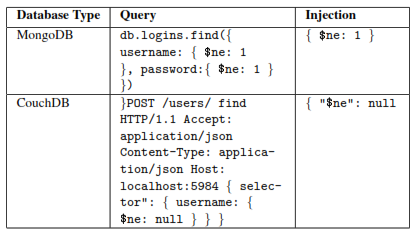
فصل چهارم به تحلیل نتایج و بحث اختصاص دارد. در این فصل، مراحل آماده‌سازی داده‌ها، اجرای مدل‌ها و نحوه ارزیابی بیان می‌شود. سپس عملکرد مدل‌ها با معیارهایی مانند دقت، شاخص‌های خطا، و معیارهای زمانی و کارایی مقایسه و تحلیل می‌گردد. همچنین، تفسیر یافته‌ها، مقایسه با پیشینه پژوهش، بررسی نقاط قوت و محدودیت‌ها، و تحلیل اثر داده‌های مصنوعی و انتخاب مدل‌ها بر عملکرد و تعمیم‌پذیری ارائه می‌شود.

فصل سوم روش پیشنهادی را تشریح می‌کند. این فصل شامل چارچوب کلی پژوهش، فرایند تولید داده مصنوعی مبتنی بر مدل‌های زبان بزرگ، معرفی مدل‌های یادگیری ماشین و مدل‌های گروهی به‌کاررفته و منطق انتخاب آن‌ها، تنظیمات کلیدی و پارامترها، و معماری پیاده‌سازی سیستم پیشنهادی است.

فصل دوم به بررسی مبانی نظری و پیشینه پژوهش اختصاص دارد. در این فصل، مفاهیم پایه مرتبط با حملات تزریق در پایگاه‌داده‌های غیررابطه‌ای، کاربردهای یادگیری ماشین در امنیت، و رویکردهای تولید داده مصنوعی با مدل‌های زبان بزرگ در مقایسه با روش‌های سنتی مرور می‌شوند. همچنین، شکاف‌های پژوهشی و تمایزات رویکرد پیشنهادی جمع‌بندی می‌گردد.

این پایان‌نامه در شش فصل تنظیم شده است و مسیر پژوهش را از مبانی نظری تا جمع‌بندی و پیشنهادهای آتی به‌صورت منسجم دنبال می‌کند.

تکنیک‌هایی که برای تزریق به پایگاه‌داده‌های پایگاه‌داده رابطه‌ای و پایگاه‌داده غیررابطه‌ای استفاده می‌شوند، تا حد زیادی مشابه هستند. در اینجا چهار نوع از تزریق‌های پایگاه‌داده غیررابطه‌ای توضیح داده شده‌اند:



براي نمونه در مدل‌هاي مبتني بر يادگيري عميق، بدون وابستگي به قواعد پايه پايگاه‌داده، از يک مدل پردازش زبان طبيعي و چارچوب يادگيري عميق استفاده مي‌کند. اين روش مي‌تواند دقت را افزايش داده و نرخ هشدارهاي غلط را کاهش دهد درحالي‌که به ماشين اجازه می‌دهد به طور خودکار ويژگي‌هاي مدل زبان حملات تزريق پرس‌وجو را ياد بگيرد، کاهش قابل‌توجهي نيز در دخالت انسان و ارائه دفاع در برابر حملات Zeroday که هرگز رخ نمي‌دهد، فراهم مي‌کند.

بررسي رويكردهاي يادگيري ماشين و يادگيري عميق

بررسي رويكردهاي سنتي مبتني بر امضا

با اينکه اين آسيب‌پذيري بيش از 20 سال است كه شناخته شده، حملات تزريق هنوز در رتبه سوم آسيب‌پذيري‌هاي وب قرار دارند. در سال 2022، 1162 آسيب‌پذيري با نوع تزريق‌ پرس‏وجو به‌عنوان CVE پذيرفته شده‌اند كه مي‏توان اين برداشت را داشت كه حملات تزريق پرس‏وجو هنوز موجود هستند .

شکل 1- 1: توزيع آسيب‌پذيري‌هاي بحراني برنامه‌هاي وب در سراسر جهان تا سال 2022

تزريق پایگاه‌داده رابطه‌ای نوعي حمله سايبري است که برنامه‌هاي مبتني بر داده را هدف قرار می‌دهد. اين اقدام شامل درج عبارات پرس‌وجوي مخرب در فيلدهاي ورودي يا عامل‌هاي يک برنامه وب براي دست‌کاري يا استخراج داده‌هاي حساس از پايگاه‌داده بوده و يكي از آسيب‏پذيري‏هاي اساسي آن‏هاست. در بررسي انجام شده سال 2022 مشخص گرديد كه 33 درصد از آسيب‌پذيري‌هاي حياتي اينترنت، ناشي از اين حملات بوده است. اين امر نشان می‌دهد كه سازمان‌ها نياز فوري به انجام اقدامات پيشگيرانه در مقابله با حملات تزريق پرس‏وجو دارند.

### 1-2-1 تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای

**[منتقل‌شده از فصل ۱]**

ممکن است بتوان عملگرهاي پرس‏وجو را براي دست‌کاری آن‏ها در پایگاه‌داده غیررابطه‌ای تزريق كرد. براي انجام اين کار، اپراتورهاي مختلف را به طور سيستماتيک در طيفي از ورودی‌های کاربر ارسال کرده و سپس پاسخ‌ها را براي پیام‌های خطا و يا تغييرات ديگر بررسي بايد كرد.

جدول 1- 1: نمونه هايي از عملگرهاي پرس‌وجو MongoDB

در پایگاه‌داده غیررابطه‌ای اغلب از عملگرهاي پرس‌وجو استفاده مي‏شود که راه‏هايي را براي تعيين شرايطي که داده‌ها بايد داشته باشند تا در نتيجه پرس‌وجو قرار گيرند، ارائه می‌دهند. نمونه‌هایی از عملگرهاي پرس‌وجو MongoDB در جدول 1-1 آمده است.

راه‏هاي مختلف ديگري نيز براي ايجاد بار براي اين دست حملات وجود دارد كه مثال فوق يك نمونه از دست‌کاری در نگارش پرس‏وجو است

اگر اجرا اين آدرس باعث تغيير در پاسخ اصلي شود، ممکن است نشان‌دهنده اين باشد که ورودي کاربر به‌درستی فيلتر يا تصفيه نشده است.

از اين رشته fuzz براي ساخت حمله زير استفاده مي‏شود:

براي تست اينکه آيا ورودي ممکن است آسیب‌پذیر باشد، یک‌رشته fuzz را در مقدار پارامتر دسته‌بندی ارسال كرده كه یک‌رشته نمونه براي MongoDB به‌صورت زير است:

اين باعث مي‌شود که برنامه يک پرس‏وجو JSON براي بازيابي محصولات مربوطه از مجموعه محصولات در پایگاه‌داده MongoDB ارسال کند:

يک وبگاه خريد که محصولات را در دسته‌بندي‌هاي مختلف نمايش می‌دهد را در نظر گرفته. هنگامي که کاربر دسته‌بندي نوشيدني‌هاي گازدار را انتخاب مي‌کند، مرورگر آن‌ها URL زير را درخواست می‌دهد:

به‌عنوان‌مثال مي‏توان اين حملات را در پایگاه‌داده MongoDB مورد بررسي قرارداد تا اين دو نوع حمله بيشتر آشنا شد:

در اين پژوهش، نحوه آزمايش آسيب‌پذيري‌هاي پایگاه‌داده غیررابطه‌ای را به‌طورکلی بررسي كرده، سپس بر روي بهره‌برداري از حملات در يكي از انواع پایگاه‌داده غیررابطه‌ای تمرکز خواهد داشت.

شکل 1- 3: يك سناريو دور زدن احراز هويت

تزريق عملگر - زماني اتفاق می‌افتد که می‌توان از عملگرهاي پرس‏وجو پایگاه‌داده غیررابطه‌ای براي دست‌کاری پرس‏وجوها استفاده کرد.

تزريق با اشتباه نوشتاري: زماني اتفاق مي‌افتد که مي‌توان نوشتار پرس‌وجو پایگاه‌داده غیررابطه‌ای را تغيير داده و محموله خود را تزريق کرد. اين متدولوژي مشابه روشي است که در تزريق پایگاه‌داده رابطه‌ای استفاده می‌شود. بااین‌حال ماهيت حمله به طور قابل‌توجهی متفاوت است، زيرا پايگاه‏هاي داده پایگاه‌داده غیررابطه‌ای از طيف وسيعي از زبان‏هاي پرس‏وجو، انواع نحو پرس‏وجو و ساختارهاي داده متفاوت استفاده مي‏کنند.

دو نوع مختلف تزريق پایگاه‌داده غیررابطه‌ای وجود دارد:

پایگاه‌داده غیررابطه‌ای داده‏ها را در قالبي غير از جداول سنتي پایگاه‌داده رابطه‌ای ذخيره و بازيابي مي‏کنند. آن‏ها از طيف گسترده‌ای از زبان‌های پرس‌وجو به‌جای يک استاندارد جهاني مانند پایگاه‌داده رابطه‌ای استفاده می‌کنند و محدودیت‌های رابطه‏اي کمتري دارند.

اجراکردن کد روی سرور

باعث انکار خدمات

استخراج يا ويرايش داده‏ها

دورزدن مکانیسم‌های احراز هويت يا حفاظت

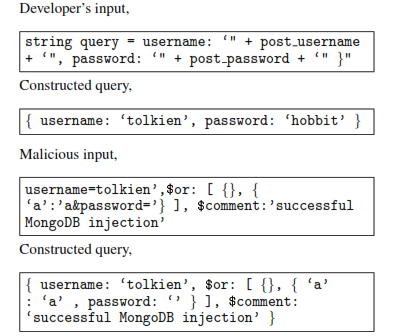
تزريق پایگاه‌داده غیررابطه‌ای آسيب‌پذيري است که در آن يک مهاجم مي‌تواند با استفاده از پرس‏وجوهايي که يک برنامه کاربردي در پایگاه‌داده غیررابطه‌ای دارد تداخل‏هاي ايجاد كرده كه اين تزريق پایگاه‌داده غیررابطه‌ای توانمندي‏هايي نظير زير را به او می‌دهد:

### 1-2-4 ارائه نمونه‌ای حمله تزريق پایگاه‌داده غیررابطه‌ای

**[منتقل‌شده از فصل ۱]**

این پرس‌وجوها اطلاعاتی را نشان می‌دهند که در آن نام کاربری و رمز عبور NULL نیستند. هکرها می‌توانند از عملگر "$ne" (به معنای "نابرابر") استفاده کنند تا بدون واردکردن نام کاربری و رمز عبور صحیح به سیستم وارد شوند[30].

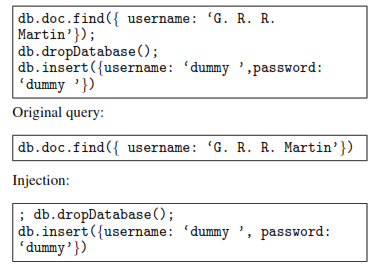
* پرس‌وجوهای Union: هکر از یک پارامتر آسیب‌پذیر برای تغییر داده‌هایی که قرار بود از یک پرس‌وجو بازگردانده شود، استفاده می‌کند. یک شرط OR برای اتصال یک عبارت خالی به ورودی استفاده می‌شود. ازآنجاکه یک عبارت خالی همیشه معتبر است، بررسی رمز عبور بی‌اثر می‌شود. به‌عنوان‌مثال:



در اینجا پرس‌وجوی خالی {} همیشه مقدار true دارد [

* تزریق جاوا اسکریپت[[10]](#footnote-11) : پایگاه‌های داده پایگاه‌داده غیررابطه‌ای اجازه اجرای کدهای جاوا اسکریپت را می‌دهند و می‌توانند پرس‌وجوها و تراکنش‌های پیچیده‌ای را روی موتور پایگاه‌داده اجرا کنند. اگر ورودی کاربر فیلتر یا اعتبارسنجی نشود، ممکن است خطر تزریق کدهای تصادفی جاوا اسکریپت وجود داشته باشد.
* پرس‌وجوهای زنجیره‌ای [[11]](#footnote-12) : در این روش، هکر از توالی‌های فرار[[12]](#footnote-13) و کاراکترهای ویژه‌ای مانند بازگشت به خط [[13]](#footnote-14)، خط جدید [[14]](#footnote-15)، آکولادهای بسته، و علامت نقطه‌ویرگول برای پایان‌دادن به یک پرس‌وجو و سپس افزودن پرس‌وجوهای مخرب اضافی برای اجرا استفاده می‌کند. این عمل می‌تواند پایگاه‌داده را به‌شدت خراب کند. برای مثال:

پرس‌وجو:



در اینجا، پس از یک نقطه‌ویرگول (;)، هکر یک پرس‌وجوی مخرب اضافی را تزریق می‌کند[30].نقض مبدا[[15]](#footnote-16) : رابط برنامه‌نویسی REST در HTTP، نوع جدیدی از آسیب‌پذیری‌ها را به وجود آورده است که به مهاجم اجازه می‌دهد حتی از یک دامنه دیگر به پایگاه‌داده غیررابطه‌ای حمله کند. در حملات Cross-Origin، یک کاربر مجاز و مرورگر وب او مورد سوءاستفاده قرار می‌گیرند تا هکر بتواند عملیات نامطلوبی را اجرا کند. به شکل جعل درخواست بین‌سایتی ، این حمله زمانی رخ می‌دهد که اعتماد سایتی به مرورگر کاربر مورد سوءاستفاده قرار می‌گیرد تا یک عملیات غیرقانونی روی پایگاه‌داده غیررابطه‌ای اجرا شود. با تزریق یک فرم HTیادگیری ماشین در یک وب‌سایت آسیب‌پذیر یا فریب کاربر برای بازدید از وب‌سایت شخصی هکر، هکر می‌تواند یک عملیات پست را روی پایگاه‌داده اجرا کند.

### 2-1-3 یادگیری ماشین

یادگیری ماشین یکی از شاخه‌های برجسته هوش مصنوعی است که به ماشین‌ها امکان می‌دهد از داده‌ها یاد بگیرند و بدون نیاز به برنامه‌نویسی صریح، عملکرد خود را بهبود دهند. در حوزه امنیت سایبری، یادگیری ماشین نقش مهمی در شناسایی تهدیدات، پیشگیری از حملات، و تحلیل رفتارهای غیرعادی ایفا می‌کند. این بخش به‌مرور مفاهیم بنیادین یادگیری ماشین، بررسی الگوریتم‌های مرتبط، و کاربردهای آن‌ها در امنیت سایبری اختصاص دارد.

یادگیری ماشین فرایندی است که در آن سیستم‌های کامپیوتری توانایی یادگیری از داده‌ها و بهبود عملکرد خود در انجام وظایف خاص را پیدا می‌کنند. این فرایند شامل مراحل زیر است[32]:

* ورود داده‌ها: داده‌ها به‌عنوان ورودی به سیستم داده می‌شوند.
* استخراج ویژگی‌ها: ویژگی‌های مرتبط با مسئله از داده‌ها استخراج می‌شوند.
* مدل‌سازی: مدل‌های آماری یا الگوریتم‌های یادگیری برای شناسایی الگوها استفاده می‌شوند.
* پیش‌بینی: مدل، داده‌های جدید را تحلیل کرده و خروجی پیش‌بینی می‌کند.
* ارزیابی: عملکرد مدل بر اساس معیارهایی نظیر دقت، یادآوری و نرخ مثبت کاذب ارزیابی می‌شود.

یادگیری ماشین به سه دسته اصلی تقسیم می‌شود:

1. یادگیری نظارت شده[[16]](#footnote-17) :

در این روش، مدل‌ها از داده‌های برچسب‌گذاری شده آموزش می‌بینند. این رویکرد برای مسائل دسته‌بندی و پیش‌بینی مناسب است.

* + مثال: تشخیص حملات سایبری بر اساس الگوهای مشخص حملات پیشین.

1. یادگیری بدون نظارت: [[17]](#footnote-18)

مدل‌ها در این روش از داده‌های بدون برچسب استفاده می‌کنند و به کشف ساختارها و الگوهای مخفی در داده‌ها می‌پردازند.

* + مثال: شناسایی خوشه‌های مشکوک در ترافیک شبکه.

1. یادگیری تقویتی[[18]](#footnote-19) :

این روش مبتنی بر یادگیری از طریق تعامل با محیط است، به‌طوری‌که عامل هوشمند با دریافت پاداش یا تنبیه به یادگیری می‌پردازد.

* + مثال: طراحی سیستم‌های تطبیقی برای شناسایی حملات در زمان واقعی.

#### 2-1-3-1 الگوریتم‌های مرتبط در یادگیری ماشین

الگوریتم‌های پایه‌ای[33]

1. رگرسیون لجستیک

این الگوریتم برای مسائل دسته‌بندی دودویی استفاده می‌شود و احتمال وقوع یک رویداد را مدل‌سازی می‌کند.

* + کاربرد: شناسایی بسته‌های شبکه‌ای مخرب.

1. درخت تصمیم[[19]](#footnote-20)

این مدل بر اساس گره‌هایی که معیارهای تصمیم‌گیری خاصی را بررسی می‌کنند، عمل می‌کند.

* + کاربرد: تحلیل رفتار کاربران و شناسایی دسترسی‌های غیرمجاز.

1. ماشین بردار پشتیبان

الگوریتمی است که داده‌ها را با استفاده از یک ابر صفحه به دودسته جدا می‌کند.

* + کاربرد: تشخیص بدافزار با تفکیک ویژگی‌های بدافزار از نرم‌افزارهای سالم.

1. االگوریتم (KNN[[20]](#footnote-21)) یکی از ساده‌ترین و در عین حال قدرتمندترین الگوریتم‌های یادگیری ماشین است که در دسته‌ی یادگیری نظارت‌شده قرار می‌گیرد. این الگوریتم برای طبقه‌بندی و رگرسیون به کار می‌رود، اما عمدتاً در مسائل طبقه‌بندی مورد استفاده قرار می‌گیرد.

الگوریتم‌های پیشرفته[34]

1. جنگل تصادفی

یک الگوریتم یادگیری گروهی که از ترکیب چندین درخت تصمیم برای افزایش دقت و کاهش بیش برازش استفاده می‌کند.

* + کاربرد: تحلیل داده‌های بزرگ امنیتی.

1. XGBoost:

یکی از قدرتمندترین الگوریتم‌های گروهی است که برای مسائل طبقه‌بندی و رگرسیون استفاده می‌شود.

* + کاربرد: شناسایی حملات پیچیده در ترافیک شبکه.

1. یادگیری عمیق:

این روش بر معماری‌های شبکه عصبی عمیق تکیه دارد و توانایی درک داده‌های پیچیده؛ مانند تصاویر، صدا، و متون را دارد.

* + کاربرد: شناسایی بدافزارهای ناشناخته.

در جدول زیر، مقایسه‌ای دقیق از مدل‌های رگرسیون لجستیک، جنگل تصادفی و XGBoost بر اساس معیارهای مختلف ارائه شده است:

جدول3- 1: مقایسه‌ای دقیق از مدل‌ها

|  |  |  |  |
| --- | --- | --- | --- |
| معیار | رگرسیون خطی | درخت تصادفی | XGBoost |
| نوع الگوریتم | مدل خطی | یادگیری گروهی | تقویت گرادیان |
| پیچیدگی محاسباتی | پایین | متوسط | بالا |
| تعامل با داده‌های غیرخطی | ضعیف | قوی | بسیار قوی |
| مقاومت در برابر بیش برازش | متوسط | بالا | بسیار بالا |
| سرعت پردازش | سریع | متوسط | سریع |
| تفسیرپذیری مدل | بسیار ساده | متوسط | پیچیده |
| مدیریت داده‌های نامتوازن | ضعیف | متوسط | قوی |
| نیاز به تنظیم پارامترها | کم | متوسط | زیاد |
| دقت در داده‌های پیچیده | پایین | بالا | بسیار بالا |

در ادامه کاربرد یادگیری ماشین در امنیت سایبری را شرح می‌دهیم[35]:

* کاربردهای تشخیص نفوذ: یکی از مهم‌ترین کاربردهای یادگیری ماشین، شناسایی نفوذ و فعالیت‌های غیرمجاز در سیستم‌های کامپیوتری است. الگوریتم‌ها با تحلیل ترافیک شبکه می‌توانند رفتارهای مشکوک را شناسایی و اقدامات لازم را انجام دهند.
* شناسایی بدافزار: الگوریتم‌های یادگیری ماشینی می‌توانند با تحلیل ویژگی‌های فایل‌ها و کدها، بدافزارها را شناسایی کنند. به‌عنوان‌مثال، درخت تصادفی و XGBoost برای شناسایی بدافزارهای پیچیده بسیار مؤثر هستند.
* تحلیل لاگ‌ها: سیستم‌های امنیتی از یادگیری ماشینی برای تحلیل حجم عظیمی از لاگ‌های تولیدشده توسط شبکه‌ها و سیستم‌ها استفاده می‌کنند. این تحلیل به شناسایی الگوهای مخرب کمک می‌کند.
* تشخیص حملات تزریق: الگوریتم‌های نظارت شده نظیر رگرسیون خطی و SVM برای تشخیص حملات تزریق پایگاه‌داده رابطه‌ای و پایگاه‌داده غیررابطه‌ای با تحلیل داده‌های ورودی و الگوهای مخرب استفاده می‌شوند.
* پیشگیری از حملات فیشینگ: با استفاده از یادگیری ماشینی، ایمیل‌ها و صفحات وب مشکوک بررسی و الگوهای حملات فیشینگ شناسایی می‌شوند. الگوریتم‌های نظارت شده برای این منظور بسیار کاربردی هستند.
* شناسایی رفتارهای مشکوک کاربران: الگوریتم‌های بدون نظارت، مانند خوشه‌بندی، می‌توانند رفتارهای کاربران را مدل‌سازی کرده و رفتارهای غیرعادی را شناسایی کنند.
* سیستم‌های توصیه‌گر امنیتی: با تحلیل الگوهای تهدید و آسیب‌پذیری، مدل‌های یادگیری ماشینی می‌توانند پیشنهادهایی برای ارتقا امنیت سیستم‌ها ارائه دهند.

چالش‌ها و محدودیت‌های یادگیری ماشین در امنیت سایبری عبارت‌اند از:

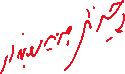
* کمبود مجموعه‌داده‌های جامع: بسیاری از مدل‌های یادگیری ماشینی برای عملکرد بهینه به داده‌های بزرگ و متنوع نیاز دارند. در حوزه امنیت سایبری، مجموعه‌داده‌های استاندارد و متنوع به‌سختی قابل‌دسترس هستند.
* حملات به مدل‌های یادگیری ماشینی: مهاجمان ممکن است مدل‌های یادگیری ماشینی را هدف حملات خود قرار دهند. برای مثال:
* حملات تزریق داده مخرب: تزریق داده‌های جعلی برای تغییر عملکرد مدل.
* حملات استنتاج مدل: شناسایی رفتارهای مدل برای استفاده از نقاط ضعف آن.
* تغییرات در الگوهای تهدید: الگوهای حملات سایبری به‌سرعت تغییر می‌کنند که باعث می‌شود مدل‌های یادگیری ماشینی نیاز به به‌روزرسانی مداوم داشته باشند.

یادگیری ماشینی به‌عنوان یکی از ابزارهای کلیدی در امنیت سایبری، توانایی تحلیل و شناسایی تهدیدات پیچیده را فراهم می‌کند. از تشخیص بدافزارها گرفته تا شناسایی رفتارهای مشکوک کاربران، این فناوری نقش مهمی در ایمن‌سازی سیستم‌ها ایفا می‌کند. بااین‌حال، چالش‌هایی نظیر کمبود داده‌های استاندارد و تغییرات سریع در الگوهای تهدیدات نشان می‌دهد که توسعه و پیاده‌سازی راهکارهای یادگیری ماشینی در امنیت سایبری نیازمند تحقیق و تلاش مداوم است.

در این پژوهش، تلاش می‌شود با استفاده از مدل‌های پیشرفته نظیر درخت تصادفی و XGBoost و تولید مجموعه‌داده‌های مصنوعی با مدل‌های زبان بزرگ، این چالش‌ها تاحدامکان مرتفع شوند و راهکاری جامع برای مقابله با حملات سایبری ارائه شود.

### 2-1-3-2 معرفی مدل‌های زبان بزرگ و مقایسه آن‌ها با روش‌های سنتی درتولید داده‌های مصنوعی

مدل‌های زبان بزرگ و شبکه‌های مولد متخاصم هر دو به‌عنوان ابزارهای پیشرفته در تولید داده‌های مصنوعی مطرح هستند. این مدل‌ها، هرچند در اهداف و کاربردها مشترکاتی دارند، اما از لحاظ ساختار، روش کار و قابلیت‌ها تفاوت‌های چشمگیری نشان می‌دهند. در این بخش، ابتدا مدل‌های زبان بزرگ معرفی می‌شوند و سپس مقایسه‌ای جامع میان این مدل‌ها و روش‌های سنتی مانند شبکه‌های مولد متخاصم ارائه می‌شود.[36]



مدل‌های زبان بزرگ، معماری‌هایی مبتنی بر شبکه‌های عصبی ترانسفورمر[[21]](#footnote-22)هستند که برای پردازش زبان طبیعی طراحی شده‌اند. این مدل‌ها به دلیل آموزش بر روی مقادیر عظیمی از داده‌های متنی، قابلیت درک و تولید متون پیچیده را دارا هستند. ویژگی‌های کلیدی مدل‌های زبان بزرگ عبارت‌اند از:

* معماری ترانسفورمر[37]:



* این مدل‌ها از معماری ترانسفورمر بهره می‌برند که امکان پردازش داده‌ها به‌صورت موازی و یادگیری روابط طولانی‌مدت میان کلمات و جملات را فراهم می‌کند.
* مقیاس‌پذیری بالا
* مدل‌های زبان بزرگ‌ها بر روی مجموعه‌داده‌های عظیم و با استفاده از میلیاردها پارامتر آموزش‌دیده‌اند.
* توانایی تولید داده‌های متنی واقعی:
* مدل‌های زبان بزرگ می‌توانند متن‌هایی تولید کنند که از نظر ساختار و معنا به متون واقعی شباهت زیادی دارند.
* کاربردهای مدل‌های زبان بزرگ در تولید داده‌های مصنوعی
* شبیه‌سازی داده‌های متنی:
* مدل‌های زبان بزرگ‌ها قادرند داده‌های متنی مصنوعی تولید کنند که می‌تواند شامل پرس‌وجوهای پایگاه‌داده، پیام‌های کاربران، یا داده‌های امنیتی باشد.
* تولید داده‌های متنوع برای امنیت سایبری:
* این مدل‌ها می‌توانند داده‌های مرتبط با حملات سایبری مانند تزریق پایگاه‌داده غیررابطه‌ای را باکیفیت بالا و تنوع زیاد تولید کنند.
* ایجاد داده‌های ترکیبی:

مدل‌های زبان بزرگ‌ها می‌توانند داده‌هایی ترکیبی شامل متن و کد تولید کنند که برای آموزش مدل‌های یادگیری ماشینی در حوزه امنیت سایبری بسیار مفید است.

شبکه‌های مولد متخاصم معماری‌هایی متشکل از دو شبکه عصبی هستند:

* مولد[[22]](#footnote-23): وظیفه تولید داده‌های مصنوعی را دارد.
* تفکیک‌کننده[[23]](#footnote-24) : وظیفه تشخیص داده‌های مصنوعی از داده‌های واقعی را برعهده دارد.

این دو شبکه در رقابتی متقابل عمل می‌کنند، به‌طوری‌که مولد تلاش می‌کند داده‌هایی تولید کند که نتوانند توسط تفکیک‌کننده شناسایی شوند. ویژگی‌های کلیدی شبکه‌های مولد متخاصم عبارت‌اند از:

* تخصص در داده‌های غیرمتنی:

شبکه‌های مولد متخاصم در تولید داده‌های تصویری و سیگنال‌های پیچیده نظیر صدا یا ویدئو بسیار موفق عمل کرده‌اند.

* عملکرد مبتنی بر رقابت:

مولد و تفکیک‌کننده در فرایندی رقابتی بهبود پیدا می‌کنند که باعث افزایش کیفیت داده‌های تولیدشده می‌شود.

محدودیت‌های شبکه‌های مولد متخاصم را در ادامه بیان می‌کنیم:

* محدودیت در تولید داده‌های متنی پیچیده:

معماری شبکه‌های مولد متخاصم به طور خاص برای داده‌های متنی طراحی نشده است و در شبیه‌سازی داده‌های متنی که نیاز به درک زبان طبیعی دارد، محدودیت‌هایی دارد.

* چالش در همگرایی:

فرایند آموزش شبکه‌های مولد متخاصم گاهی دچار ناپایداری می‌شود و ممکن است مولد نتواند داده‌هایی باکیفیت تولید کند.

در ادامه این بخش مقایسه مدل‌های زبان بزرگ و شبکه‌های مولد متخاصم برای تولید داده‌های مصنوعی می‌پردازیم[39], [40]:

**1**-تفاوت در ساختار

* مدل‌های زبان بزرگ بر معماری ترانسفورمر و پردازش زبان طبیعی تمرکز دارد و برای تولید داده‌های متنی بسیار مناسب است.
* شبکه‌های مولد متخاصم برای داده‌های غیرمتنی (تصاویر، سیگنال‌ها و غیره) طراحی شده و بر رقابت میان مولد و تفکیک‌کننده استوار است.

2- توانایی در تولید داده‌های متنی

* مدل‌های زبان بزرگ: به‌دلیل تمرکز بر پردازش زبان، داده‌های متنی پیچیده و شبیه به واقعیت تولید می‌کند.
* شبکه‌های مولد متخاصم: توانایی کمتری در تولید داده‌های متنی با ساختار پیچیده دارد.

3- کاربرد در امنیت سایبری

* مدل‌های زبان بزرگ می‌تواند داده‌های مصنوعی مرتبط با حملات سایبری، پرس‌وجوهای تزریق پایگاه‌داده غیررابطه‌ای و متون مرتبط با امنیت را تولید کند.
* شبکه‌های مولد متخاصم بیشتر برای تولید داده‌های تصویری یا داده‌هایی که تحلیل بصری نیاز دارند (مانند تصاویر نقشه‌برداری بدافزارها) کاربرد دارد.

4- سهولت پیاده‌سازی

* مدل‌های زبان بزرگ تنظیم و استفاده از مدل‌های زبان بزرگ‌ها برای تولید داده‌های متنی ساده‌تر است.
* شبکه‌های مولد متخاصم فرایند آموزش پیچیده‌تر و مستلزم تنظیمات دقیق برای رسیدن به داده‌های باکیفیت است.

5- چالش‌های فنی

* مدل‌های زبان بزرگ نیازمند منابع محاسباتی قوی و داده‌های گسترده برای آموزش است.



* شبکه‌های مولد متخاصم مستعد مشکلاتی نظیر همگرایی ناپایدار یا تولید داده‌های بی‌کیفیت در صورت عدم تنظیم دقیق است.

مدل‌های زبان بزرگ به دلایل زیر برای تولید داده‌های مصنوعی مرتبط با حملات پایگاه‌داده غیررابطه‌ای نسبت به شبکه‌های مولد متخاصم برتری دارند:

1. توانایی درک ساختار پرس‌وجوهای پایگاه‌داده:

مدل‌های زبان بزرگ‌ها می‌توانند پرس‌وجوهای پیچیده و شبه کدهای مرتبط با حملات پایگاه‌داده غیررابطه‌ای را تولید کنند.

1. تولید داده‌های متنی با تنوع بالا:

برخلاف شبکه‌های مولد متخاصم، مدل‌های زبان بزرگ قادرند داده‌های متنی با ساختارها و الگوهای متنوع تولید کنند که برای امنیت سایبری حیاتی است.

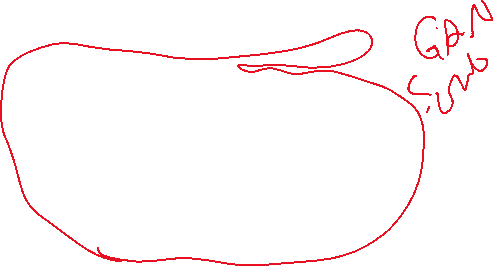
1. سادگی در تولید داده‌های ترکیبی:

مدل‌های زبان بزرگ‌ها می‌توانند داده‌هایی ترکیبی از متن و کد تولید کنند که برای شبیه‌سازی حملات سایبری و آموزش مدل‌ها ایده‌آل است.

چالش‌ها و محدودیت‌های مدل‌های زبان بزرگ و GAN

1-چالش‌های مدل‌های زبان بزرگ

* هزینه بالای محاسباتی برای آموزش.



* نیاز به تنظیمات دقیق برای تولید داده‌های متناسب با اهداف خاص.

2- چالش‌های GAN

* ناپایداری در فرایند آموزش.
* محدودیت در تولید داده‌های متنی.

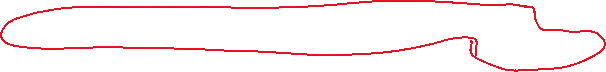
مدل‌های زبان بزرگ به دلیل قابلیت درک زبان طبیعی، توانایی تولید داده‌های متنی پیچیده و انعطاف‌پذیری بالا، جایگزینی مناسب و پیشرفته برای روش‌های سنتی مانند GAN در تولید داده‌های مصنوعی هستند[39]. در حوزه امنیت سایبری و به‌ویژه تولید داده‌های مرتبط با حملات پایگاه‌داده غیررابطه‌ای، استفاده از مدل‌های زبان بزرگ می‌تواند بهبود قابل‌توجهی در کیفیت و تنوع داده‌ها ایجاد کند. این ویژگی‌ها باعث می‌شود که مدل‌های زبان بزرگ‌ها ابزاری ارزشمند در تحقیقات امنیت سایبری و آموزش مدل‌های یادگیری ماشینی باشند.

## 2-4 پیشینه پژوهش

### 2-4-1 تزریق پایگاه‌داده غیررابطه‌ای

در ادامه چند پژوهش انجام شده مورد بررسی قرار میگیرند :

نام مقالهSECURE-D: Framework For Detecting and Preventing Attacks in SQL and NoSQL Databases [41]



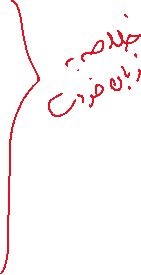
این مقاله یک چارچوب امنیتی تحت عنوان SECURE-D را معرفی می‌کند که برای شناسایی و جلوگیری از حملات به پایگاه داده‌های SQL و NoSQL طراحی شده است. این چارچوب از طریق یک پراکسی معکوس عمل می‌کند و تمامی درخواست‌های HTTP را قبل از رسیدن به سرور برنامه مورد بررسی قرار می‌دهد. این سیستم می‌تواند حملات SQL Injection و NoSQL Injection را شناسایی کند و در صورت شناسایی تهدید، از اجرای درخواست‌های مخرب جلوگیری نماید.  
بررسی‌ها نشان می‌دهند که SECURE-D در برابر حملات مختلف مانند تزریق SQL (مثل Tautologies، Union Queries، Piggy-backed Queries، و Stored Procedures) و همچنین حملات NoSQL (مثل NoSQL JavaScript Injection و Cross-Origin Violations) مؤثر است.



مهم‌ترین مشارکت‌های این مقاله عبارت‌اند از:



* ارائه یک چارچوب جدید برای محافظت هم‌زمان از پایگاه‌های داده SQL و NoSQL در برابر حملات تزریقی، که این امر در تحقیقات پیشین مورد بررسی قرار نگرفته بود.
* استفاده از پراکسی معکوس برای تجزیه و تحلیل درخواست‌های ورودی پیش از رسیدن به سرور، که باعث افزایش امنیت وب‌سایت‌ها در برابر تهدیدات اینترنتی می‌شود.
* شناسایی انواع مختلف حملات SQL و NoSQL و جلوگیری از اجرای درخواست‌های آلوده در سطح پایگاه داده.

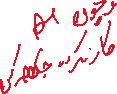


* ثبت و نمایش گزارشات حملات برای مدیران امنیتی، همراه با قابلیت مسدودسازی دستی آدرس‌های IP مخرب.
* مقایسه با روش‌های موجود مانند SEA WAF و DNIARS، که نشان می‌دهد SECURE-D طیف وسیع‌تری از حملات را شناسایی و مسدود می‌کند.

متدولوژی مقاله: محققان برای پیاده‌سازی و ارزیابی SECURE-D، سه ماژول اصلی را در معماری سیستم معرفی کرده‌اند:



1. ماژول تجزیه‌گر [[24]](#footnote-25)
   * درخواست‌های HTTP ورودی را تجزیه کرده و به URL، داده‌های GET و POST تبدیل می‌کند.
   * داده‌های ورودی را از نظر وجود الگوهای مخرب بررسی می‌کند.

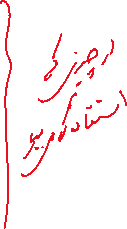


1. ماژول تشخیص حمله [[25]](#footnote-26)



* + شامل یک لیست از کلیدواژه‌های مخرب SQL و NoSQL مانند "SELECT"، "DROP"، "DELETE" برای SQL و "insertOne"، "create" برای NoSQL است.
  + در صورت شناسایی هر یک از این کلمات کلیدی در ورودی، درخواست مسدود می‌شود.
  + می‌تواند حملاتی مانند Tautologies، Logically Incorrect Queries، Union Queries، Piggy Backed Queries، و Stored Procedures در SQL و NoSQL Injection، NoSQL JavaScript Injection، و Cross-Origin Violations را شناسایی کند.

1. ماژول ثبت لاگ [[26]](#footnote-27)



* + تمامی حملات شناسایی‌شده را ثبت کرده و اطلاعاتی نظیر تعداد حملات، نوع حمله، و آدرس IP منبع حمله را نمایش می‌دهد.
  + مدیران امنیتی می‌توانند به‌صورت دستی IPهای مشکوک را مسدود کنند.

این سیستم در یک وب‌سایت آسیب‌پذیر تست شده است و نشان داده که می‌تواند به‌صورت لحظه‌ای حملات را شناسایی و متوقف کند.

محدودیت‌های مقاله

* محدودیت در شناسایی حملات پیچیده: اگر حمله‌کننده از تکنیک‌های Obfuscation (مبهم‌سازی) برای پنهان کردن کد مخرب خود استفاده کند، ممکن است سیستم SECURE-D در شناسایی آن‌ها دچار مشکل شود.
* نیاز به تنظیمات دستی برای برخی از حملات: این سیستم به‌طور خودکار همه حملات را مسدود نمی‌کند و در برخی موارد نیاز به دخالت دستی مدیران امنیتی برای به‌روزرسانی لیست کلیدواژه‌های مخرب وجود دارد.
* عدم بررسی سایر حملات امنیتی: تمرکز این مقاله صرفاً بر روی حملات تزریقی (SQL و NoSQL) است و حملات دیگری مانند XSS (Cross-Site Scripting) یا CSRF (Cross-Site Request Forgery) را پوشش نمی‌دهد.
* وابستگی به معماری پراکسی معکوس: این سیستم فقط در صورتی کارایی دارد که در مسیر تمامی درخواست‌های ورودی به سرور قرار گیرد، که ممکن است در برخی از زیرساخت‌های ابری یا سیستم‌های توزیع‌شده چالش‌برانگیز باشد.

نتیجه‌گیری مقاله

* SECURE-D یک چارچوب کارآمد برای شناسایی و جلوگیری از حملات پایگاه داده‌های SQL و NoSQL است.
* این سیستم درخواست‌های HTTP ورودی را پردازش کرده و حملات تزریقی را در زمان اجرا شناسایی و مسدود می‌کند.
* بررسی‌های عملی نشان دادند که این روش در مقایسه با روش‌های دیگر مانند SEA WAF و DNIARS عملکرد بهتری دارد و دامنه وسیع‌تری از حملات را پوشش می‌دهد.
* مزیت اصلی این چارچوب، ترکیب قابلیت‌های تشخیص حملات SQL و NoSQL در یک سیستم واحد است، که در روش‌های قبلی وجود نداشت.
* استفاده از ماژول‌های تجزیه و تحلیل، شناسایی و ثبت گزارشات باعث افزایش امنیت پایگاه داده‌های تحت وب می‌شود.

مقاله(Basic NoSQL Injection Analysis and Detection on MongoDB) [42]

این مقاله به بررسی حملات تزریق NoSQL در پایگاه داده MongoDB می‌پردازد. با توجه به افزایش استفاده از پایگاه‌های داده NoSQL به دلیل ویژگی‌هایی مانند مقیاس‌پذیری بالا، انعطاف‌پذیری و دسترسی سریع به داده‌ها، این پایگاه‌ها نیز مانند SQL در معرض حملات تزریقی قرار دارند.

این مطالعه به تحلیل آسیب‌پذیری‌های NoSQL و نحوه اجرای حملات تزریق در MongoDB می‌پردازد. برای مقابله با این تهدیدات، نویسندگان دو روش دفاعی را پیشنهاد می‌کنند:

1. اعتبارسنجی ورودی کاربران
2. استفاده از دستورات پارامتری [[27]](#footnote-28)

نتایج نشان می‌دهند که MongoDB، با وجود مزایای خود، همچنان در برابر حملات تزریقی آسیب‌پذیر است و باید اقدامات امنیتی لازم برای مقابله با این تهدیدات انجام شود.

مشارکت‌های کلیدی این تحقیق شامل موارد زیر است:

* تحلیل آسیب‌پذیری پایگاه داده‌های NoSQL در برابر حملات تزریقی، که نشان می‌دهد با وجود استفاده از JSON Queries، همچنان این پایگاه‌ها در معرض تهدید هستند.
* بررسی نحوه اجرای حملات تزریق در پایگاه داده MongoDB و شبیه‌سازی این حملات با استفاده از زبان‌های PHP و JavaScript.
* ارائه روش‌های مقابله با حملات تزریق NoSQL، شامل:
  1. محدود کردن ورودی کاربران با اعتبارسنجی مقادیر ورودی.
  2. استفاده از دستورات پارامتری به جای درج مستقیم داده‌های ورودی در دستورات پایگاه داده.
* ایجاد یک مدل برای تشخیص حملات تزریقی که به توسعه‌دهندگان کمک می‌کند تا آسیب‌پذیری‌های سیستم خود را شناسایی و برطرف کنند.

این مقاله بر حملات تزریق NoSQL در MongoDB تمرکز دارد و روش‌های اجرای این حملات و همچنین روش‌های مقابله با آن‌ها را مورد بررسی قرار داده است.

1. مراحل تحقیق و آزمایش:
2. تحلیل آسیب‌پذیری‌های MongoDB:
   * بررسی ساختار کلی پایگاه داده MongoDB و عملکرد آن.
   * تجزیه و تحلیل نحوه اجرای حملات تزریقی در محیط NoSQL.
3. اجرای حملات تزریقی در MongoDB:
   * استفاده از PHP و JavaScript برای ارسال کدهای مخرب به پایگاه داده.
   * بررسی آسیب‌پذیری‌های ورودی‌های کاربر، که ممکن است باعث اجرای کدهای مخرب شود.
   * نمایش نمونه‌ای از یک حمله تزریق که از طریق فرم ورودی انجام شده و داده‌های غیرمجاز را از پایگاه داده استخراج می‌کند.
4. پیشنهاد راهکارهای دفاعی:
   * استفاده از اعتبارسنجی ورودی: جلوگیری از ورود مقادیر غیرمجاز در فیلدهای ورودی.
   * استفاده از دستورات پارامتری: جدا کردن ورودی کاربر از دستورات پایگاه داده برای جلوگیری از اجرای کدهای مخرب.
   * استفاده از فیلترهای خاص برای حذف کاراکترهای مخرب در درخواست‌های کاربر.

با وجود ارائه راهکارهای مؤثر برای مقابله با حملات تزریقی NoSQL، این تحقیق دارای چند محدودیت است:

* محدود بودن مطالعه به MongoDB

این مقاله تنها MongoDB را بررسی کرده و روش‌های تزریق NoSQL در سایر پایگاه‌های داده مانند Cassandra، Redis یا CouchDB بررسی نشده‌اند.



* عدم بررسی تکنیک‌های پیچیده‌تر تزریق NoSQL

این مقاله بر حملات پایه‌ای NoSQL تمرکز دارد و روش‌های پیچیده‌تری مانند تزریق چند مرحله‌ای یا ترکیب حملات را پوشش نمی‌دهد.

* عدم ارائه آزمایش عملی در مقیاس وسیع:

این مطالعه بر یک محیط آزمایشی کوچک انجام شده و تأثیر حملات در سیستم‌های واقعی و پیچیده سازمانی مورد بررسی قرار نگرفته است.

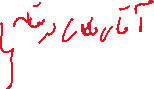
* عدم بررسی سایر تهدیدات امنیتی NoSQL

مقاله تنها به حملات تزریقی تمرکز دارد و سایر تهدیدات امنیتی مانند حملات XSS، CSRF و حملات مبتنی بر رمزنگاری را پوشش نمی‌دهد.

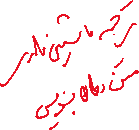
نتیجه‌گیری مقاله

* MongoDB و سایر پایگاه‌های داده NoSQL همچنان در معرض حملات تزریقی قرار دارند، اگرچه از SQL استفاده نمی‌کنند.
* روش‌های تزریق در NoSQL شباهت زیادی به SQL Injection دارد، اما به جای SQL، از JSON Query Language برای اجرا استفاده می‌شود.
* حملات تزریقی در NoSQL می‌توانند منجر به دسترسی غیرمجاز به اطلاعات، حذف داده‌ها و حتی تخریب کامل پایگاه داده شوند.
* برای افزایش امنیت، توسعه‌دهندگان باید از روش‌هایی مانند اعتبارسنجی ورودی، استفاده از دستورات پارامتری و فیلتر کردن مقادیر کاربر استفاده کنند.
* استفاده از یک مدل تشخیص حمله، می‌تواند به شناسایی فعالیت‌های مخرب کمک کند و امنیت پایگاه داده را افزایش دهد.

مقاله NoSQL Racket: A Testing Tool for Detecting NoSQL Injection Attacks in Web Applications [43]



این مقاله یک ابزار جدید به نام NoSQL Racket را معرفی می‌کند که برای شناسایی حملات تزریق NoSQL در برنامه‌های وب طراحی شده است. حملات تزریق NoSQL به این دلیل رخ می‌دهند که برنامه‌های وب، ورودی‌های کاربر را بدون اعتبارسنجی به پرس‌وجوهای پایگاه داده وارد می‌کنند. این مسئله به مهاجمان اجازه می‌دهد که کدهای مخرب خود را به پایگاه داده ارسال کنند و از این طریق به اطلاعات حساس دسترسی پیدا کنند یا آن‌ها را تغییر دهند.



ویژگی اصلی ابزار NoSQL Racket این است که ساختار کدهای پرس‌وجوی NoSQL را در دو سطح تحلیل استاتیک (در کد منبع) و تحلیل دینامیک (در زمان اجرا) بررسی و با یکدیگر مقایسه می‌کند. این ابزار روی چهار نوع پایگاه داده مختلف (MongoDB، Cassandra، CouchDB و Amazon DynamoDB) آزمایش شده و اثبات شده که توانایی شناسایی حملات تزریق NoSQL را دارد، در حالی که ابزارهای تست رایج مانند Netsparker، Vega و Skipfish در این زمینه ناکام بوده‌اند.

در مقاله‌ی NoSQL Racket گزارش شده است که ابزارهای رایج آزمون نفوذ و اسکن آسیب‌پذیری مانند Netsparker، Vega و Skipfish در سناریوهای ارزیابی آن پژوهش، توانایی تشخیص تزریق NoSQL را نداشته‌اند. با این حال، در پژوهش حاضر مقایسه‌ی مستقیم روش پیشنهادی با این ابزارها انجام نشده است؛ زیرا (1) این ابزارها عمدتاً در رده‌ی DAST/اسکنرهای عمومی وب قرار می‌گیرند و رفتار آن‌ها به تنظیمات محیط، مسیرهای خزیدن (Crawling)، و سازگاری با فناوری‌های سمت سرور وابسته است، در حالی‌که روش پیشنهادی این پایان‌نامه یک رویکرد مبتنی بر یادگیری ماشین و طبقه‌بندی ورودی/درخواست است و خروجی آن بر پایه‌ی تحلیل ویژگی‌های داده و الگوهای تزریق تولید می‌شود؛ (2) بازتولید دقیق شرایط آزمون مقاله‌ی NoSQL Racket (شامل پیاده‌سازی وب‌اپلیکیشن‌های هدف، پیکربندی چهار پایگاه‌داده و سناریوهای اجرایی) برای اجرای منصفانه‌ی این ابزارها، خارج از دامنه‌ی اجرایی و منابع این پژوهش بوده است.

با وجود این، به‌منظور جای‌دهی صحیح کار حاضر در ادبیات، از نتایج مقاله‌ی NoSQL Racket به‌عنوان شاهدی بر محدودیت ابزارهای اسکن رایج در تشخیص NoSQL Injection استفاده شده و روش پیشنهادی این پایان‌نامه به‌عنوان یک راهکار مکمل (نه الزاماً جایگزین مستقیم) برای افزایش قابلیت تشخیص، به‌ویژه در سطح داده/ورودی و الگوهای پرس‌وجو، معرفی می‌شود. همچنین، اجرای یک بنچمارک تجربی با Netsparker، Vega و Skipfish بر روی سناریوهای قابل بازتولید به‌عنوان پیشنهاد پژوهش آتی در نظر گرفته می‌شود.

مهم‌ترین مشارکت‌های این تحقیق شامل موارد زیر است:

1. ارائه یک ابزار جدید برای شناسایی حملات تزریق NoSQL که برخلاف ابزارهای رایج، بدون نیاز به زبان مشخص و مدل داده‌ای خاص، قادر به شناسایی حملات NoSQL در پایگاه‌های داده مختلف است.
2. بررسی و تحلیل روش‌های مختلف تزریق NoSQL در پایگاه‌های داده MongoDB، Cassandra، CouchDB و Amazon DynamoDB.
3. مقایسه ابزار NoSQL Racket با ابزارهای تست امنیت رایج (Netsparker، Vega و Skipfish) و نشان دادن برتری آن در تشخیص حملات تزریقی NoSQL.
4. ارائه یک مدل ترکیبی برای شناسایی حملات NoSQL که ترکیبی از تحلیل استاتیک کد و تحلیل دینامیک اجرای پرس‌وجو را به کار می‌گیرد.
5. اجرای آزمایش‌های عملکردی برای بررسی سرعت و کارایی NoSQL Racket در محیط‌هایی با بارگذاری‌های سنگین و کاربرهای هم‌زمان.

مراحل اصلی متدولوژی تحقیق و پیاده‌سازی:

1. تحلیل حملات تزریق NoSQL:
   * بررسی انواع حملات تزریق NoSQL که از طریق فرم‌های ورودی، کوکی‌ها و درخواست‌های HTTP انجام می‌شوند.
   * ارائه یک مثال عملی از نحوه‌ی تزریق NoSQL در MongoDB که به مهاجم اجازه می‌دهد بدون داشتن رمز عبور معتبر، وارد سیستم شود.
2. طراحی و پیاده‌سازی ابزار NoSQL Racket:
   * این ابزار از جدولی به نام "Driverstbl" استفاده می‌کند که شامل کلمات کلیدی، عملگرهای منطقی و عملگرهای رابطه‌ای NoSQL است.
   * کدهای پرس‌وجوی NoSQL در دو حالت (تحلیل استاتیک و دینامیک) مورد بررسی قرار می‌گیرند و ساختار آن‌ها با یکدیگر مقایسه می‌شود.
   * در صورتی که ساختار پرس‌وجو در زمان اجرا با ساختار اولیه مطابقت نداشته باشد، ابزار حمله را تشخیص داده و اجرای درخواست را متوقف می‌کند.
3. مقایسه با سایر ابزارهای امنیتی:
   * ابزار NoSQL Racket روی چهار پایگاه داده مختلف آزمایش شده است.
   * ابزارهای تست امنیتی معروف Netsparker، Vega و Skipfish به کار گرفته شدند اما نتوانستند حملات NoSQL را شناسایی کنند.
   * ابزار NoSQL Racket موفق شد حملات را در تمامی پایگاه‌های داده آزمایش‌شده تشخیص دهد.
4. آزمایش‌های عملکردی:
   * ابزار NoSQL Racket با استفاده از ابزار LoadComplete تحت بارگذاری سنگین قرار گرفت.
   * نتایج نشان دادند که این ابزار می‌تواند ۵۰ کاربر هم‌زمان در هر ثانیه را مدیریت کند.
   * میانگین زمان بارگذاری صفحات ۷۵ میلی‌ثانیه بود که نشان‌دهنده‌ی عملکرد مناسب ابزار است.

با وجود اثربخشی ابزار NoSQL Racket، این تحقیق دارای برخی محدودیت‌ها است:

* عدم پوشش تمام پایگاه‌های داده NoSQL

ابزار فقط روی MongoDB، Cassandra، CouchDB و Amazon DynamoDB آزمایش شده و سایر پایگاه‌های داده مانند Redis، Neo4j و Firebase بررسی نشده‌اند.

* محدودیت در شناسایی حملات پیچیده‌تر:

ابزار فقط حملات تزریق مستقیم را شناسایی می‌کند و ممکن است در برابر تزریقات پیشرفته‌تر مانند حملات چندمرحله‌ای (Multi-step Injection) یا ترکیبی با XSS و CSRF دچار مشکل شود.

* وابستگی به تنظیمات اولیه:

ابزار NoSQL Racket نیاز دارد که ابتدا جدول Driverstbl را به درستی تنظیم کند. در صورتی که یک پایگاه داده جدید معرفی شود، باید این جدول به‌روزرسانی شود.

* عدم بررسی سایر تهدیدات امنیتی:

مقاله تنها به حملات تزریق NoSQL پرداخته و سایر حملات امنیتی مانند سرقت نشست‌ها، XSS و حملات مبتنی بر رمزنگاری را پوشش نداده است.

نتیجه‌گیری مقاله

* NoSQL Injection یک تهدید جدی برای امنیت پایگاه‌های داده NoSQL است و بسیاری از برنامه‌های وب در برابر این نوع حملات آسیب‌پذیر هستند.
* ابزارهای تست امنیت رایج مانند Netsparker، Vega و Skipfish قادر به تشخیص حملات NoSQL نیستند.
* ابزار NoSQL Racket می‌تواند به طور دقیق حملات تزریقی NoSQL را شناسایی کند و مانع از اجرای پرس‌وجوهای مخرب شود.
* این ابزار بر روی چهار پایگاه داده‌ی NoSQL مختلف آزمایش شده و نتایج نشان داده‌اند که در تمامی موارد حملات را شناسایی کرده است.
* آزمایش‌های عملکردی نشان دادند که این ابزار تحت بارگذاری‌های سنگین عملکرد خوبی دارد و می‌تواند در محیط‌های واقعی مورد استفاده قرار گیرد.

مقاله (Analysis on Database Security Model Against NoSQL Injection)[44]

این مقاله به بررسی حملات تزریق NoSQL و راهکارهای امنیتی برای مقابله با این تهدیدات در پایگاه‌های داده NoSQL مانند MongoDB و Cassandra می‌پردازد. با توجه به افزایش استفاده از پایگاه‌های داده NoSQL به دلیل مقیاس‌پذیری بالا، انعطاف‌پذیری و دسترسی سریع به داده‌ها، این سیستم‌ها در برابر حملات مختلف از جمله تزریق NoSQL و حملات Cross-Site Request Forgery (CSRF) آسیب‌پذیر شده‌اند.

مدل امنیتی پیشنهادی در این مقاله شامل استفاده از الگوی امنیتی متمرکز بر داده‌ها (Data-Centric Security Model) است که اطلاعات را قبل از ذخیره در پایگاه داده رمزگذاری می‌کند. علاوه بر این، احراز هویت کاربران در شبکه‌های غیرقابل اعتماد از طریق پروتکل Kerberos انجام می‌شود.

* نکات کلیدی مقاله:
* بررسی حملات تزریق NoSQL و نحوه سوءاستفاده مهاجمان از آن‌ها.
* بررسی آسیب‌پذیری‌های MongoDB و Cassandra در برابر حملات تزریقی.
* معرفی پروتکل امنیتی Kerberos برای احراز هویت کاربران و افزایش امنیت پایگاه داده.
* بررسی روش‌های جلوگیری از حملات، از جمله اعتبارسنجی ورودی کاربران و رمزگذاری داده‌ها.
* آزمایش حملات تزریق NoSQL با استفاده از JavaScript و PHP.

مهم‌ترین مشارکت‌های این تحقیق شامل موارد زیر است:

1. تحلیل حملات تزریق NoSQL و ارائه الگویی برای طبقه‌بندی آسیب‌پذیری‌های امنیتی در این پایگاه‌های داده.
2. بررسی و مقایسه امنیت MongoDB و Cassandra در برابر حملات تزریقی، از جمله تحلیل مکانیزم‌های احراز هویت و رمزگذاری داده‌ها.
3. ارائه یک مدل امنیتی مبتنی بر داده[[28]](#footnote-29) که داده‌ها را قبل از ذخیره شدن در پایگاه داده رمزگذاری می‌کند.
4. استفاده از پروتکل Kerberos برای احراز هویت کاربران در پایگاه داده‌های NoSQL، که می‌تواند به جلوگیری از دسترسی‌های غیرمجاز کمک کند.
5. ارائه راهکارهای امنیتی برای مقابله با حملات تزریق NoSQL، از جمله:
   * اعتبارسنجی ورودی کاربران
   * استفاده از احراز هویت
   * فعال‌سازی ثبت گزارش‌های امنیتی
   * رمزگذاری داده‌ها قبل از ذخیره‌سازی در پایگاه داده

مراحل اصلی تحقیق و پیاده‌سازی:

* تحلیل حملات تزریق NoSQL:
  + بررسی نحوه عملکرد پایگاه‌های داده NoSQL مانند MongoDB و Cassandra.
  + شناسایی نقاط ضعف امنیتی و نحوه اجرای حملات تزریق NoSQL در این سیستم‌ها.
  + ارائه مثال‌هایی از تزریق NoSQL در محیط MongoDB و Node.js.
* بررسی آسیب‌پذیری‌های NoSQL در برابر حملات تزریقی:
  + تحلیل نمونه‌هایی از حملات تزریق در MongoDB که در آن‌ها از کوئری‌های مخرب استفاده شده است.
  + بررسی تأثیر حملات تزریقی در Cassandra و راه‌های جلوگیری از آن.
* ارائه راهکارهای امنیتی برای مقابله با این حملات:
  + معرفی پروتکل Kerberos برای احراز هویت و جلوگیری از دسترسی غیرمجاز.
  + بررسی روش‌های رمزگذاری داده‌ها قبل از ذخیره در پایگاه داده.
  + بررسی روش‌های جلوگیری از اجرای کوئری‌های مخرب در پایگاه داده.
* مقایسه امنیت MongoDB و Cassandra:
  + بررسی مکانیزم‌های امنیتی هر دو پایگاه داده.
  + مقایسه وضعیت احراز هویت، رمزگذاری و محافظت در برابر حملات تزریقی در MongoDB و Cassandra.
* ارائه یک مدل امنیتی ترکیبی:
  + استفاده از Kerberos برای احراز هویت و مدیریت نشست‌های کاربری.
  + رمزگذاری داده‌ها قبل از ذخیره‌سازی برای جلوگیری از حملات تزریقی.
  + استفاده از ثبت گزارش‌های امنیتی (Auditing) برای پایش فعالیت‌های کاربران.

با وجود ارائه راهکارهای امنیتی مؤثر، این تحقیق دارای برخی محدودیت‌ها است:

* تمرکز بر MongoDB و Cassandra

این مطالعه فقط دو پایگاه داده NoSQL را بررسی کرده و سایر پایگاه‌های داده مانند CouchDB، Firebase و Redis پوشش داده نشده‌اند.

* عدم بررسی حملات ترکیبی

تحقیق تنها به حملات تزریق NoSQL پرداخته و سایر تهدیدات مانند XSS، CSRF و حملات مهندسی اجتماعی بررسی نشده‌اند.

* وابستگی به Kerberos

پیشنهاد اصلی مقاله استفاده از Kerberos برای احراز هویت کاربران است، اما این روش در برخی سیستم‌ها به دلیل پیچیدگی پیاده‌سازی دشواری‌هایی دارد.

* عدم آزمایش در محیط‌های واقعی و بزرگ:

آزمایش‌های امنیتی انجام‌شده در یک محیط آزمایشی کوچک صورت گرفته‌اند و تأثیر روش‌های پیشنهادی در سیستم‌های مقیاس‌پذیر سازمانی بررسی نشده است.

نتیجه‌گیری مقاله

* حملات تزریق NoSQL یک تهدید جدی برای امنیت پایگاه‌های داده NoSQL هستند، زیرا این سیستم‌ها به دلیل نداشتن ساختار ثابت، در برابر کوئری‌های مخرب آسیب‌پذیرتر هستند.
* MongoDB و Cassandra به دلیل انعطاف‌پذیری بالا، در برابر حملات تزریقی آسیب‌پذیر هستند.
* پروتکل Kerberos یک راهکار مؤثر برای احراز هویت کاربران و جلوگیری از دسترسی غیرمجاز به پایگاه داده است.
* رمزگذاری داده‌ها قبل از ذخیره در پایگاه داده، یک لایه امنیتی اضافی ایجاد می‌کند و مانع از افشای اطلاعات حساس در صورت وقوع حملات می‌شود.
* فعال‌سازی ثبت گزارش‌های امنیتی (Auditing) می‌تواند به شناسایی حملات و فعالیت‌های مخرب کمک کند.
* پایگاه‌های داده NoSQL نیاز به راهکارهای امنیتی قوی‌تری دارند تا بتوانند در برابر حملات پیچیده‌ای مانند تزریق NoSQL مقاومت کنند.

شبکه متخاصم مولد متشکل از مولدها و تمايز کنندگاني است که عليه يکديگر بازي مي‌کنند. داده‌هاي ورودي با نمونه‌گيري تصادفي در فضاي پنهان به‌دست‌آمده و به شبکه مولد منتقل مي‌شود. شبکه مولد، يک رويکرد مبتني بر سنتز شبکه، نمونه‌هاي متنوع‌تري را در مقايسه با تكنيك‌هاي سنتي افزايش داده توليد مي‌کند، اگرچه فرايند آن پيچيده‌تر است [12].

اما مجموعه‌داده‌ها با برچسب‌هاي دسته‌بندي خاص به‌سختي به دست مي‌آيند، در اين مقاله بر تقويت داده‌ها براي يادگيري نمايش ويژگي‌هاي مشابه از داده‌هاي اصلي براي بهبود دقت مدل‌هاي طبقه‌بندي تمرکز كرده و شبکه‌هاي متخاصم مولد کانولوشني عميق همراه با الگوريتم‌هاي ژن‌شناختي در زمينه حملات آسيب‌پذيري وب باهدف حل مشکل تعداد ناکافي نمونه‌هاي تزريق پرس‏وجو اعمال شده است. همچنين انتظار مي‌رود اين روش براي توليد نمونه براي انواع ديگر حملات آسيب‌پذيري نيز اعمال شود .[12]

همچنين در تحقيق آقاي لودانگ‏ژو و همكاران، با دانش به اينكه، انتخاب مجموعه‏داده نمونه، عامل تعيين‌کننده‌اي است و پاسخ به سؤال: آيا الگوريتم‌هاي هوش مصنوعي مي‌توانند در آن به نتايج خوبي دست يابند؟ به سراغ توليد مجموعه‏داده تزريق با مدل مبتني بر GAN اقدام شد [12].

در اين مقاله، نمونه‌هاي حمله توسط مجموعه‌داده‌هاي BATADAL و مجموعه‌داده‌هاي تجاري يک انبار نفت توليد مي‌شوند و داده‌ها از طريق الگوريتم 100 برابر گسترش مي‌يابند [13].

در تحقيق آقاي ون‏ژو و محققين ديگر، مسئله کمبود داده‌هاي تجاري در سامانه واپايش صنعتي موردمطالعه قرار گرفت و الگوريتم توليد نمونه حمله ارائه شده، ابتدا، نتايج گروه‌بندي همبستگي به‌وسيله توزيع وزن و عضويت به دست مي‌آيد و سپس نتايج گروه‌بندي ارتباط قوي براي حمله به‌دست‌آوردن نمونه‌هاي حمله مورد حمله قرار مي‌گيرد. در نهايت، از GAN براي گسترش نمونه استفاده مي‌شود. اين مقاله از مجموعه‌داده‌هاي باز و يک مجموعه‌داده از نفتکش ساختمان نمونه‌هاي حمله، درجه تطابق و سازگاري را توليد مي‌کند [13].

توليد مجموعه‌داده مبتني بر GAN

آقاي رامشوار و ساير محققين در سال 2023، اذعان داشتند كه در خصوص پایگاه‌داده رابطه‌ای i كارهاي بسياري انجام شده است. اما آنچه در اين پژوهش به طور ويژه موردبحث است حملات تزريق پایگاه‌داده غیررابطه‌ای است كه مقاله فوق به توليد 400 نمونه داده آن هم صرفاً براي MongoDB پرداخته است. ازاين‌رو يک مجموعه‌داده جامع شامل 400 دستور تزريق پایگاه‌داده غیررابطه‌ای جمع‌آوري‌شده است. اين دستورات به دودسته تقسيم مي‌شوند: 221 فرمان مخرب و 179 فرمان خوش‌خيم. مجموعه‌داده به‌دقت با ترکيب دستورات نوشته شده به‌صورت دستي و دستورات به‌دست‌آمده از طريق وب از منابع معتبر مانند GitHub و ساير وبگاه‌هاي مشخص شده که در بخش منابع اين مقاله توضيح داده شده است، تنظيم شده است. مجموعه‌داده‌هاي جمع‌آوري‌شده به‌عنوان منبعي ارزشمند براي مطالعه و تحليل آسيب‌پذيري‌هاي تزريق پایگاه‌داده غیررابطه‌ای، ارائه بينش‌هايي در مورد تهديدات امنيتي بالقوه و کمک به توسعه سازوکارهاي حفاظتي قوي در برابر چندين حمله عمل مي‌کند. مجموعه‌داده شامل ترکيبي از دستورات پيچيده و ساده است که هنوز براي يادگيري ماشيني و تجزيه‌وتحليل داده‌ها، به‌ويژه براي علاقه‌مندان به امنيت، مناسب است [10].

توليد مجموعه‌داده براي حملات تزريق پایگاه‌داده غیررابطه‌ای

آقاي امين اقباليان و همكاران در سال 1401 در پژوهشي به نام توليد مجموعه‌داده استاندارد جهت تشخيص آسيب‌پذيري‌هايي منشأ پایگاه‌داده رابطه‌ای I براي استفاده در راهكارهاي مبتني بر يادگيري ماشيني مجموعه‌دادة براي اين منظور توليد كردند. اين مقاله براي نيل به اين منظور رفتار يك هكر واقعي و اطلاعاتي كه يك هكر جمع‏آوري می‌کند ملاك عمل براي توليد اين مجموعه‏داده قرار گرفته است. پس از ارزيابي، ميزان دقت كلي راهكار پيشنهادي توليد اين مجموعه‏داده براي تشخيص آسيب‏پذيري برنامه تحت آزمون %٩٨ درصد است.

توليد مجموعه‌داده براي پایگاه‌داده رابطه‌ای

مجموعه‌داده‌هاي بزرگ حاوي مواد خام براي درک دنياي اطراف ما هستند. تا کنون، محاسبات یادگیری ماشین در مقياس بزرگ در ديتاسنترهاي بزرگ حاوي انبوهي از GPUها که در اصل براي تسريع گرافيک طراحي شده بودند، انجام شده است. ما در حال گذار به عصري هستيم که در آن ديتاسنترها پر از رايانه‌هايي مي‌شوند که صرفاً براي محاسبات یادگیری ماشین طراحي شده‌اند. آن‏ها نه‌تنها از توليدکنندگان GPU، بلکه از استارتاپ‌ها، از خطوط توليد جديد عرضه‌کنندگان ريزپردازنده‌هاي سنتي، و از شرکت‌هاي اينترنتي معتبرتر که قبلاً پردازنده‌هاي خود را ساخته بودند، ارائه مي‌شوند.

منابع محاسباتي مورداستفاده براي تجزيه‌وتحليل آن‏ها.

در مجموعه‌داده‌هاي موجود

آقاي يانگ در سال 2019 به يك انقلاب در حوزه رايانه اشاره دارند كه با مجموعه‏داده ImageNet رخ داد، براي اين تعريف جهت‏گيري‏هايي نيز تعريف كرده‏اند كه انقلاب فعلي یادگیری ماشین به دو نوع مقياس نياز دارد:

تأثیر توليد مجموعه‌داده‏ها

در حوزه حملات تزريق پایگاه‌داده غیررابطه‌ای نيز مي‏توان به تلاش آقاي لاندايت و همكاران در سال 2024 اشاره داشت، در اين مقاله، نتايج يک بررسي عميق از خطرات مربوط به تزريق در پايگاه‌داده گراف Neo4j و اکوسيستم گسترده‌تر آن را ارائه شده كه بر اساس دو معيار تقسيم شده است. اول، بررسي اجراي توزيع شده پرس‌وجوهاي عاملي، از اتصال‌دهنده‌هاي زبان ويژه كاربران، تا ارتباطات (پروتکل Bolt) و اجرا در Neo4j (در طرح‌هاي پرس‌وجو)، و دوم شناسايي مشکلات تزريق باقيمانده در مواردي که پرس‌وجوهاي ثابت و عاملي شده کافي نيستند. اين مطالعه شامل بررسي جريان داده‌هاي کد محور با پايه کد Neo4j است و با مجموعه آزمايشي موارد آزمون تزريق تکميل مي‌شود.

آقاي داس و همكاران در سال 2019 نيز در خصوص آسيب‌پذيري از تزريق پرس‌وجو كار كردند، آن‏ها بر اين باور هستند كه هر زمان که برنامه وب دستورات پایگاه‌داده رابطه‌ای پويا را اجرا مي‌کند ممکن است تحت حمله تزريق پرس‏وجو قرار گيرد. براي ارزيابي شيوه‌هاي موجود تشخيص آن، ما دو سناريو امنيتي مختلف را براي احراز هويت برنامه‌هاي وب در نظر مي‌گيريم که پرس‌وجوي پایگاه‌داده رابطه‌ای پويا را با داده‌هاي ورودي کاربر ايجاد مي‌کند. براين‌اساس، دو مجموعه‌داده مختلف را با درنظرگرفتن تمام آسيب‌پذيري‌هاي احتمالي در فواصل زماني اجرا توليد نموده. رويکرد پيشنهادي مبتني بر فاصله ويرايش براي طبقه‌بندي يک پرس‌وجوي پایگاه‌داده رابطه‌ای پويا به‌عنوان عادي يا مخرب با استفاده از نمايه وب آماده‌شده با پرس‌وجوهاي پایگاه‌داده رابطه‌ای پويا در طول مرحله آموزش ارائه مي‌شود. مجموعه‌داده را با استفاده از رويکرد پيشنهادي و برخي از رويکردهاي طبقه‌بندي نظارت شده معروف، ارزيابي نموده كه در نتيجه روش آن‏ها در تشخيص حمله تزريق پرس‏وجو تحت هر دو سناريو امنيت احراز هويت مؤثرتر است .

در پايان‌نامه آقاي رومياني در سال 1402 با استفاده از مدل‏هاي يادگيري ماشين در خصوص نحوه پيشگيري از حملات تزريق راه‌حل‌هايي ارائه شده است. در جلسات دفاع نيز نسبت به کم‌بودن حجم مجموعه‌داده‏ و دشواري زيادي كه ايشان براي به‌دست‌آوردن مجموعه‏داده داشتند، اشاره شد .

تشخيص حملات تزريق پایگاه‌داده غیررابطه‌ای

با بررسي در مطالعات انجام شده متناسب با اين پژوهش به دسته‌بندي از مقالات اصلي خواهيم رسيد:

## 1-4 سابقه تحقيقات و مطالعات انجام‌گرفته

**[منتقل‌شده از فصل ۱]**

جدول 2- 2: مقایسه مقالات [42], [43], [44]

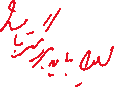
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| رفرنس | تمرکز/هدف | روش | خروجی کلیدی | محدودیت/شکاف | ارتباط با پژوهش حاضر |
| [42] | تحلیل و کشف NoSQLi در MongoDB | تست نفوذ و سناریوهای حمله (PHP/JS) + پیشنهاد دفاع | تأیید آسیب‌پذیری MongoDB؛ توصیه به Validation و Parameterized Queries | تک‌پایگاه‌داده (فقط MongoDB)؛ بدون ارزیابی مدل‌های داده‌محور | نیاز به روش قابل تعمیم و داده‌محور برای تشخیص |
| [43] | توسعه ابزار تشخیص NoSQLi در وب | ابزار NoSQL Racket (Static + Dynamic Analysis) | تشخیص موفق NoSQLi؛ ناکامی برخی اسکنرهای عمومی (Netsparker/Vega/Skipfish) | محدود به 4 DB؛ تمرکز ابزارمحور (نه مدل ML) | الهام برای سنجش/بنچمارک؛ تأکید بر خلأ دیتاست/مدل ML |
| [44] | مدل امنیتی DB در برابر NoSQLi | مقایسه MongoDB/Cassandra + احراز هویت Kerberos | Kerberos و Encryption به‌عنوان لایه دفاعی مؤثر | محدود به 2 DB؛ تمرکز روی Auth نه تشخیص حمله؛ عدم پوشش تهدیدات دیگر | مکمل امنیتی؛ اما خلأ تشخیص مبتنی بر داده باقی است |



جدول 2- 3: مقایسه مقالات [42], [43], [44] از دیدگاه محتلف

|  |  |  |  |
| --- | --- | --- | --- |
| **معیار مقایسه** | **[42]** | **[43]** | **[44]** |
| نوع پژوهش | تحلیلی – آزمایشی | ابزاری – تشخیصی | معماری – امنیتی |
| تمرکز اصلی | نحوه وقوع حمله | شناسایی حمله | جلوگیری معماری‌محور |
| سطح دفاع | سطح Query | سطح ابزار تست | سطح سیستم و احراز هویت |
| کاربرد عملی | متوسط | بسیار بالا | بالا |
| مناسب برای | فهم حمله | تست نفوذ | طراحی امن سیستم |

### 2-4-2 رویکرد یادگیری ماشین



شناسایی و پیشگیری از تزریق پایگاه‌داده غیررابطه‌ای با رویکرد یادگیری ماشین

یک مدل یادگیری ماشین برای شناسایی تزریق پایگاه‌داده غیررابطه‌ای با استفاده از یادگیری نظارت شده مبتنی بر ویژگی‌ها توسعه داده شد[45].

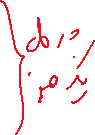
در این روش، آن‌ها مجموعه داده‌ای از پرس‌وجوهای سالم و مخرب مربوط به MongoDB ایجاد کردند، زیرا مجموعه داده‌ای برچسب‌گذاری شده برای تزریق پایگاه‌داده غیررابطه‌ای موجود نبود. آن‌ها از منابع مختلفی مانند OWASP، راهنمای MongoDB و غیره استفاده کردند.

برای انتخاب ویژگی‌های مدل، از ابزار WEKA با روش ClassifierSubsetEval و الگوریتم‌های طبقه‌بندی مانند J48 درخت تصمیم) و IBK (k-نزدیک‌ترین همسایگی و جستجوی گام‌به‌گام حریصانه با حذف معکوس استفاده کردند. این روش‌ها به شناسایی 10 ویژگی برتر بر اساس اطلاعات به‌دست‌آمده[[29]](#footnote-30) و همبستگی[[30]](#footnote-31) کمک کردند[24], [46]. این ویژگی‌های انتخاب‌شده در جدول 2-1 ذکر شده‌اند.

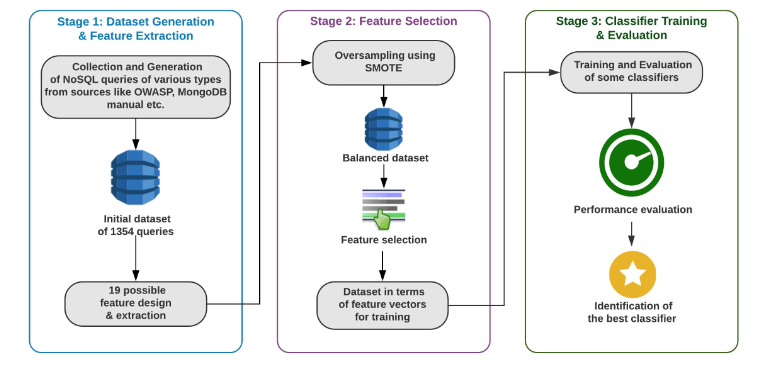
در مجموعه‌داده آن‌ها، انواع مختلفی از حملات مورد بررسی قرار گرفتند، از جمله:

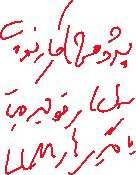
* تزریق آرایه PHP
* تزریق OR در پایگاه‌داده غیررابطه‌ای
* تزریق مبتنی بر جاوا اسکریپت
* پرس‌وجوهای زنجیره‌ای[[31]](#footnote-32)

جدول 2- 4: ویژگی‌های انتخاب‌شده



|  |  |
| --- | --- |
| رتبه | ویژگی |
| 1 | **مقایسه محتویات (Contains Comparison):** بررسی وجود مقادیر مشابه در پرس‌وجو. |
| 2 | **پرس‌وجوی جدید (New Query):** وجود یک پرس‌وجوی جدید در داده‌ها. |
| 3 | **شامل رشته خالی (Contains Empty String):** بررسی حضور رشته‌های خالی در پرس‌وجو. |
| 4 | **شامل عبارت نابرابر (Contains Not Equal):** وجود عملگر "نابرابر" در پرس‌وجو. |
| 5 | **شامل بار مخرب (Contains Payload):** بررسی وجود داده‌های مخرب در پرس‌وجو. |
| 6 | **حضور عبارت بازگشتی (Presence of Return):** بررسی حضور عبارت‌های بازگشتی در پرس‌وجو. |
| 7 | **عبارت همیشه صحیح (Always True Expression):** شناسایی عبارات منطقی که همیشه نتیجه **true** دارند. |
| 8 | **عملیات ارزیابی پرس‌وجو (Evaluation Query Operation):** ارزیابی نحوه عملکرد پرس‌وجوها. |
| 9 | **شامل عملگر منطقی (Contains Logical Operator):** بررسی وجود عملگرهای منطقی مانند **AND** و **OR**. |
| 10 | **عملیات پرس‌وجوی عنصری (Element Query Operation):** انجام عملیات پرس‌وجو بر روی عناصر مشخص‌شده. |





شکل2- 2: مدل مفهومی یادگیری ماشین[46]

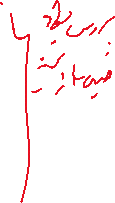
با استفاده از 10 ویژگی ذکر شده در جدول 1-2 در این مطالعه، آن‌ها بر روی دسته‌بندی دودویی کار کردند که شامل دو کلاس سالم و تزریق بود. برای این کار، از طبقه‌بندی‌کننده‌های یادگیری ماشین مانند الگوریتم مبتنی بر درخت تصمیم ID3، شبکه عصبی مصنوعی با پس‌انتشار خطا، جنگل تصادفی، AdaBoost، نزدیک‌ترین همسایه k، ماشین‌های بردار پشتیبان و XGBoost استفاده کردند.

آن‌ها از اعتبارسنجی متقابل 10 قسمتی [[32]](#footnote-33)برای ارزیابی عملکرد طبقه‌بندی‌کننده‌ها استفاده کردند [25]. سپس از یک ابزار تولید تزریق پایگاه‌داده غیررابطه‌ای به نام پایگاه‌داده غیررابطه‌ای Map برای ایجاد مجموعه‌داده تست استفاده کردند [26].

پایگاه‌داده غیررابطه‌ای Map برای تولید مجموعه‌داده اصلی آن‌ها استفاده نشده بود. این مجموعه‌داده تست پر از تزریق پایگاه‌داده غیررابطه‌ای بود و آن‌ها مدل خود را با این مجموعه‌داده آزمایش کردند. روش آن‌ها در مقایسه باSqreen عملکرد بهتری داشت و نرخ شناسایی به طور متوسط 36.25٪ بیشتر از Sqreen بود.

### 2-4-3 رویکرد غیر یادگیری ماشین

برخی از روش‌های غیر یادگیری ماشین برای شناسایی و پیشگیری از حملات تزریق پایگاه‌داده غیررابطه‌ای عبارت‌اند از:



1. خودکارساز: در این رویکرد، یک مدل شناسایی مبتنی بر خودکارساز برای تزریق پایگاه‌داده غیررابطه‌ای ایجاد می‌شود [39]. این مدل عمدتاً بر روی حملات تزریق زمان‌بندی‌شده و کور متمرکز است.

در تزریق مبتنی بر زمان، هکر تلاش می‌کند یک تابع جاوا اسکریپت را همراه با یک نشانه معتبر پایگاه‌داده غیررابطه‌ای اضافه کند. با این کار، مهاجم پایگاه‌داده را در حالت تعلیق قرار می‌دهد. برای مثال، پرس‌وجوی زیر، پایگاه‌داده MongoDB را برای 5 ثانیه در حالت تعلیق قرار می‌دهد اگر ورودی آن "John" باشد:

John’where: ‘function(){ sleep(5000); return this.name ==\John’" }

این رویکرد زمان و هزینه تولید داده‌های مصنوعی را به طور قابل‌توجهی کاهش داده و نیاز به داده‌های واقعی گسترده را به حداقل می‌رساند.

تسهیل فرایند تولید داده‌ها:

استفاده از داده‌های مصنوعی تولیدشده با مدل‌های زبان بزرگ، موجب بهبود فرایند آموزش مدل‌های یادگیری ماشینی و افزایش دقت آن‌ها در تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای می‌شود.

افزایش دقت مدل‌های یادگیری ماشینی:

مدل‌های زبان بزرگ‌داده‌هایی تولید می‌کنند که شباهت زیادی به داده‌های واقعی دارند و می‌توانند برای شبیه‌سازی حملات استفاده شوند.

تولید داده‌های مصنوعی متناسب با حملات پایگاه‌داده غیررابطه‌ای:

دستاوردهای کاربردی

پشتیبانی بهتر از تولید داده‌های خاص نظیر پرس‌وجوهای پایگاه‌داده.

انعطاف‌پذیری بالا در شبیه‌سازی سناریوهای پیچیده.

کاهش پیچیدگی‌های پیاده‌سازی و تنظیم.

تولید داده‌های متنی و ساختاریافته بادقت و کیفیت بالا.

مزایای استفاده از مدل‌های زبان بزرگ در برابر شبکه‌های مولد متخاصم

داده‌های تولیدشده به‌عنوان بخشی از مجموعه‌داده نهایی برای آموزش مدل‌های یادگیری ماشینی استفاده می‌شوند.

استفاده از داده‌های تولیدشده در مدل‌های یادگیری ماشینی:

داده‌های تولیدشده با استفاده از معیارهای مختلف نظیر شباهت به داده‌های واقعی، تنوع، و پوشش ویژگی‌های مهم ارزیابی می‌شوند.

ارزیابی داده‌های تولیدشده:

مدل مدل‌های زبان بزرگ با استفاده از داده‌های اولیه و برچسب‌گذاری شده، برای تولید داده‌هایی متناسب با سناریوهای حملات تزریق پایگاه‌داده غیررابطه‌ای آموزش داده می‌شود.

آموزش و تنظیم مدل مدل‌های زبان بزرگ:

ابتدا مجموعه ویژگی‌های مرتبط با حملات تزریق پایگاه‌داده غیررابطه‌ای، نظیر نوع پایگاه‌داده، نوع پرس‌وجو و رفتار مخرب، شناسایی و تعریف می‌شوند.

بررسی و تعریف ویژگی‌های موردنیاز:

در این پژوهش از مدل‌های زبان بزرگ برای تولید داده‌های مصنوعی استفاده شده است. فرایند تولید داده‌ها شامل مراحل زیر است:

رویکرد پیشنهادی برای تولید داده‌ها

مهندسی ویژگی دستی و هدفمند نقش تعیین‌کننده‌تری نسبت به تقویت داده در بهبود عملکرد مدل‌ها داشت

اختلاف ناچیز دیتاست‌های مصنوعی تولیدشده با دیتاست اولیه نشان می‌دهد که این داده‌های مصنوعی می‌توانند در آینده برای کاربردهای یادگیری عمیق (Deep Learning) مورد استفاده قرار گیرند

دیتاست اولیه با مهندسی دقیق ویژگی‌ها از کیفیت مناسبی برخوردار است

نتایج نشان داد که:

سپس رویکرد به سمت **مهندسی خودکار ویژگی‌ها** تغییر کرد و مجدداً مدل‌های مختلف ارزیابی شدند. نتایج نشان داد که دیتاست اولیه صرفاً با مهندسی ویژگی‌های دستی به نتایج بهتری دست یافت. تولید دیتاست با روش‌های مختلف نشان داد که این روش‌ها نتوانستند دیتاست بهتری نسبت به دیتاست اولیه ارائه دهند.

در مرحله بعد، **مهندسی ویژگی** انجام شد تا ویژگی‌های مناسب‌تری برای حملات NoSQL استخراج گردد. پس از این مرحله، تلاش شد با **تقویت داده (Augmentation)** و تولید نمونه‌های مصنوعی، عملکرد مدل‌ها بهبود یابد؛ اما نتایج قابل‌توجهی حاصل نشد.

این پژوهش با بهره‌گیری از ویژگی‌های استخراج‌شده توسط رومیانی در زمینه حملات تزریق SQL آغاز شد. هدف اولیه، بررسی کارآمدی این ویژگی‌ها برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای (NoSQL) نظیر MongoDB بود. نتایج اولیه نشان داد که مدل‌های مبتنی بر این ویژگی‌ها برای پایگاه‌داده‌های غیررابطه‌ای ناکارآمد هستند.

با پیشرفت فناوری هوش مصنوعی و یادگیری ماشین، استفاده از این فناوری‌ها برای شناسایی حملات تزریقی در پایگاه‌های داده به طور چشمگیری افزایش یافته است. یکی از چالش‌های اساسی در این حوزه، فراهم‌سازی مجموعه‌داده‌های کافی و مناسب برای آموزش و ارزیابی مدل‌های یادگیری ماشینی است.

### 1-2-5 بررسي و تشريح مدل

**[منتقل‌شده از فصل ۱]**

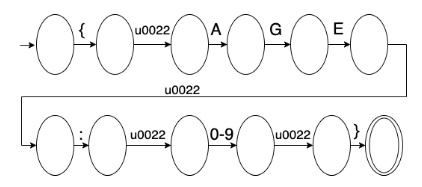
در تزریق کور مبتنی بر بولی (Blind Based Boolean Injection)، مهاجمان از قابلیت‌های MongoDB استفاده می‌کنند تا به لیستی از مجموعه‌ها (Collections)، تعداد مجموعه‌ها و اطلاعات مشابه دسترسی پیدا کنند.

return (db.getCollectionNames().length == 1);

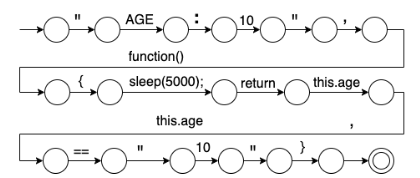
return(tojsononeline (db.collectionname.find() [0]).length == 1);

return(db.getCollectionNames() [0] [0] == ‘a’);

برای شناسایی و پیشگیری از تزریق پایگاه‌داده غیررابطه‌ای، ابتدا نقاطی در کد منبع که از آن‌ها فراخوانی پایگاه‌داده انجام می‌شود، شناسایی می‌کنند. به طور معمول، در MongoDB این نقاط شامل توابع find()، insert()، remove() و update() هستند. سپس برای پرس‌وجوهای معتبر یا ایمن که می‌توانند از این نقاط حساس (Hotspots) تولید شوند، مدل‌های NFA ماشین‌های حالت محدود غیرقطعی) ایجاد می‌کنند و از کتابخانه JSA (تحلیلگر رشته جاوا برای این کار استفاده می‌کنند[47].



شکل2- 3: نمونه کوئری در منگو دیبی



شکل2- 4: مدل نمونه برای ورودی کاربر مورد نظر حمله[47]

سپس آن‌ها پرس‌وجوهای دینامیکی را هنگام دریافت ورودی از کاربر ایجاد می‌کنند. اگر این پرس‌وجوها با مدل‌های NFA ازپیش‌ساخته‌شده مطابقت داشته باشند، پرس‌وجو به‌عنوان معتبر یا ایمن برای سیستم آن‌ها شناخته می‌شود. تنها پرس‌وجوهای معتبر مجاز هستند به پایگاه‌داده ارسال شوند. در شکل 5 می‌توانیم ببینیم که یک ورودی نامعتبر ارسال شده است. اگر شکل 4 و شکل 5 را مقایسه کنیم، متوجه خواهیم شد که مدل‌ها مطابقت ندارند. بنابراین، پرس‌وجوی شکل 4 اجازه ارسال به پایگاه‌داده را نخواهد داشت. به این ترتیب، راه‌حل آن‌ها برای شناسایی و پیشگیری از تزریق پایگاه‌داده غیررابطه‌ای مبتنی بر خودکارساز کار می‌کند.

1. اعتبارسنجی ورودی کاربر: توسعه‌دهندگان در هنگام ساخت سیستم برای شناسایی و پیشگیری از حملات تزریق پایگاه‌داده رابطه‌ای، اقدامات احتیاطی مختلفی انجام می‌دهند. به‌عنوان‌مثال، در MongoDB می‌توان با افزودن کد زیر، اندازه فیلدهای ورودی را محدود کرد[29] :

|  |
| --- |
| onkeypress = return"event.keyCode>= 48&&event.keyCode<=57" |

این کد تنها اعداد را می‌پذیرد. نمادها، فاصله‌ها یا برخی کاراکترهای خاص دیگر نیز بررسی و فیلتر می‌شوند تا از تزریق کد مخرب جلوگیری شود.

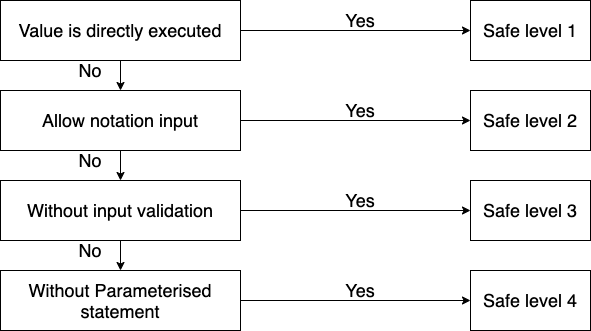
1. پارامتری‌سازی: متغیرهای ورودی کاربر نباید مستقیماً در عبارت شرطی درج شوند و باید فیلتر شوند. در پارامتری‌سازی، از دستورات پارامتری برای ارسال متغیرهای ورودی استفاده می‌شود. به‌جای درج مستقیم متغیرهای ورودی کاربر در عبارت شرطی، از پارامترها استفاده می‌شود. کد مربوطه در زیر نشان‌داده‌شده است:

if(is numeric(✩usearchtwo)=="true"){} else

echo "Incorrect.";

این قطعه کد بررسی می‌کند که آیا پرس‌وجو حاوی عدد است و در صورت وجود، مقدار را می‌پذیرد .[29]

1. شناسایی ویژگی‌های مخرب: شناسایی ویژگی‌های مخرب برای تشخیص این است که آیا سیستم یا نرم‌افزار دارای ویژگی‌هایی است که برای امنیت خطرناک هستند یا خیر. این شناسایی بر اساس کدها و ویژگی‌های مخرب انجام می‌شود. این فرایند می‌تواند به توسعه‌دهندگان کمک کند تا سطح ایمنی پروژه‌های خود را ارزیابی کنند. هرچه عدد سطح ایمنی بالاتر باشد، پایگاه‌داده غیررابطه‌ای امن‌تر است.



شکل2- 5: فلوچارت کد جاوا اسکریپت برای محدود کردن ورودی[29]

# فصل سوم روش پیشنهادی

این فصل به معرفی و تشریح روش پیشنهادی پژوهش می‌پردازد که هدف آن ارائه راهکاری جامع برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای است. در این روش، از مدل‌های زبان بزرگ برای تولید داده‌های مصنوعی باکیفیت و تنوع بالا استفاده می‌شود که می‌تواند محدودیت‌های موجود در دسترسی به مجموعه‌داده‌های واقعی را کاهش دهد. داده‌های تولیدشده در ادامه برای آموزش مدل‌های یادگیری ماشینی مانند رگرسیون خطی، درخت تصادفی، و XGBoost مورداستفاده قرار می‌گیرند. انتخاب این مدل‌ها بر اساس ویژگی‌های آن‌ها در شناسایی الگوهای پیچیده و قابلیت ترکیب در یک سیستم یادگیری گروهی انجام شده است.

علاوه بر این، روش پیشنهادی شامل تنظیم دقیق پارامترهای مدل‌ها و بهینه‌سازی آن‌ها برای افزایش دقت و کاهش نرخ خطا است. معماری کلی سیستم پیشنهادی و مراحل مختلف پیاده‌سازی، از تولید داده‌های مصنوعی تا ارزیابی مدل‌های یادگیری ماشینی، در این فصل تشریح می‌شود. همچنین، چالش‌های مرتبط با اجرای این روش، از جمله پیچیدگی تولید داده‌ها و تنظیم مدل‌ها، موردبحث قرار گرفته و راه‌حل‌های ارائه‌شده برای غلبه بر آن‌ها بررسی می‌شود.

هدف این فصل، ارائه یک نمای جامع از روش پیشنهادی و گام‌های لازم برای پیاده‌سازی آن است تا بتواند به‌عنوان یک ابزار مؤثر در شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای به کار گرفته شود.

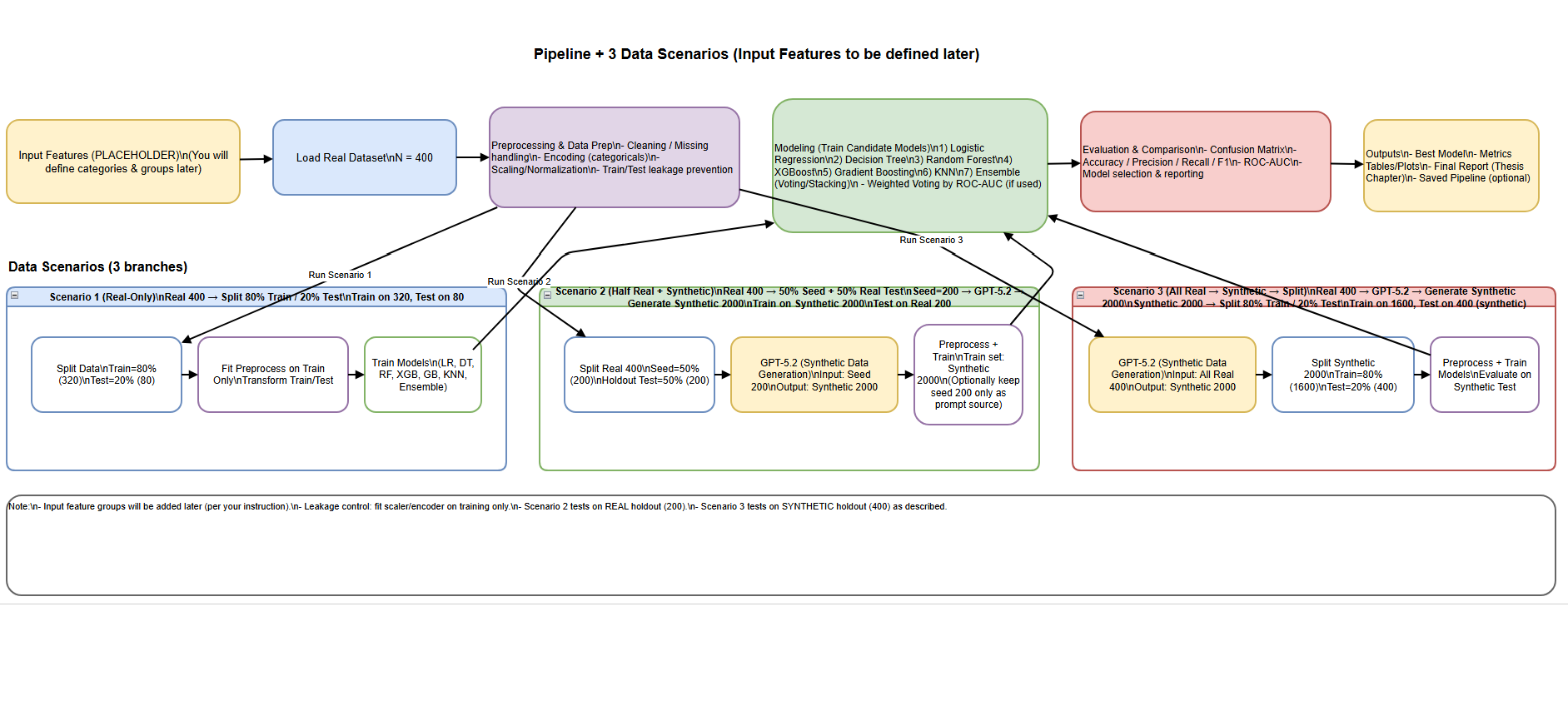
## 3-1 معرفی و توصیف رویکرد کلی پژوهش

رویکرد کلی این پژوهش برای مقابله با حملات تزریق در پایگاه‌داده‌های غیررابطه‌ای، بر یک خط‌لوله یادگیری ماشین مبتنی است که (1) از داده واقعی به‌عنوان مبنای ارزیابی و (2) از داده مصنوعی تولیدشده توسط مدل زبانی بزرگ برای افزایش تنوع سناریوها و پوشش الگوهای حمله استفاده می‌کند. در این چارچوب، پس از تعریف/استخراج ویژگی‌های ورودی، داده‌ها وارد مرحله پیش‌پردازش و سپس وارد مرحله مدل‌سازی (آموزش، اعتبارسنجی و تجمیع مدل‌ها) می‌شوند و در نهایت با معیارهای استاندارد ارزیابی و گزارش‌دهی می‌گردند.

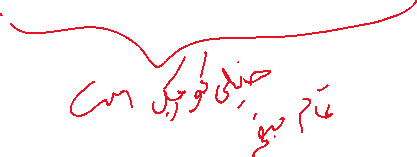
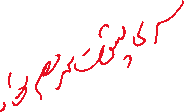
اهداف اصلی پژوهش

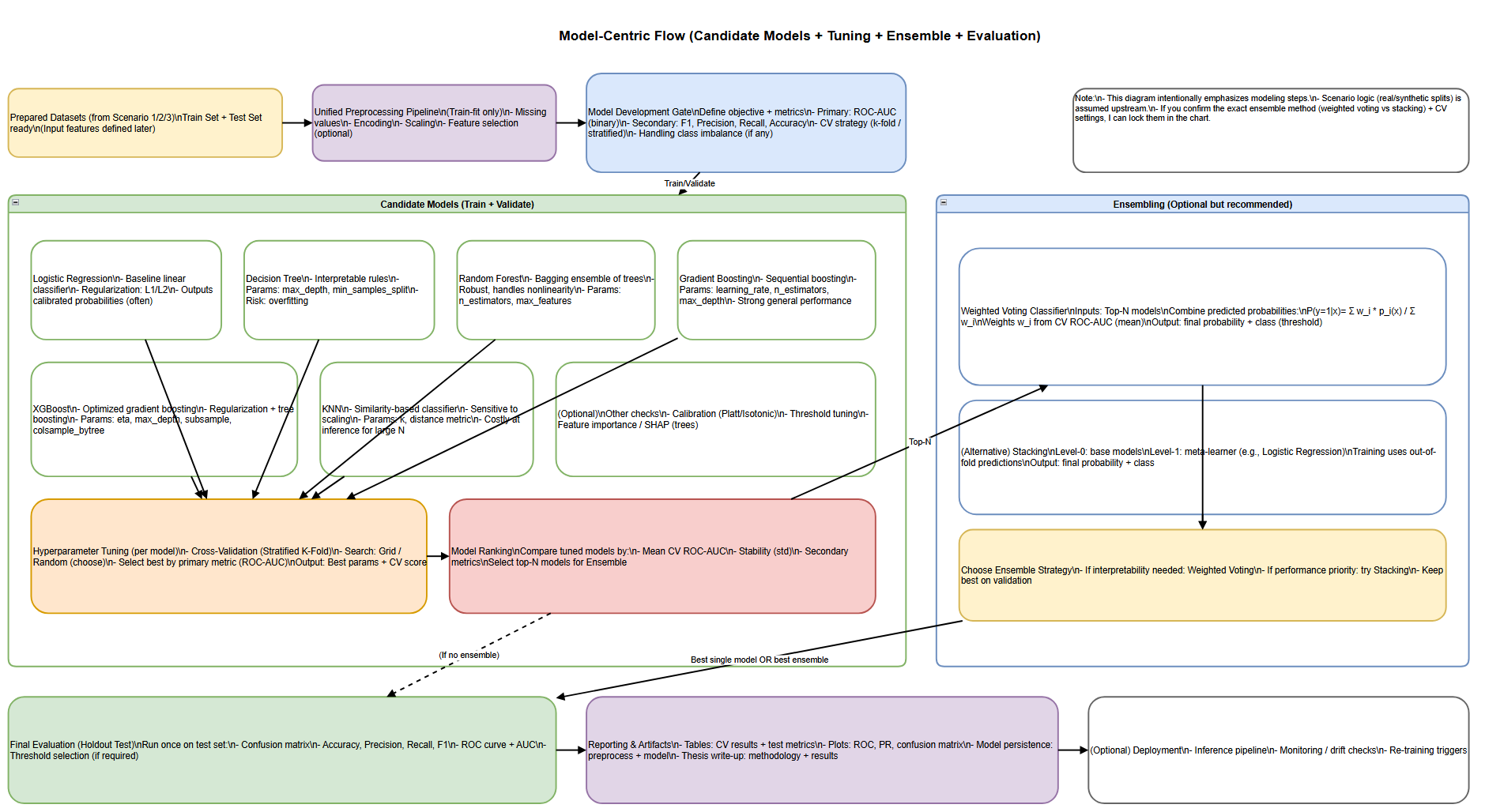
اهداف این پژوهش در چهار محور زیر خلاصه می‌شود:

* شناسایی حملات تزریق NoSQL با یادگیری ماشین: طراحی و ارزیابی مدل‌های طبقه‌بندی دودویی برای تشخیص حمله/غیرحمله.
* رفع کمبود و محدودیت داده با تولید داده مصنوعی: تولید نمونه‌های مصنوعی با کیفیت و تنوع بالا برای تقویت یادگیری مدل‌ها و کاهش حساسیت به محدودیت‌های داده واقعی.
* بهبود دقت و تعمیم‌پذیری با سناریوهای ترکیبی داده: بررسی عملکرد مدل‌ها در حالت استفاده از داده واقعی، داده مصنوعی و ترکیب آن‌ها در سناریوهای مختلف.
* به‌کارگیری LLM برای گسترش فضای حملات: استفاده از مدل زبان بزرگ جهت تولید نمونه‌های جدید (با حفظ ساختار و ویژگی‌های کلیدی) برای پوشش بهتر انواع الگوهای تزریق و حالت‌های نزدیک به داده عملیاتی.



شکل 3- 1: مدل مفهومی جزییات دیتا ست ها





شکل 3- 2: مدل مفهومی جزییات پیاده سازی مدل پیشنهادی

### 3-1-1 خط‌لوله کلی پژوهش

مطابق مدل مفهومی، مسیر پردازش از ورودی تا خروجی شامل مراحل زیر است:

1. تعریف/استخراج ویژگی‌ها (Input Features):

ویژگی‌های ورودی در قالب مجموعه‌ای از متغیرهای عددی/دسته‌ای از داده‌های درخواست/پرس‌وجو (طبق طرح پژوهش) تعریف و آماده می‌شوند.

1. بارگذاری داده واقعی:

داده واقعی با حجم محدود (N=400 در سناریوهای نشان‌داده‌شده) به‌عنوان مرجع اصلی اعتبارسنجی به کار می‌رود.

1. پیش‌پردازش و آماده‌سازی داده:

پاک‌سازی، مدیریت داده‌های گمشده، کدگذاری ویژگی‌های دسته‌ای، نرمال‌سازی/ استانداردسازی و رعایت کنترل نشت اطلاعات (Train/Test Leakage) انجام می‌شود؛ به‌نحوی که هرگونه برازش تبدیل‌ها روی داده آموزشی انجام و سپس روی داده آزمون اعمال شود.

1. مدل‌سازی و آموزش مدل‌های کاندید:

مجموعه‌ای از مدل‌های طبقه‌بندی آموزش داده می‌شوند و سپس بهترین‌ها (یا بهترین تجمیع) انتخاب می‌گردند.

1. ارزیابی و مقایسه:

ارزیابی با معیارهایی نظیر Accuracy، Precision، Recall، F1 و ROC-AUC و همچنین ماتریس درهم‌ریختگی انجام می‌شود. علاوه بر این، اعتبارسنجی متقاطع با StratifiedKFold برای پایداری نتایج به کار می‌رود (تقسیم‌بندی طبقه‌بندی‌شده برای حفظ توزیع کلاس‌ها در فولدها).

1. خروجی‌ها و گزارش نهایی:

انتخاب بهترین مدل/بهترین تجمیع، تولید جداول/نمودارها، گزارش معیارها و درج خروجی‌ها در قالب فصل نتایج پایان‌نامه.

### 3-1-2سناریوهای داده (سه شاخه اجرایی)

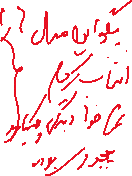
این پژوهش شامل طراحی و ارزیابی یک سیستم یادگیری گروهی است که از مدل‌های مختلف مانند رگرسیون لجستیک، جنگل تصادفی و XGBoost تشکیل شده است. این مدل‌ها با ترکیب قدرت یادگیری فردی خود، دقت شناسایی حملات را بهبود می‌بخشند. دستاوردهای مورد انتظار این پژوهش شامل ارائه روشی نوین برای شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای، ایجاد مجموعه‌دادگان جامع و متنوع، و بهبود دقت مدل‌های یادگیری ماشینی در محیط‌های واقعی است.

مفاهیم کلیدی این پژوهش شامل شناسایی و مقابله با حملات تزریق پایگاه‌داده غیررابطه‌ای است که از نقاط ضعف این نوع پایگاه‌ها سوءاستفاده می‌کنند. این حملات معمولاً از طریق ارسال درخواست‌های مخرب و دست‌کاری پرس‌وجوها انجام می‌گیرند. همچنین، تحلیل و طراحی مجموعه‌دادگان شامل داده‌های اولیه و داده‌های مصنوعی تولیدشده با مدل‌های زبان بزرگ، یکی از محورهای اصلی پژوهش است. در این راستا، به‌جای استفاده از شبکه‌های مولد متخاصم، از مدل‌های زبان بهره گرفته می‌شود که توانایی تولید داده‌های متنی و ساختاریافته باکیفیت بالا را دارند.

این پژوهش با هدف ارائه راهکاری جامع برای تشخیص حملات تزریق پایگاه‌داده غیررابطه‌ای طراحی شده است. برای دست‌یابی به این هدف، مجموعه‌داده‌های اولیه به‌صورت دستی تحلیل و برچسب‌گذاری می‌شوند تا کیفیت داده‌های پایه تضمین گردد. سپس از مدل‌های زبان بزرگ برای تولید داده‌های مصنوعی متنوع و واقعی‌تر استفاده می‌شود که به غنی‌سازی مجموعه‌دادگان کمک می‌کند. در مرحله بعد، تکنیک‌های مختلف یادگیری ماشینی برای ارزیابی و دسته‌بندی داده‌های موجود و مصنوعی به‌کار گرفته می‌شوند.

### 3-1-3 مدل‌های یادگیری ماشین و منطق انتخاب

در مرحله مدل‌سازی، یک سبد از مدل‌های کاندید (برای مقایسه جامع) و همچنین دو رویکرد تجمیعی استفاده می‌شود:



الف) مدل‌های کاندید (Candidate Models)

* رگرسیون خطی
* درخت تصمیم
* Gradient Boosting
* نزدیکترین همسایه
* درخت تصادفی
* XGBoost
* LightGBM
* Extra Trees
* Support Vector Machine
* AdaBoost

این تنوع باعث می‌شود هم مدل‌های ساده/تفسیرپذیر (مانند Logistic/Tree) و هم مدل‌های قوی در الگوهای غیرخطی (مانند خانواده Boosting و جنگل‌ها) و هم روش‌های مبتنی بر فاصله/حاشیه (KNN/SVM) پوشش داده شوند.

ب) یادگیری گروهی (Ensembling)

برای افزایش پایداری و کاهش خطای وابسته به یک مدل منفرد، دو نوع تجمیع استفاده می‌شود:

1. VotingClassifier با رأی‌گیری نرم (Soft Voting) و وزن‌دهی

در رأی‌گیری نرم، کلاس نهایی بر اساس بیشینه مجموع (یا مجموع وزن‌دار) احتمال‌های پیش‌بینی‌شده تعیین می‌شود؛ این روش برای ترکیب مدل‌هایی که خروجی احتمالاتی تولید می‌کنند توصیه می‌شود.

1. StackingClassifier با stack\_method='predict\_proba'

در Stacking، یک مدل نهایی (Meta-learner) بر پایه خروجی مدل‌های پایه آموزش می‌بیند؛ در حالت predict\_proba برای طبقه‌بندی دودویی، ستون اول احتمال هر مدل حذف می‌شود تا هم‌خطی کامل رخ ندهد.

### 3-1-4 راهبرد ارزیابی و گزارش‌دهی

برای ارزیابی منصفانه و پایدار:

* از اعتبارسنجی متقاطع طبقه‌بندی‌شده (StratifiedKFold) جهت حفظ نسبت کلاس‌ها در هر فولد و کنترل تصادفی‌سازی استفاده می‌شود.
* برای هر مدل، علاوه بر گزارش معیارهای عملکرد، ماتریس درهم‌ریختگی و شاخص‌های تکمیلی ارائه می‌گردد.
* در نهایت، بهترین مدل منفرد یا بهترین مدل تجمیعی انتخاب و نتایج در قالب جداول/نمودارها و گزارش فصل نتایج مستندسازی می‌شود.

بنابراین، رویکرد این پژوهش یک چارچوب داده‌محور و مدل‌محور است که با تعریف سناریوهای مختلف داده (واقعی/مصنوعی/ترکیبی)، اجرای مجموعه‌ای از مدل‌های کاندید و به‌کارگیری روش‌های تجمیعی (Soft Voting و Stacking)، تلاش می‌کند تشخیص حملات تزریق NoSQL را هم از نظر دقت و هم از نظر تعمیم‌پذیری تقویت کند.

## 3-2روش جمع‌آوری داده



در این بخش، فرایند تولید داده‌های مصنوعی برای حملات تزریق پایگاه‌داده غیررابطه‌ای با استفاده از مدل‌های زبان بزرگ (مدل‌های زبان بزرگ) نظیر GPT تشریح می‌شود. این فرایند شامل استخراج ویژگی‌های مهم از مجموعه‌داده اولیه و تولید داده‌های مصنوعی تصادفی و متنوع با استفاده از مدل‌های زبان بزرگ است. در ادامه، مراحل این فرایند به طور علمی و گام‌به‌گام تشریح می‌شود.

**3-3-1 استخراج ویژگی‌های کلیدی از مجموعه‌داده اولیه**

1- مرحله آماده‌سازی مجموعه‌داده اولیه

برای شروع تولید داده‌های مصنوعی، یک مجموعه‌داده اولیه شامل داده‌های مرتبط با پرس‌وجوهای پایگاه‌داده غیررابطه‌ای به مدل GPT ارائه شد. هدف این مرحله شناسایی ویژگی‌های کلیدی از داده‌های واقعی به‌منظور استفاده به‌عنوان مبنایی برای تولید داده‌های مصنوعی بود.

جدول 3- 2: نمونه ای از داده اولیه

|  |  |
| --- | --- |
| text | label |
| {"user": "{}", "password": "{}"} | 0 |
| {"username": {"$regex": "^{}"}} | 1 |

2- درخواست استخراج ویژگی‌ها از مدل مدل‌های زبان بزرگ

مدل GPT با استفاده از یک پرامپت مشخص، وظیفه استخراج ویژگی‌های مهم و معنادار از مجموعه‌داده اولیه را برعهده گرفت. این ویژگی‌ها شامل پارامترهایی بودند که ارتباط مستقیم با ساختار داده‌های پایگاه‌داده غیررابطه‌ای و امکان تشخیص الگوهای مخرب داشتند. نتیجه این مرحله منجر به شناسایی ویژگی‌های زیر شد:

جدول 3- 3: ویژگی های استخراج شده

|  |  |
| --- | --- |
| ویژگی‌ها | شرح ویژگی |
| num\_keys | تعداد کلیدها در ساختار داده. |
| num\_operators | تعداد عملگرهای مورداستفاده در پرس‌وجو. |
| length\_user\_value | طول مقادیر مرتبط با کاربر. |
| length\_password\_value | طول مقادیر مرتبط با رمز عبور. |
| length\_username\_value | طول مقادیر مرتبط با نام کاربری. |
| contains\_email\_pattern | وجود الگوهای آدرس ایمیل در داده‌ها. |
| contains\_ip\_pattern | وجود الگوهای مرتبط با آدرس IP در داده‌ها. |
| nested\_keys\_exist | وجود کلیدهای تودرتو در ساختار داده. |
| num\_nested\_keys | تعداد کلیدهای تودرتو موجود. |
| contains\_numeric\_values | وجود مقادیر عددی در داده‌ها. |
| contains\_boolean\_values | وجود مقادیر بولی (True/False) در داده‌ها. |
| contains\_empty\_strings | وجود مقادیر خالی در داده‌ها. |
| contains\_fixed\_values | وجود مقادیر ثابت (Hardcoded) در داده‌ها. |
| unique\_key\_count | تعداد کلیدهای یکتا در ساختار داده. |
| data\_type\_diversity | تنوع انواع داده‌ها (عددی، رشته‌ای و ...). |
| total\_record\_length | طول کل رکورد داده. |
| special\_characters\_exist | وجود کاراکترهای خاص در مقادیر ورودی. |
| nested\_depth | عمق تودرتو بودن داده‌ها. |
| label | برچسب‌گذاری داده‌ها برای مشخص‌کردن مخرب یا غیرمخرب بودن. |

تحلیل ویژگی‌های استخراج‌شده: ویژگی‌های استخراج‌شده، به‌عنوان معیارهایی برای توصیف رفتار و ساختار داده‌ها به کار گرفته شدند. این ویژگی‌ها نقش کلیدی در تولید داده‌های مصنوعی داشتند، زیرا داده‌های تولیدشده باید این ویژگی‌ها را رعایت کنند تا بتوانند شباهت زیادی به داده‌های واقعی داشته باشند.

### 3-2-1 تولید داده‌های مصنوعی با استفاده از مدل ChatGPT (GPT-5.2)

در این پژوهش، به‌منظور غلبه بر محدودیت حجم و تنوع داده‌های واقعی و افزایش پوشش سناریوهای حمله/غیرحمله، از مدل زبانی بزرگ GPT-5.2 در محیط ChatGPT برای تولید داده‌های مصنوعی ساختاریافته استفاده شد. GPT-5.2 به‌عنوان یکی از مدل‌های پیشرفته OpenAI با توانایی پردازش و تولید محتوای ساختاریافته و پشتیبانی از خروجی‌های دقیق و قابل‌کنترل انتخاب شد.

الف) طراحی پرامپت برای تولید داده مصنوعی

پس از استخراج ویژگی‌های کلیدی از داده واقعی، یک پرامپت هدفمند تدوین شد تا مدل GPT-5.2 بتواند نمونه‌های مصنوعی را با حفظ ساختار داده و رعایت قیود آماری/ساختاری تولید کند. اصول طراحی پرامپت با اتکا به راهنمای رسمی «Prompt engineering» و الگوهای پیشنهادشده برای «Synthetic Data Generation» تنظیم شد.

اجزای اصلی پرامپت شامل موارد زیر بود:

* تعریف دقیق قالب خروجی (ساختار رکوردها، نوع هر ستون/فیلد، محدودیت‌ها و فرمت ثابت خروجی برای جلوگیری از تولید متن آزاد(
* توصیف ویژگی‌های استخراج‌شده از داده واقعی (برای نمونه: توزیع یا بازه‌های مقادیر، تعداد کلیدها، عمق تودرتویی، وجود/عدم وجود الگوهای مشخص و …)
* الزام به تولید نمونه‌های متنوع با حفظ قواعد ساختاری (تنوع در مقادیر در عین سازگاری با ساختار)
* تعیین حجم تولید (تولید تعداد موردنیاز نمونه‌های مصنوعی؛ در این پژوهش 20,000 رکورد)

ب) فرآیند تولید و آماده‌سازی داده مصنوعی

با اجرای پرامپت تدوین‌شده روی GPT-5.2، مجموعه‌ای از داده‌های مصنوعی تولید شد که:

* از نظر ساختار و قالب با داده واقعی هم‌راستا است،
* از نظر تنوع سناریو دامنه وسیع‌تری از حالت‌های حمله/غیرحمله را پوشش می‌دهد،
* و برای استفاده در مراحل آموزش/اعتبارسنجی مدل‌های یادگیری ماشین قابل بهره‌برداری است.

سپس داده‌های مصنوعی تولیدشده، با توجه به سناریوی اجرایی پژوهش، به یکی از شکل‌های زیر مورد استفاده قرار گرفت:

* ترکیب با داده واقعی برای افزایش حجم/تنوع آموزش،
* یا آموزش روی مصنوعی و آزمون روی واقعی برای سنجش تعمیم‌پذیری،
* یا تقسیم داده مصنوعی به آموزش/آزمون برای تحلیل کیفیت داده تولیدی (مطابق سناریوهای تعریف‌شده در بخش رویکرد و دیاگرام‌ها).

ج) دستاوردهای تولید داده مصنوعی با GPT-5.2

فرآیند تولید داده مصنوعی با GPT-5.2 نتایج زیر را برای پژوهش به همراه داشت:

1. افزایش تنوع و پوشش سناریوها: ایجاد نمونه‌های جدید که می‌تواند فضای حالت‌های تزریق و الگوهای غیرخطی را بهتر پوشش دهد.
2. کاهش اثر کمبود داده واقعی: با تولید 20,000 نمونه مصنوعی، محدودیت حجم داده واقعی تا حد زیادی جبران شد و امکان آموزش/ارزیابی پایدارتر فراهم گردید.
3. تقویت کیفیت آموزش مدل‌ها: داده متنوع‌تر معمولاً به بهبود یادگیری الگوها و کاهش حساسیت مدل به موارد خاص داده واقعی کمک می‌کند (به‌ویژه در مسائل کم‌داده).
4. صرفه‌جویی در زمان و هزینه تولید داده: تولید داده با LLMها می‌تواند مسیر ساخت دیتاست را نسبت به جمع‌آوری و برچسب‌گذاری دستی سریع‌تر و مقرون‌به‌صرفه‌تر کند، مشروط به اعمال کنترل کیفیت.

## 4-3 معرفی مدل‌های یادگیری ماشین و مدل‌های تجمیعی به‌کاررفته

در این پژوهش، برای تشخیص دودویی حمله/غیرحمله در داده‌های مرتبط با تزریق NoSQL، مجموعه‌ای از الگوریتم‌های یادگیری ماشین شامل مدل‌های خطی، مبتنی بر فاصله، مبتنی بر درخت و روش‌های تقویتی (Boosting) به‌همراه دو رویکرد تجمیع مدل‌ها (Voting و Stacking) پیاده‌سازی و ارزیابی شده‌اند. هدف از به‌کارگیری این طیف متنوع از مدل‌ها، مقایسه عملکرد روش‌های پایه و پیشرفته و همچنین بررسی اثر تجمیع چند مدل بر بهبود دقت و پایداری تشخیص است.

رگرسیون لجستیک (رگرسیون خطی)

رگرسیون لجستیک یک الگوریتم یادگیری نظارت‌شده برای مسائل طبقه‌بندی دودویی است که با استفاده از تابع سیگموئید، احتمال تعلق نمونه به کلاس مثبت (حمله) را در بازه [0,1]برآورد می‌کند. این مدل به‌عنوان یک روش پایه و تفسیرپذیر در پژوهش استفاده شده است.  
در پیاده‌سازی انجام‌شده، رگرسیون لجستیک با تنظیمات penalty='l2'، مقدار منظم‌سازی C=10.0، حل‌گر liblinear و max\_iter=2000 اجرا شده است تا هم پایداری همگرایی تضمین شود و هم عملکرد مناسب در طبقه‌بندی دودویی فراهم گردد.

دلیل استفاده: مدل پایه برای مقایسه با سایر الگوریتم‌ها و ارائه یک معیار مرجع ساده و قابل تفسیر.

درخت تصمیم (درخت تصمیم)

درخت تصمیم یک مدل مبتنی بر قوانین تصمیم‌گیری است که فضای ویژگی‌ها را به‌صورت بازگشتی تقسیم می‌کند و در هر گره بر اساس معیارهایی مانند Gini بهترین تقسیم را انتخاب می‌کند.  
در این پژوهش، این مدل با محدودیت عمق (max\_depth=10) و تنظیم حداقل نمونه‌های تقسیم و برگ min\_samples\_split=5، min\_samples\_leaf=2 پیاده‌سازی شده است تا از بیش‌برازش شدید جلوگیری شود.

دلیل استفاده: ارائه یک مدل قابل تفسیر و مقایسه عملکرد روش‌های تک‌درختی با روش‌های جنگلی و تقویتی.

نزدیک‌ترین همسایه‌ها (نزدیکترین همسایه)

KNN یک روش مبتنی بر شباهت است که بر اساس فاصله نمونه از همسایگان نزدیک، کلاس را تعیین می‌کند. در این پژوهش KNN با n\_neighbors=3، وزن‌دهی فاصله‌ای (weights='distance') و معیار فاصله Manhattan metric='manhattan' با p=1استفاده شده است.

دلیل استفاده: سنجش کارایی یک مدل مبتنی بر فاصله در تشخیص الگوهای حمله/غیرحمله، خصوصاً زمانی که مرز تصمیم غیرخطی و محلی باشد.

ماشین بردار پشتیبان (Support Vector Machine)

SVM با هدف یافتن مرز تصمیم بهینه بین کلاس‌ها به‌کار می‌رود و در مسائل غیرخطی می‌تواند از کرنل استفاده کند. در این پژوهش، SVM با کرنل RBF و پارامتر C=10.0 اجرا شده و گزینه probability=True فعال شده است تا امکان تولید احتمال و محاسبه معیارهایی مانند AUC و همچنین مشارکت در مدل رأی‌گیری نرم فراهم شود.

دلیل استفاده: توانایی مدل‌سازی مرزهای تصمیم غیرخطی و مقایسه با مدل‌های درختی و تقویتی.

جنگل تصادفی (درخت تصادفی)

جنگل تصادفی یک روش یادگیری گروهی مبتنی بر Bagging است که با ساخت تعداد زیادی درخت تصمیم و ترکیب خروجی آن‌ها، پایداری و دقت را افزایش می‌دهد.  
در این پژوهش، جنگل تصادفی با n\_estimators=500، عمق max\_depth=12، انتخاب ویژگی به روش max\_features='sqrt'، و فعال‌سازی bootstrap=True اجرا شده است. همچنین گزینه oob\_score=True برای برآورد خطای خارج از کیسه (Out-of-Bag) فعال شده است.

دلیل استفاده: قدرت بالا در کشف روابط غیرخطی و مقاومت مناسب در برابر نویز و بیش‌برازش نسبت به درخت تصمیم منفرد.

درخت‌های کاملاً تصادفی (Extra Trees)

Extra Trees مشابه جنگل تصادفی است، اما در انتخاب نقاط تقسیم تصادفی‌تر عمل می‌کند و می‌تواند واریانس را کاهش دهد. در این پژوهش این مدل با تنظیمات هم‌راستا با جنگل تصادفی (n\_estimators=500، max\_depth=12، max\_features='sqrt') اجرا شده است.

دلیل استفاده: مقایسه با درخت تصادفی و بررسی اثر تصادفی‌سازی بیشتر در تقسیم‌ها بر عملکرد تشخیص.

گرادیان بوستینگ (Gradient Boosting)

Gradient Boosting یک روش تقویتی است که درخت‌ها را به‌صورت ترتیبی می‌سازد و هر مرحله خطای مرحله قبل را کاهش می‌دهد. در این پژوهش مدل گرادیان بوستینگ با n\_estimators=300، learning\_rate=0.15، subsample=0.9 و max\_depth=6 پیاده‌سازی شده است. همچنین از سازوکار early stopping داخلی (n\_iter\_no\_change=15, tol=1e-5) و validation\_fraction=0.15 برای کنترل بیش‌برازش استفاده شده است.

دلیل استفاده: ارائه یک مدل تقویتی کلاسیک برای مقایسه با روش‌های Boosting پیشرفته‌تر مانند XGBoost و LightGBM.

آدابوست (AdaBoost)

AdaBoost با تاکید بیشتر بر نمونه‌هایی که در مراحل قبل اشتباه طبقه‌بندی شده‌اند، مدل‌های ضعیف را به‌صورت ترتیبی تقویت می‌کند. در این پژوهش، AdaBoost با n\_estimators=200 و learning\_rate=1.2 و الگوریتم SAMME.R اجرا شده است.

دلیل استفاده: بررسی توان یک روش Boosting ساده‌تر در مقایسه با Gradient Boosting و XGBoost/LightGBM.

XGBoost

XGBoost یک نسخه بهینه‌شده از Boosting مبتنی بر گرادیان است که با پیاده‌سازی کارآمد و منظم‌سازی، در مسائل طبقه‌بندی عملکرد بالایی دارد. در این پژوهش، XGBoost با n\_estimators=400، learning\_rate=0.15، max\_depth=8، و پارامترهای منظم‌سازی reg\_alpha=0.05 و reg\_lambda=0.05 اجرا شده است. همچنین برای افزایش سرعت از tree\_method='hist' استفاده شده و تابع هدف binary:logistic انتخاب شده است.

دلیل استفاده: مدل پیشرفته برای تشخیص الگوهای پیچیده و غیرخطی با کنترل مناسب بیش‌برازش و سرعت اجرایی بالا.

LightGBM

LightGBM یک روش Boosting مبتنی بر درخت است که به‌دلیل طراحی کارآمد، برای داده‌های بزرگ و ویژگی‌های متعدد مناسب است. در این پژوهش مدل LightGBM با n\_estimators=400، learning\_rate=0.15، max\_depth=8، num\_leaves=63، و پارامترهای منظم‌سازی reg\_alpha=0.05 و reg\_lambda=0.05 اجرا شده است. همچنین از subsample=0.9 و colsample\_bytree=0.9 برای کنترل بیش‌برازش بهره گرفته شده است.

دلیل استفاده: مقایسه یک روش Boosting مدرن و سریع با XGBoost و سایر مدل‌ها در تشخیص حملات.

مدل تجمیعی رأی‌گیری نرم (Soft Voting Ensemble)

به‌منظور بهره‌گیری از نقاط قوت چند الگوریتم، یک مدل VotingClassifier با رأی‌گیری نرم (Soft Voting) پیاده‌سازی شده است. در این روش، به‌جای رأی‌گیری بر مبنای کلاس نهایی، میانگین وزن‌دار احتمال‌های پیش‌بینی‌شده توسط مدل‌ها محاسبه و کلاس نهایی بر اساس آن تعیین می‌شود.  
در این پژوهش، اجزای مدل رأی‌گیری نرم شامل: KNN، درخت تصادفی، XGBoost، LightGBM و SVM هستند و برای ترکیب احتمال‌ها وزن‌ها به‌صورت [3, 2, 2, 2, 1] تنظیم شده است؛ به این ترتیب وزن بیشتری به KNN اختصاص یافته است.

دلیل استفاده: افزایش پایداری و کاهش وابستگی به یک مدل منفرد از طریق ترکیب مدل‌های ناهمگون.

مدل تجمیعی انباشته (Stacking Ensemble)

در روش Stacking، چند مدل پایه (Base Learners) ابتدا آموزش داده می‌شوند و سپس خروجی‌های آن‌ها به‌عنوان ورودی به یک مدل نهایی (Meta-Learner) داده می‌شود تا تصمیم نهایی را اتخاذ کند.

در این پژوهش، مدل Stacking با مدل‌های پایه KNN، درخت تصادفی، XGBoost، LightGBM و AdaBoost تعریف شده و مدل نهایی (Final Estimator) یک درخت تصادفی با n\_estimators=100 و max\_depth=8 است. همچنین برای تولید ورودی مدل نهایی از predict\_proba استفاده شده و اعتبارسنجی داخلی Stacking با cv=3 تنظیم شده است.

**3-5 ارائه جزئیات مربوط به تنظیم پارامترها**

جدول 3- 1: پارامترهای رگرسیون خطی

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| C | 10.0 |
| penalty | 'l2' |
| solver | 'liblinear' |
| random\_state | 42 |
| max\_iter | 2000 |

جدول 3- 2: پارامترهای درخت تصمیم

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| max\_depth | 10 |
| min\_samples\_split | 5 |
| min\_samples\_leaf | 2 |
| max\_features | None |
| criterion | 'gini' |
| random\_state | 42 |

جدول 3- 3: پارامترهای Gradient Boosting

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| n\_estimators | 300 |
| learning\_rate | 0.15 |
| subsample | 0.9 |
| max\_depth | 6 |
| min\_samples\_split | 5 |
| min\_samples\_leaf | 2 |
| max\_features | 'sqrt' |
| random\_state | 42 |
| validation\_fraction | 0.15 |
| n\_iter\_no\_change | 15 |
| tol | 1e-5 |

جدول 3- 4: پارامترهای نزدیکترین همسایه

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| n\_neighbors | 3 |
| weights | 'distance' |
| metric | 'manhattan' |
| p | 1 |

جدول 3- 5: پارامترهای درخت تصادفی

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| n\_estimators | 500 |
| max\_depth | 12 |
| min\_samples\_split | 3 |
| min\_samples\_leaf | 1 |
| max\_features | 'sqrt' |
| bootstrap | True |
| oob\_score | True |
| random\_state | 42 |
| n\_jobs | -1 |
| criterion | 'gini' |

جدول 3- 6: پارامترهای XGBoost

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| objective | 'binary:logistic' |
| n\_estimators | 400 |
| learning\_rate | 0.15 |
| max\_depth | 8 |
| min\_child\_weight | 1 |
| gamma | 0.1 |
| colsample\_bytree | 0.9 |
| subsample | 0.9 |
| reg\_alpha | 0.05 |
| reg\_lambda | 0.05 |
| random\_state | 42 |
| eval\_metric | 'logloss' |
| tree\_method | 'hist' |

جدول 3- 7: پارامترهای LightGBM

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| objective | 'binary' |
| n\_estimators | 400 |
| learning\_rate | 0.15 |
| max\_depth | 8 |
| num\_leaves | 63 |
| min\_child\_samples | 5 |
| subsample | 0.9 |
| colsample\_bytree | 0.9 |
| reg\_alpha | 0.05 |
| reg\_lambda | 0.05 |
| random\_state | 42 |
| verbose | -1 |
| boosting\_type | 'gbdt' |

جدول 3- 8:پارامترهای Extra Trees

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| n\_estimators | 500 |
| max\_depth | 12 |
| min\_samples\_split | 3 |
| min\_samples\_leaf | 1 |
| max\_features | 'sqrt' |
| bootstrap | True |
| random\_state | 42 |
| n\_jobs | -1 |
| criterion | 'gini' |

جدول 3- 9: پارامترهای Support Vector Machine

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| C | 10.0 |
| kernel | 'rbf' |
| gamma | 'scale' |
| probability | True |
| random\_state | 42 |

جدول 3- 10: پارامترهای AdaBoost

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| n\_estimators | 200 |
| learning\_rate | 1.2 |
| algorithm | 'SAMME.R' |
| random\_state | 42 |

**پارامترهای مدل‌های تجمیعی**

جدول 3- 11: پارامترهای Voting Ensemble

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| voting | 'soft' |
| weights | [3, 2, 2, 2, 1] |
| estimators | KNN, RF, XGB, LGB, SVM |

جدول 3- 12: پارامترهای Estimators داخل Voting Ensemble

|  |  |
| --- | --- |
| **Estimator** | **پارامترها** |
| KNN | n\_neighbors=3, weights='distance', metric='manhattan' |
| RF | n\_estimators=500, max\_depth=12, min\_samples\_split=3, min\_samples\_leaf=1, max\_features='sqrt', random\_state=42 |
| XGB | objective='binary:logistic', n\_estimators=400, learning\_rate=0.15, max\_depth=8, min\_child\_weight=1, gamma=0.1, colsample\_bytree=0.9, subsample=0.9, reg\_alpha=0.05, reg\_lambda=0.05, random\_state=42 |
| LGB | objective='binary', n\_estimators=400, learning\_rate=0.15, max\_depth=8, min\_child\_samples=5, subsample=0.9, colsample\_bytree=0.9, reg\_alpha=0.05, reg\_lambda=0.05, random\_state=42, verbose=-1 |
| SVM | C=10.0, kernel='rbf', gamma='scale', probability=True, random\_state=42 |

جدول 3- 13:پارامترهای Stacking Ensemble

|  |  |
| --- | --- |
| **پارامتر** | **مقدار** |
| stack\_method | 'predict\_proba' |
| cv | 3 |
| n\_jobs | -1 |
| final\_estimator | RandomForestClassifier |

جدول 3- 14: پارامترهای Estimators داخل Stacking Ensemble

|  |  |
| --- | --- |
| **Estimator** | **پارامترها** |
| KNN | n\_neighbors=3, weights='distance', metric='manhattan' |
| RF | n\_estimators=300, max\_depth=10, min\_samples\_split=3, min\_samples\_leaf=1, max\_features='sqrt', random\_state=42 |
| XGB | objective='binary:logistic', n\_estimators=300, learning\_rate=0.15, max\_depth=6, min\_child\_weight=1, gamma=0.1, colsample\_bytree=0.9, subsample=0.9, reg\_alpha=0.05, reg\_lambda=0.05, random\_state=42 |
| LGB | objective='binary', n\_estimators=300, learning\_rate=0.15, max\_depth=6, min\_child\_samples=5, subsample=0.9, colsample\_bytree=0.9, reg\_alpha=0.05, reg\_lambda=0.05, random\_state=42, verbose=-1 |
| AdaBoost | n\_estimators=100, learning\_rate=1.0, random\_state=42 |
| Final RF | n\_estimators=100, max\_depth=8, min\_samples\_split=3, min\_samples\_leaf=1, random\_state=42 |

## 3-6 پیاده‌سازی

سیستم پیشنهادی برای تشخیص حملات تزریق پایگاه‌داده‌های غیررابطه‌ای (NoSQL Injection) بر یک خط‌لوله یادگیری ماشین استوار است که شامل آماده‌سازی داده، ساخت سناریوهای داده واقعی/مصنوعی، آموزش و مقایسه چندین مدل کاندید، به‌کارگیری روش‌های تجمیعی (Ensemble) و در نهایت ارزیابی مبتنی بر اعتبارسنجی متقاطع طبقه‌بندی‌شده و آزمون نهایی است. همچنین، برای رفع محدودیت حجم و تنوع داده واقعی، داده‌های مصنوعی با استفاده از ChatGPT (GPT-5.2) تولید و در سناریوهای تعریف‌شده به کار گرفته شده‌اند.

در ادامه، مراحل پیاده‌سازی به‌صورت گام‌به‌گام ارائه می‌شود.

گام 1) بارگذاری داده و بررسی اولیه

ابتدا مجموعه‌داده(ها) از مسیرهای تعریف‌شده بارگذاری می‌شوند. در این مرحله، بررسی‌های مقدماتی زیر انجام می‌گیرد:

1. بررسی ساختار داده: تعداد رکوردها، تعداد ویژگی‌ها، نوع داده هر ستون و شناسایی ستون‌های عددی/دسته‌ای.
2. کنترل کیفیت اولیه: شناسایی مقادیر گمشده، مقادیر غیرمجاز و ناسازگاری‌های احتمالی در قالب داده.
3. بررسی توزیع برچسب‌ها: محاسبه توزیع کلاس‌ها (حمله/غیرحمله) به‌منظور تشخیص عدم‌توازن احتمالی در داده.

گام 2) تولید داده مصنوعی ChatGPT (GPT-5.2)

با توجه به محدودیت حجم و تنوع داده واقعی، در سناریوهای مشخص، داده مصنوعی با ChatGPT (GPT-5.2) تولید شد. تولید داده مصنوعی با پرامپت ساختارمند انجام گرفت تا:

* قالب رکوردها، نوع متغیرها و قیود ساختاری رعایت شود؛
* تنوع سناریوهای حمله/غیرحمله افزایش یابد؛
* داده تولیدی با ویژگی‌های کلیدی داده واقعی هم‌راستا باقی بماند.

خروجی این مرحله، یک مجموعه‌داده مصنوعی (Synthetic) قابل استفاده در مرحله آموزش/ارزیابی است.

گام 3) پیش‌پردازش داده (Data Preprocessing)

پس از آماده شدن داده (واقعی/مصنوعی/ترکیبی)، عملیات پیش‌پردازش با تمرکز بر سازگاری داده و جلوگیری از نشت اطلاعات انجام می‌شود. مؤلفه‌های اصلی این مرحله عبارت‌اند از:

3-1 ) مدیریت داده‌های گمشده

* برای ویژگی‌های عددی، مقداردهی جایگزین به‌صورت روش‌های استاندارد (مانند میانگین/میانه) انجام می‌شود.
* برای ویژگی‌های دسته‌ای، مقدار پرتکرار (Most Frequent) یا یک برچسب مشخص برای «نامشخص» لحاظ می‌شود.

3-2) کدگذاری ویژگی‌های دسته‌ای

ویژگی‌های دسته‌ای به نمایش عددی قابل پردازش برای مدل‌های یادگیری ماشین تبدیل می‌شوند (از روش‌های رایج کدگذاری متناسب با نوع داده).

3-3) مقیاس‌بندی/استانداردسازی

برای مدل‌هایی که نسبت به مقیاس ویژگی‌ها حساس‌اند (به‌ویژه روش‌های مبتنی بر فاصله و مدل‌های خطی)، ویژگی‌ها استانداردسازی می‌شوند. این کار باید با رعایت اصل عدم نشت اطلاعات انجام شود (برازش تنها روی داده آموزش و اعمال روی داده آزمون).

نکته روش‌شناسی: هرگونه تبدیل (Encoding/Scaling) باید روی داده آموزش برازش شود و سپس بر روی اعتبارسنجی/آزمون اعمال گردد تا برآورد عملکرد خوش‌بینانه نشود.

گام 4) تعریف سناریوهای داده برای آموزش و آزمون

به‌منظور سنجش اثر داده مصنوعی و بررسی تعمیم‌پذیری، سه سناریو اجرایی مطابق مدل مفهومی تعریف می‌شود:

1. سناریو 1 (Real-Only): تقسیم داده واقعی به آموزش/آزمون و اجرای کل فرآیند مدل‌سازی روی همین داده.
2. سناریو 2 (Half Real + Synthetic): بخشی از داده واقعی به‌عنوان «بذر تولید» و بخش دیگر به‌عنوان «آزمون واقعی» کنار گذاشته می‌شود؛ آموزش روی داده مصنوعی (یا ترکیب مشخص‌شده) و آزمون روی داده واقعی انجام می‌گردد.
3. سناریو 3 (All Real → Synthetic → Split): از کل داده واقعی برای تولید داده مصنوعی استفاده می‌شود؛ سپس داده مصنوعی به آموزش/آزمون تقسیم و ارزیابی انجام می‌گیرد.

گام 5) مدل‌سازی: آموزش مدل‌های کاندید و مدل‌های تجمیعی

در مرحله مدل‌سازی، دو دسته مدل پیاده‌سازی شده‌اند:

5-1) مدل‌های کاندید (Candidate Models)

برای مقایسه جامع، مجموعه‌ای از الگوریتم‌ها شامل مدل‌های خطی، مبتنی بر درخت، مبتنی بر فاصله، و تقویتی انتخاب و اجرا شدند، از جمله:

* رگرسیون خطی
* درخت تصمیم
* Gradient Boosting
* نزدیکترین همسایه
* درخت تصادفی
* XGBoost
* LightGBM
* Extra Trees
* Support Vector Machine
* AdaBoost

هدف این مرحله، مقایسه عملکرد خانواده‌های مختلف مدل و انتخاب گزینه‌های مناسب برای تجمیع (Ensembling) است.

5-2)مدل‌های تجمیعی (Ensemble Learning)

برای کاهش خطای مدل منفرد و افزایش پایداری، دو راهبرد تجمیعی پیاده‌سازی شد:

1. Soft Voting (VotingClassifier): ترکیب احتمالات خروجی چند مدل پایه با رأی‌گیری نرم و وزن‌دهی.
2. Stacking (StackingClassifier): استفاده از خروجی احتمالاتی چند مدل پایه به‌عنوان ورودی یک مدل نهایی (Meta-Learner) جهت تصمیم‌گیری نهایی.

گام 6) ارزیابی مدل‌ها

6-1) اعتبارسنجی متقاطع طبقه‌بندی‌شده (Stratified K-Fold CV)

برای برآورد پایدار عملکرد و کاهش وابستگی به یک تقسیم‌بندی خاص، از StratifiedKFold با 10 فولد استفاده می‌شود تا نسبت کلاس‌ها در هر فولد حفظ گردد.

6-2) ارزیابی روی مجموعه آزمون (Holdout Test)

پس از اعتبارسنجی، هر مدل روی کل داده آموزش آموزش داده شده و روی داده آزمون ارزیابی می‌شود. علاوه بر گزارش معیارها، ماتریس درهم‌ریختگی نیز برای تحلیل خطاها استخراج می‌گردد.

6-3) معیارهای عملکرد

برای مقایسه مدل‌ها، مجموعه‌ای از معیارها محاسبه می‌شود:

* Accuracy
* ROC-AUC
* Precision (macro)
* Recall (macro)
* F1-score (macro)
* Cohen’s Kappa
* Matthews Correlation Coefficient (MCC)
* Confusion Matrix

این معیارها امکان تحلیل عملکرد مدل‌ها را هم از منظر دقت کلی و هم از منظر پایداری در شرایط عدم‌توازن یا خطاهای طبقه‌بندی فراهم می‌کنند.

6-4) تحلیل زمان اجرا

به‌منظور بررسی کارایی محاسباتی، زمان آموزش و زمان پیش‌بینی مدل‌ها چندبار اندازه‌گیری و میانگین/انحراف معیار گزارش می‌شود. این مرحله به ویژه برای مقایسه مدل‌های سنگین‌تر (مانند Boostingها)با مدل‌های سبک‌تر اهمیت دارد.

گام 7) خروجی‌گیری و ذخیره نتایج

در پایان:

* خلاصه نتایج اعتبارسنجی متقاطع،
* نتایج آزمون نهایی،
* و نتایج تحلیل زمان اجرا، در قالب فایل‌های خروجی (CSV) ذخیره شده و برای ارائه در قالب جدول/نمودار در فصل نتایج استفاده می‌شوند.

پیاده‌سازی سیستم پیشنهادی به‌صورت مرحله‌ای و با رعایت اصول پیش‌پردازش صحیح (کنترل نشت اطلاعات)، تعریف سناریوهای داده واقعی/مصنوعی و ارزیابی دقیق انجام شده است. این چارچوب با مقایسه طیف متنوعی از مدل‌ها و نیز استفاده از روش‌های تجمیعی (Soft Voting و Stacking)، امکان انتخاب بهترین راهکار از منظر دقت، پایداری و کارایی محاسباتی را فراهم می‌کند و در نهایت راهکاری داده‌محور برای تشخیص حملات تزریق NoSQL ارائه می‌دهد.

|  |
| --- |
| **Pseudocode for the Proposed Algorithm (Modeling + Evaluation Pipeline)** |
| 1. **Import Required Libraries**  * Import libraries for:   + Data handling: pandas, numpy   + Model evaluation: StratifiedKFold, metrics (Accuracy, ROC-AUC, Recall, Precision, F1, Kappa, MCC, Confusion Matrix)   + Models: رگرسیون خطی, درخت تصمیم, KNN, SVM, درخت تصادفی, Gradient Boosting, Extra Trees, AdaBoost, VotingClassifier, StackingClassifier, XGBoost, LightGBM   + Visualization: matplotlib, seaborn   + Timing: time  1. **Define Cross-Validation Settings**  * Set n\_splits = 10  1. **Define Cross-Validation Evaluation Function**  * Define evaluate\_model\_cv(model, model\_name, X\_train, y\_train):   + Initialize StratifiedKFold(n\_splits=10, shuffle=True, random\_state=42)   + Create an empty table fold\_performance with columns:     - Model, Fold, Accuracy, AUC, Recall, Precision, F1, Kappa, MCC   + For each fold:     - Split X\_train, y\_train into train\_fold and val\_fold     - Fit the model on train\_fold     - Predict classes on val\_fold     - Predict probabilities on val\_fold (take column [:,1])     - Compute metrics:       * Accuracy       * ROC-AUC       * Recall (macro)       * Precision (macro)       * F1 (macro)       * Cohen’s Kappa       * Matthews Correlation Coefficient (MCC)     - Store fold results in fold\_performance   + Return fold\_performance  1. **Define Test-Set Evaluation Function**  * Define evaluate\_model\_test(model, model\_name, X\_train, y\_train, X\_test, y\_test):   + Fit model on full X\_train, y\_train and record training time   + Predict classes and probabilities on X\_test and record prediction time   + Compute metrics:     - Accuracy, ROC-AUC, Recall (macro), Precision (macro), F1 (macro), Kappa, MCC   + Compute Confusion Matrix   + Plot Confusion Matrix as heatmap   + Return a dictionary containing metrics + times + confusion matrix  1. **Define Timing Analysis Function**  * Define evaluate\_model\_timing(model, model\_name, X\_train, y\_train, X\_test, y\_test, n\_runs=5):   + Repeat n\_runs times:     - Fit model and store training time     - Predict on test set and store prediction time   + Return mean and standard deviation for training and prediction time  1. **Define Candidate Models (with Fixed Hyperparameters)**  * Create a dictionary models including:   + رگرسیون خطی   + درخت تصمیم   + Gradient Boosting   + KNN   + درخت تصادفی   + XGBoost   + LightGBM   + Extra Trees   + SVM (probability=True)   + AdaBoost   + Voting Ensemble (Soft Voting with weights)   + Stacking Ensemble (predict\_proba + final estimator)  1. **Run Cross-Validation for All Models**  * Initialize all\_cv\_performances as empty table * For each (model\_name, model) in models:   + Run evaluate\_model\_cv(...)   + Append fold results to all\_cv\_performances  1. **Summarize Cross-Validation Results**  * Group all\_cv\_performances by Model * Compute mean and std (where applicable) for:   + Accuracy, AUC, Recall, Precision, F1, Kappa, MCC * Print the summary table * Save summary as cross\_validation\_results.csv  1. **Evaluate All Models on the Test Set**  * Initialize test\_results as empty list * For each (model\_name, model) in models:   + Run evaluate\_model\_test(...)   + Store results in test\_results * Convert results to DataFrame * Drop Confusion\_Matrix column for clean display * Print test performance table * Save as test\_set\_results.csv  1. **Run Timing Analysis**  * Initialize timing\_results as empty list * For each (model\_name, model) in models:   + Run evaluate\_model\_timing(...)   + Store results * Convert to DataFrame and print timing table * Save as timing\_results.csv  1. **Output Final Artifacts**  * Display/print:   + Cross-validation summary   + Test-set performance   + Timing analysis * Save all CSV outputs for reporting and thesis documentation |

# فصل چهارم- تجزیه‌وتحلیل داده‌ها (یافته‌ها)

این فصل به ارائه و تحلیل یافته‌های پژوهش اختصاص دارد. در راستای تحقق اهداف پژوهش، مجموعه‌داده‌های پردازش‌شده با استفاده از مدل‌های پیشنهادی مورد تجزیه‌وتحلیل قرار گرفته‌اند. این فصل شامل بررسی عملکرد مدل‌های یادگیری ماشینی، تحلیل نتایج به‌دست‌آمده، و مقایسه دقیق معیارهای ارزیابی است.

معیارهای کلیدی برای ارزیابی عملکرد مدل‌ها عبارت‌اند از صحت، نرخ بازیابی [[33]](#footnote-34)، دقت پیش‌بینی[[34]](#footnote-35)، امتیاز F1 [[35]](#footnote-36)، و مساحت زیر منحنی مشخصه عملکرد سیستم [[36]](#footnote-37) این معیارها برای ارزیابی جامع توانایی مدل‌ها در شناسایی حملات تزریق پایگاه‌داده غیررابطه‌ای و تفکیک داده‌های مخرب از داده‌های قانونی استفاده می‌شوند.

علاوه بر این، تحلیل ماتریس درهم‌ریختگی[[37]](#footnote-38) و نمودارهای مربوط به عملکرد مدل‌ها به‌صورت دقیق موردبحث قرار می‌گیرد.

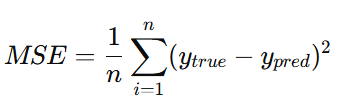
این فصل تلاش می‌کند تا با تحلیل دقیق نتایج و ارائه یافته‌های حاصل از مدل‌های پیشنهادی، شفافیت و جامعیت لازم را برای ارزیابی و درک عملکرد سیستم ارائه کند.

## 4-1 معیارهای ارزیابی عملکرد

ارزیابی عملکرد مدل‌های یادگیری ماشینی در پروژه‌های مرتبط با تشخیص حملات سایبری، به‌ویژه در سیستم‌های پیچیده‌ای؛ مانند حملات تزریق پایگاه‌داده غیررابطه‌ای، نقش بسیار مهمی در اطمینان از دقت و کارایی مدل‌ها ایفا می‌کند. در این پروژه، مجموعه‌ای از معیارهای کلیدی ارزیابی برای تحلیل عملکرد مدل‌های پیشنهادی و مقایسه آن‌ها با روش‌های موجود استفاده شده است. این معیارها به دودسته عددی و طبقه‌بندی تقسیم شده‌اند. در ادامه، این معیارها به همراه کاربردهای آن‌ها و دلایل اهمیت آن‌ها در پروژه توضیح داده شده‌اند.

**خطای میانگین مربعات**

[[38]](#footnote-39)MSE میانگین مجموع مربعات تفاوت بین مقادیر واقعی (ytrue) و پیش‌بینی‌شده (ypred) را محاسبه می‌کند.

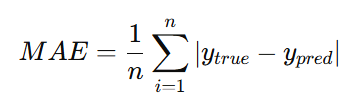


ویژگی‌ها:

* نسبت به خطاهای بزرگ بسیار حساس است، زیرا اختلاف‌ها به توان دو می‌رسند.
* برای تحلیل دقیق تأثیر خطاهای بزرگ مفید است.
* هدف: کاهش MSE نشان‌دهنده بهبود دقت مدل است.

**میانگین خطای مطلق**

[[39]](#footnote-40) MAE میانگین قدرمطلق تفاوت بین مقادیر واقعی و پیش‌بینی‌شده را محاسبه می‌کند.

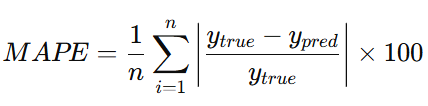


**ویژگی‌ها:**

* تفسیر آن ساده‌تر از MSE است؛ زیرا از توان دو استفاده نمی‌کند.
* کمتر به خطاهای بزرگ حساس است.
* **هدف:** کاهش MAE به معنی دقت بالاتر در پیش‌بینی‌ها است.

**میانگین درصد خطای مطلق**

[[40]](#footnote-41) MAPE میانگین درصد خطای پیش‌بینی را نسبت به مقادیر واقعی محاسبه می‌کند.

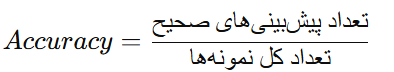


ویژگی‌ها:

* به واحد مقیاس داده‌ها وابسته نیست.
* مناسب برای ارزیابی مدل‌هایی که خروجی آن‌ها در بازه‌های مختلف قرار دارد.
* هدف: نشان‌دادن درصد خطای پیش‌بینی نسبت به مقدار واقعی.

**صحت [[41]](#footnote-42)**

* تعریف: صحت نسبت تعداد پیش‌بینی‌های صحیح به تعداد کل نمونه‌ها است.



**ویژگی‌ها:**

* برای داده‌های متعادل مناسب است.
* حساس به داده‌های نامتعادل نیست.
* هدف: بالابردن صحت نشان‌دهنده تعداد بیشتری از پیش‌بینی‌های صحیح مدل است.

**ناحیه زیر منحنی**

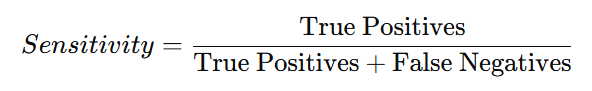
[[42]](#footnote-43) AUC معیاری برای سنجش عملکرد مدل طبقه‌بندی است که حساسیت [[43]](#footnote-44) و ویژگی [[44]](#footnote-45) را در سطوح مختلف آستانه ارزیابی می‌کند.

**ویژگی‌ها:**

* مقدار AUC بین 0 و 1 است، هرچه مقدار به 1 نزدیک‌تر باشد، مدل قوی‌تر است.
* مناسب برای داده‌های نامتعادل.
* **هدف:** مقایسه عملکرد مدل‌های مختلف بر اساس توانایی آن‌ها در تمایز بین کلاس‌ها.

**حساسیت**

حساسیت[[45]](#footnote-46) نشان می‌دهد که چه درصدی از نمونه‌های مثبت واقعی به درستی توسط مدل شناسایی شده‌اند.



**ویژگی‌ها:**

* برای کاربردهایی که شناسایی نمونه‌های مثبت اهمیت دارد، بسیار حیاتی است.
* هدف: افزایش حساسیت به معنای کاهش خطای ازدست‌دادن نمونه‌های مثبت است.

**ویژگی Specificity**

ویژگی Specificity نشان می‌دهد چه درصدی از نمونه‌های منفی واقعی به‌درستی توسط مدل شناسایی شده‌اند. ​

* + در کاربردهایی که شناسایی نمونه‌های منفی مهم است، اهمیت دارد.

**دقت پیش‌بینی**

تعریف: دقت پیش‌بینی نشان می‌دهد که چه درصدی از نمونه‌های پیش‌بینی‌شده مثبت، واقعاً مثبت هستند. ​

ویژگی‌ها:

* + مناسب برای کاربردهایی که هزینه پیش‌بینی نادرست مثبت بالاست.

**تحلیل معیارها در پروژه ما**

* + جدول زیر مقایسه معیارهای استفاده‌شده در پروژه ما را ارائه می‌دهد:

جدول 4- 1: معیارهای ارزیابی

|  |  |  |  |
| --- | --- | --- | --- |
| معیار | تعریف | کاربرد | پروژه ما |
| MSE | حساس به خطاهای بزرگ؛ میانگین مربعات خطا. | تحلیل خطای کلی مدل | ✅ |
| MAE | میانگین خطای مطلق؛ کمتر حساس به خطاهای بزرگ. | تحلیل میانگین خطا | ✅ |
| MAPE | میانگین درصد خطای مطلق؛ مستقل از مقیاس داده‌ها. | تحلیل درصد خطای پیش‌بینی | ✅ |
| Accuracy | نسبت پیش‌بینی‌های صحیح. | ارزیابی کلی پیش‌بینی‌ها | ✅ |
| AUC | توانایی مدل در تمایز بین کلاس‌ها. | تحلیل مدل در داده‌های نامتعادل | ✅ |
| Sensitivity | درصد شناسایی صحیح نمونه‌های مثبت. | شناسایی دقیق حملات | ✅ |
| Specificity | درصد شناسایی صحیح نمونه‌های منفی. | کاهش پیش‌بینی‌های نادرست مثبت | ✅ |
| Precision | درصد پیش‌بینی‌های مثبت که واقعاً مثبت هستند. | شناسایی دقیق نمونه‌های مثبت | ✅ |

## 4-2 بررسی دیتاست‌ها

در این بخش به تحلیل و بررسی دو مجموعه‌داده مورداستفاده در پروژه پرداخته می‌شود. این دو دیتاست شامل یک مجموعه اصلی و یک مجموعه مصنوعی تولیدشده با استفاده از مدل‌های زبان بزرگ (مدل‌های زبان بزرگ) هستند که به‌منظور جبران کمبود داده‌های واقعی و بهبود آموزش مدل‌های یادگیری ماشینی طراحی شده‌اند. هر یک از این مجموعه‌ها شامل ستون‌هایی با ویژگی‌های کلیدی است که مشخصات و ساختار داده‌های مرتبط با حملات تزریق پایگاه‌داده غیررابطه‌ای را توصیف می‌کنند.

**دیتاست اصلی**

1.1 مشخصات دیتاست اصلی

* تعداد رکوردها: 400 رکورد
* نوع داده‌ها: داده‌های واقعی که به‌صورت دستی جمع‌آوری و برچسب‌گذاری شده‌اند.
* ستون‌ها: شامل 18 ستون با مشخصات کلیدی برای توصیف داده‌ها، از جمله ویژگی‌های عددی و منطقی مرتبط با حملات پایگاه‌داده غیررابطه‌ای.

1.2 تحلیل ویژگی‌ها

* num\_keys و num\_operators: تعداد کلیدها و عملگرهای استفاده‌شده در هر پرس‌وجو.
* length\_user\_value و length\_password\_value: طول مقادیر مرتبط با نام کاربری و رمز عبور.
* contains\_email\_pattern و contains\_ip\_pattern: وجود الگوهای ایمیل و آدرس IP در داده‌ها.
* nested\_keys\_exist و nested\_depth: وجود و عمق کلیدهای تودرتو.
* contains\_numeric\_values و contains\_boolean\_values: وجود مقادیر عددی و بولی در داده‌ها.
* Label: برچسبی که نشان می‌دهد رکورد مربوط به حمله (1) است یا غیر حمله (0).

1.3 تحلیل محدودیت‌ها

دیتاست اصلی، علی‌رغم اینکه داده‌های واقعی و برچسب‌گذاری شده را شامل می‌شود، دارای محدودیت‌هایی است:

* تعداد کم رکوردها: تعداد رکوردها برای آموزش مدل‌های یادگیری ماشینی پیچیده ناکافی است.
* تنوع محدود: تنوع سناریوهای حمله و داده‌های غیرمخرب در این مجموعه نسبتاً کم است.
* عدم تعمیم‌پذیری: به دلیل حجم پایین داده، مدل‌های آموزش‌دیده ممکن است عملکرد ضعیفی روی داده‌های ناشناخته داشته باشند.

دیتاست مصنوعی

1.4 مشخصات دیتاست مصنوعی

* تعداد رکوردها: 2000 رکورد
* نوع داده‌ها: داده‌های تولیدشده به‌صورت مصنوعی با استفاده از مدل‌های زبان بزرگ
* ستون‌ها: شامل همان 18 ستون کلیدی که در دیتاست اصلی وجود دارند.

1.5 فرایند تولید دیتاست

دیتاست مصنوعی با استفاده از مدل مدل‌های زبان بزرگ و بر اساس تحلیل ویژگی‌های دیتاست اصلی تولید شده است. مراحل اصلی تولید عبارت‌اند از:

1. استخراج ویژگی‌های کلیدی از دیتاست اصلی: شناسایی ویژگی‌های مهم نظیر تعداد کلیدها، طول مقادیر، وجود مقادیر خاص (مانند ایمیل یا IP)، و برچسب‌ها.
2. تولید داده‌های مصنوعی توسط مدل‌های زبان بزرگ: مدل مدل‌های زبان بزرگ با استفاده از پرامپت‌های طراحی‌شده داده‌هایی مشابه دیتاست اصلی تولید کرده است.
3. تنوع‌سازی: اطمینان از ایجاد تنوع در داده‌ها، شامل سناریوهای مختلف حملات و داده‌های غیرمخرب.

1.6 تحلیل ویژگی‌ها

ویژگی‌های موجود در دیتاست مصنوعی مشابه دیتاست اصلی است، اما با تنوع و حجم بیشتری:

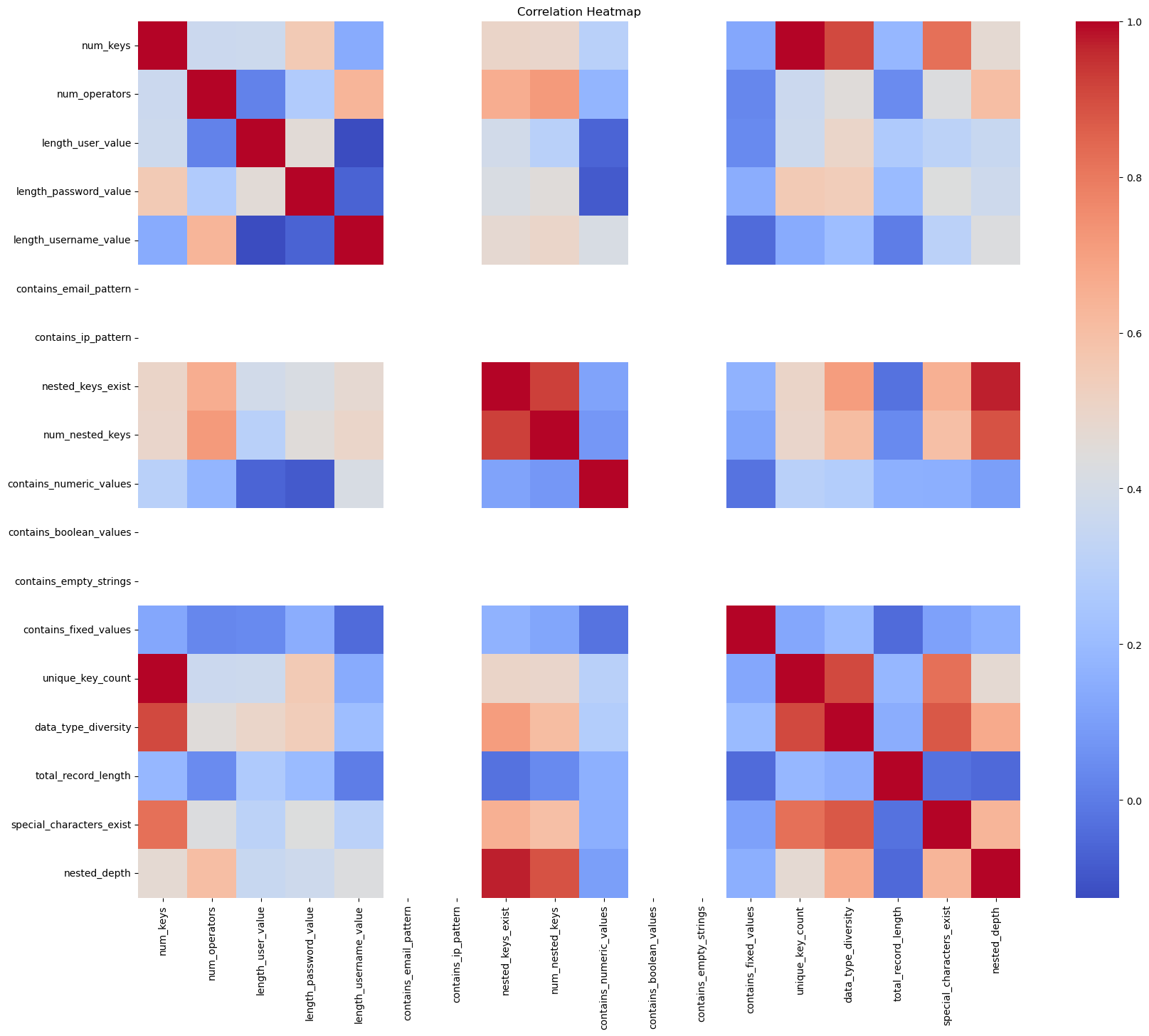
* num\_nested\_keys و nested\_depth: مقادیر تولیدشده شامل تنوع بیشتری در تعداد کلیدهای تودرتو و عمق آن‌ها است.
* contains\_email\_pattern و contains\_ip\_pattern: داده‌های مصنوعی شامل الگوهای ایمیل و IP بیشتری برای شبیه‌سازی شرایط واقعی‌تر است.
* Label: برچسب‌ها به‌صورت متوازن توزیع شده‌اند تا مدل بتواند تعمیم‌پذیری بهتری داشته باشد.

جدول 4- 2: تحلیل مقایسه‌ای دیتاست ها



|  |  |  |  |
| --- | --- | --- | --- |
| **ویژگی** | **دیتاست واقعی (Real-400)** | **دیتاست مصنوعی 1 (Syn-2000 / Seed=50% Real)** | **دیتاست مصنوعی 2 (Syn-2000 / Seed=All Real-400)** |
| **تعداد رکوردها** | 400 | 2,000 | 2,000 |
| **منشأ داده** | واقعی، برچسب‌گذاری‌شده | مصنوعی، تولیدشده با ChatGPT (GPT-5.2) با استفاده از ۵۰٪ داده واقعی به‌عنوان بذر/نمونه مرجع | مصنوعی، تولیدشده با ChatGPT (GPT-5.2) با استفاده از کل 400 رکورد واقعی به‌عنوان بذر/نمونه مرجع |
| **هدف ساخت** | مرجع پایه و معیار ارزیابی روی داده واقعی | افزایش تنوع با اتکا به بخشی از داده واقعی و کاهش هزینه داده‌سازی | بیشینه‌کردن شباهت ساختاری/الگویی به داده واقعی با استفاده از کل داده واقعی |
| **تنوع سناریوها** | محدود به داده موجود | متوسط تا بالا (وابسته به طراحی پرامپت و قیود) | بالا (به دلیل بذر کامل‌تر و پوشش الگوهای بیشتر) |
| **توزیع برچسب‌ها** | ممکن است نامتعادل باشد | قابل تنظیم (متعادل یا مشابه توزیع واقعی، بسته به روش تولید/پس‌پردازش) | قابل تنظیم (متعادل یا مشابه توزیع واقعی، بسته به روش تولید/پس‌پردازش) |
| **اعتبار برچسب** | بالا (Ground Truth) | متوسط (ریسک نویز/عدم‌قطعیت برچسب در داده مصنوعی) | متوسط (ریسک نویز/عدم‌قطعیت برچسب در داده مصنوعی) |
| **ریسک شکاف نسبت به داده واقعی (Fidelity Gap)** | کم | متوسط (به دلیل بذر ناقص) | کمتر از Syn-2000/50% (به دلیل بذر کامل‌تر) |
| **کاربرد پیشنهادی در پژوهش** | آزمون نهایی / سنجش تعمیم‌پذیری | آموزش یا آموزش+اعتبارسنجی در سناریوهای ترکیبی | آموزش یا آموزش+اعتبارسنجی؛ مناسب‌تر برای نزدیک‌سازی به الگوهای واقعی |
| **مزایا** | اصالت بالا؛ معیار قابل اتکا | افزایش حجم با هزینه کم؛ تنوع بیشتر نسبت به Real-400 | افزایش حجم + شباهت بیشتر به Real-400 (به‌واسطه بذر کامل‌تر) |
| **معایب/چالش‌ها** | حجم کم؛ پوشش محدود | ریسک نویز برچسب و کاهش فیدلیتی؛ نیاز به کنترل کیفیت | همچنان ریسک نویز برچسب و سوگیری مولد؛ نیاز به کنترل کیفیت |

در ادامه تحلیل ماتریس همبستگی در دو دیتاست را بررسی می‌کنیم:

****



شکل 4- 1: ماتریس همبستگی دیتا اصلی

تحلیل ماتریس همبستگی ویژگی‌ها

ماتریس همبستگی ارائه‌شده به‌صورت یک Heatmap، روابط میان ویژگی‌های مختلف در دیتاست مصنوعی را نشان می‌دهد. این ماتریس همبستگی خطی بین متغیرها را در بازه [−1,1]نمایش می‌دهد:

* 1+ → همبستگی مثبت کامل
* 0 → عدم همبستگی
* 1- → همبستگی منفی کامل

رنگ‌های گرم‌تر نشان‌دهنده همبستگی مثبت قوی‌تر و رنگ‌های سردتر بیانگر همبستگی ضعیف یا منفی هستند. بر اساس شکل، تحلیل روابط به شرح زیر است.

1. تحلیل روابط کلیدی

1.1 ویژگی‌های با همبستگی مثبت بالا

ساختارهای تودرتو (Nested Structure Features)

* nested\_keys\_exist ، num\_nested\_keys و nested\_depth
  + این سه ویژگی همبستگی مثبت بسیار قوی با یکدیگر دارند.
  + این موضوع نشان می‌دهد که وجود کلیدهای تودرتو معمولاً با افزایش تعداد آن‌ها و افزایش عمق ساختار داده همراه است.
  + این رابطه منطقی بوده و نشان‌دهنده سازگاری ساختار داده‌های مصنوعی با ساختار واقعی پایگاه‌های داده NoSQL است.

ویژگی‌های ساختاری رکورد

* unique\_key\_count ، data\_type\_diversity و special\_characters\_exist
  + بین این ویژگی‌ها همبستگی مثبت قابل توجهی مشاهده می‌شود.
  + رکوردهایی با تعداد کلیدهای بیشتر معمولاً دارای تنوع نوع داده و استفاده بیشتر از کاراکترهای خاص هستند.
  + این موضوع نشان‌دهنده افزایش پیچیدگی ساختار رکورد با افزایش تعداد ویژگی‌ها است.

ویژگی‌های طول فیلدها

* length\_user\_value ، length\_password\_value و length\_username\_value
  + همبستگی مثبت متوسط بین این ویژگی‌ها نشان می‌دهد که طول فیلدهای متنی مرتبط معمولاً رفتار مشابهی دارند.
  + این امر بیانگر الگوی تولید داده یکنواخت در مقادیر متنی است.

1.2 ویژگی‌های با همبستگی منفی یا ضعیف

* contains\_fixed\_values با بیشتر ویژگی‌های پیچیدگی داده
  + این ویژگی همبستگی منفی یا بسیار ضعیف با تنوع داده و طول رکورد دارد.
  + وجود مقادیر ثابت معمولاً نشان‌دهنده ساختار ساده‌تر رکوردها است.
* contains\_numeric\_values با برخی ویژگی‌های طول فیلد
  + در تصویر، همبستگی منفی یا بسیار کم با طول مقادیر متنی مشاهده می‌شود.
  + این نشان می‌دهد رکوردهای دارای مقادیر عددی الزاماً طول رشته بیشتری ندارند.

2. ویژگی‌های تقریباً مستقل

برخی ویژگی‌ها تقریباً بدون ارتباط خطی با سایر متغیرها هستند:

* contains\_email\_pattern
* contains\_ip\_pattern
* contains\_boolean\_values
* contains\_empty\_strings

استقلال این ویژگی‌ها نشان می‌دهد که این الگوها به‌صورت مستقل در داده تولید شده‌اند و وابستگی ساختاری مستقیمی با پیچیدگی رکورد ندارند. این موضوع باعث افزایش تنوع داده و جلوگیری از وابستگی مصنوعی بین ویژگی‌ها می‌شود.

3. خوشه‌بندی طبیعی ویژگی‌ها در ماتریس

بر اساس Heatmap می‌توان سه خوشه اصلی از ویژگی‌ها را مشاهده کرد:

1. ویژگی‌های ساختار کلیدها و پیچیدگی داده

(nested\_keys، nested\_depth، unique\_key\_count، data\_type\_diversity)

1. ویژگی‌های مربوط به طول مقادیر متنی

(length\_user\_value، length\_password\_value، length\_username\_value)

1. ویژگی‌های مربوط به الگوهای خاص داده

(email، ip، boolean، empty strings)

وجود این خوشه‌بندی نشان می‌دهد داده‌های مصنوعی ساختار منطقی و تفکیک‌پذیری مناسبی دارند.

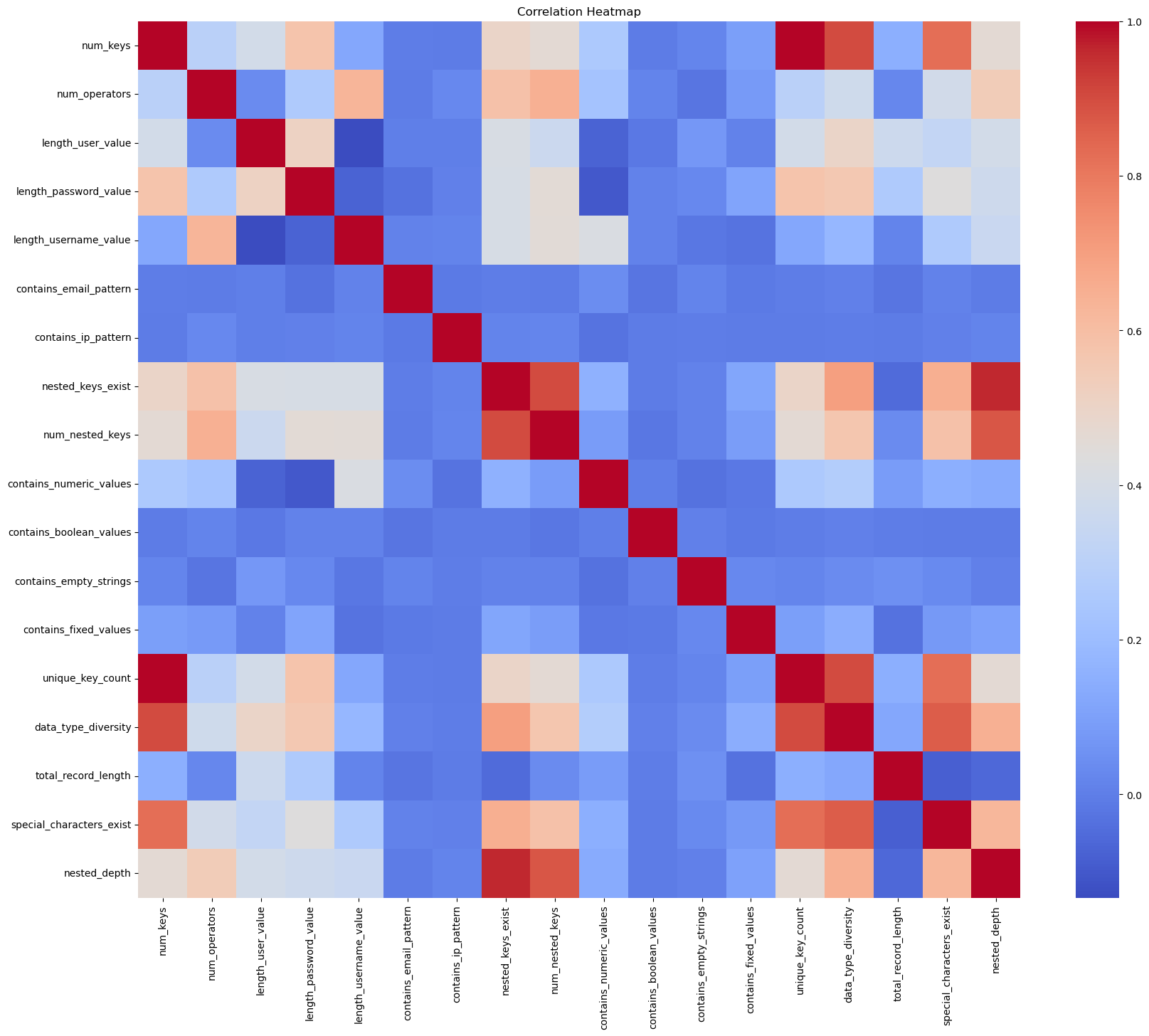
4.ارزیابی کیفیت داده‌های مصنوعی

نقاط قوت

* روابط منطقی بین ویژگی‌های ساختاری (nested و depth) به‌خوبی بازتاب داده شده‌اند.
* وجود خوشه‌های مشخص ویژگی‌ها نشان‌دهنده ساختار مناسب داده است.
* توزیع متعادل همبستگی‌ها نشان‌دهنده تنوع مناسب ویژگی‌ها است.

نقاط قابل بهبود

* برخی ویژگی‌ها تقریباً مستقل از سایر متغیرها هستند که ممکن است نشان‌دهنده ساده بودن فرآیند تولید داده برای این متغیرها باشد.
* می‌توان وابستگی‌های واقعی‌تری بین ویژگی‌های الگو (email، ip و ...) ایجاد کرد.



شکل 4- 2: ماتریس همبستگی دیتا مصنوعی

ماتریس همبستگی ارائه‌شده برای دیتاست مصنوعی، روابط خطی بین ویژگی‌های مختلف این مجموعه‌داده را نشان می‌دهد. این ماتریس به‌صورت گرافیکی در قالب Heatmap نمایش داده شده و میزان همبستگی قوی، متوسط و ضعیف میان ویژگی‌ها را مشخص می‌کند. مقادیر همبستگی در بازه [−1,1]قرار داشته و رنگ‌های گرم‌تر نشان‌دهنده همبستگی مثبت قوی‌تر و رنگ‌های سردتر نشان‌دهنده همبستگی ضعیف یا منفی هستند. در ادامه، روابط کلیدی میان ویژگی‌ها بررسی می‌شود.

1. تحلیل روابط کلیدی

1.1 ویژگی‌های با همبستگی مثبت بالا

ویژگی‌های ساختار تودرتو

* nested\_keys\_exist و num\_nested\_keys
  + همبستگی مثبت بسیار قوی بین این دو ویژگی مشاهده می‌شود.
  + این رابطه نشان می‌دهد که وجود کلیدهای تودرتو مستقیماً با تعداد کلیدهای تودرتو مرتبط است.
  + این ارتباط منطقی بوده و نشان‌دهنده سازگاری داده‌های مصنوعی با ساختار واقعی داده‌های NoSQL است.
* nested\_depth با nested\_keys\_exist و num\_nested\_keys
  + عمق تودرتویی داده‌ها نیز همبستگی مثبت قوی با تعداد و وجود کلیدهای تودرتو دارد.
  + این نتیجه نشان می‌دهد که با افزایش پیچیدگی ساختار داده، عمق ساختار نیز افزایش می‌یابد.

ویژگی‌های پیچیدگی ساختار داده

* unique\_key\_count و data\_type\_diversity
  + همبستگی مثبت بسیار قوی بین این دو ویژگی مشاهده می‌شود.
  + رکوردهایی با تعداد کلیدهای بیشتر معمولاً تنوع بیشتری در انواع داده دارند.
* special\_characters\_exist با data\_type\_diversity و unique\_key\_count
  + همبستگی مثبت نسبتاً بالا نشان می‌دهد که رکوردهای پیچیده‌تر از کاراکترهای خاص بیشتری استفاده می‌کنند.

ویژگی‌های طول مقادیر متنی

* length\_user\_value ، length\_password\_value و length\_username\_value
  + همبستگی مثبت متوسط بین این ویژگی‌ها مشاهده می‌شود.
  + این موضوع نشان‌دهنده الگوی مشابه در تولید داده‌های متنی در رکوردهای مصنوعی است.

1.2 ویژگی‌های با همبستگی منفی یا ضعیف

* contains\_fixed\_values با ویژگی‌های پیچیدگی داده
  + این ویژگی همبستگی ضعیف یا منفی با متغیرهایی مانند تنوع داده و طول رکورد دارد.
  + این موضوع نشان می‌دهد که وجود مقادیر ثابت معمولاً با کاهش پیچیدگی ساختار داده همراه است.
* contains\_numeric\_values با برخی ویژگی‌های متنی
  + همبستگی ضعیف یا منفی نشان می‌دهد که وجود مقادیر عددی الزاماً با افزایش طول داده‌های متنی همراه نیست.

1.3 ویژگی‌های مستقل

* contains\_email\_pattern و contains\_ip\_pattern
  + این ویژگی‌ها تقریباً با سایر متغیرها همبستگی بسیار کمی دارند.
  + این استقلال نشان می‌دهد که این الگوها به‌صورت مستقل در داده‌های مصنوعی تولید شده‌اند.
* contains\_empty\_strings و contains\_boolean\_values
  + همبستگی ضعیف با سایر ویژگی‌ها نشان می‌دهد که این مقادیر به‌صورت پراکنده و مستقل در رکوردها وجود دارند.

2. جمع‌بندی ساختار داده مصنوعی

نتایج نشان می‌دهد که داده‌های مصنوعی دارای ساختار منطقی و خوشه‌بندی مشخصی از ویژگی‌ها هستند:

* ویژگی‌های مرتبط با ساختار تودرتو دارای وابستگی قوی هستند.
* ویژگی‌های مرتبط با پیچیدگی رکورد با یکدیگر همبستگی مثبت دارند.
* ویژگی‌های مربوط به الگوهای خاص داده (email، ip، boolean و …) تقریباً مستقل از سایر ویژگی‌ها هستند.

این نتایج نشان می‌دهد که فرآیند تولید داده مصنوعی توانسته است روابط منطقی بین ویژگی‌ها را حفظ کرده و در عین حال تنوع مناسبی در ساختار داده ایجاد کند.

**4-3 مقایسه عملکرد مدل‌ها**

**4-3-1 عملکرد مدل‌های مختلف یادگیری ماشینی روی دیتاست اصلی**

در این بخش، عملکرد مجموعه‌ای از مدل‌های یادگیری ماشین (شامل مدل‌های پایه و مدل‌های یادگیری گروهی) روی دیتاست اصلی ارزیابی و با یکدیگر مقایسه می‌شود. ارزیابی به‌صورت دو مرحله‌ای انجام شده است: (1) اعتبارسنجی متقاطع طبقه‌بندی‌شده برای برآورد پایدار عملکرد، و (2) آزمون نهایی روی مجموعه آزمون برای تحلیل رفتار واقعی مدل‌ها در شرایط تعمیم. در ادامه، نتایج آزمایش‌ها به‌صورت مرحله‌به‌مرحله گزارش و تفسیر می‌گردد.

گام 1) روش ارزیابی و معیارهای سنجش

1. اعتبارسنجی متقاطع Stratified K-Fold (با 10 فولد)

به‌منظور کاهش وابستگی به یک تقسیم‌بندی خاص و حفظ نسبت کلاس‌ها در فولدهای آموزش/اعتبارسنجی، از StratifiedKFold استفاده شده است؛ این روش توزیع نسبی کلاس‌ها را در هر فولد حفظ می‌کند و برای مسائل طبقه‌بندی دودویی/چندکلاسه توصیه می‌شود.

1. محاسبه معیارهای عملکرد:

در هر فولد و همچنین در آزمون نهایی، معیارهای زیر محاسبه شده‌اند: Accuracy، ROC-AUC، Recall (macro)، Precision (macro)، F1 (macro)، Cohen’s Kappa و MCC

* ROC-AUC شاخصی برای سنجش توان تفکیک‌پذیری مدل بر اساس نمرات/احتمالات پیش‌بینی است و مستقل از یک آستانه ثابت تفسیر می‌شود.
* MCC معیار متوازن‌تری است که همزمان TP/TN/FP/FN را لحاظ می‌کند و حتی در شرایط عدم‌توازن نیز قابل اتکاست.
* Cohen’s Kappa میزان توافق مدل با برچسب‌های واقعی را با اصلاح توافق تصادفی می‌سنجد.

گام 2) نتایج اعتبارسنجی متقاطع (10- Fold CV) روی دیتاست اصلی

جدول 4-3 خلاصه نتایج آزمایش‌های CV را برای هر مدل نشان می‌دهد (میانگین و در صورت وجود انحراف معیار برای برخی شاخص‌ها). این جدول مبنای مقایسه “پایداری” و “میانگین عملکرد” مدل‌هاست.

جدول 4- 3: خلاصه نتایج اعتبارسنجی متقاطع 10-فولدی روی دیتاست اصلی

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | Accuracy (Mean±Std) | AUC (Mean±Std) | Recall | Precision | F1 | Kappa | MCC |
| Ensemble Model (Soft Voting) | 0.781 ± 0.036 | 0.820 ± 0.053 | 0.781 | 0.784 | 0.780 | 0.562 | 0.565 |
| درخت تصادفی | 0.764 ± 0.050 | 0.818 ± 0.052 | 0.764 | 0.769 | 0.763 | 0.528 | 0.533 |
| Extra Trees | 0.747 ± 0.049 | 0.814 ± 0.052 | 0.747 | 0.751 | 0.746 | 0.495 | 0.499 |
| LightGBM | 0.742 ± 0.056 | 0.811 ± 0.061 | 0.742 | 0.744 | 0.741 | 0.483 | 0.486 |
| KNN | 0.755 ± 0.048 | 0.808 ± 0.053 | 0.755 | 0.761 | 0.753 | 0.510 | 0.516 |
| Stacking Ensemble | 0.758 ± 0.055 | 0.806 ± 0.051 | 0.759 | 0.768 | 0.756 | 0.517 | 0.527 |
| XGBoost | 0.767 ± 0.053 | 0.792 ± 0.069 | 0.767 | 0.771 | 0.766 | 0.534 | 0.538 |
| Gradient Boosting | 0.747 ± 0.050 | 0.786 ± 0.057 | 0.747 | 0.752 | 0.746 | 0.494 | 0.500 |
| AdaBoost | 0.672 ± 0.079 | 0.699 ± 0.074 | 0.672 | 0.678 | 0.668 | 0.343 | 0.349 |
| درخت تصمیم | 0.657 ± 0.069 | 0.711 ± 0.078 | 0.657 | 0.667 | 0.651 | 0.314 | 0.324 |
| SVM | 0.599 ± 0.090 | 0.649 ± 0.100 | 0.599 | 0.610 | 0.586 | 0.198 | 0.208 |
| رگرسیون خطی | 0.588 ± 0.074 | 0.644 ± 0.094 | 0.588 | 0.591 | 0.582 | 0.175 | 0.179 |

تفسیر نتایج :

* در نتایج آزمایش‌های CV، مدل Soft Voting بالاترین Accuracy و ROC-AUC را ثبت کرده است. این امر نشان می‌دهد ترکیب چند مدل و تجمیع احتمال‌ها می‌تواند به بهبود پایداری و کاهش واریانس منجر شود (به‌ویژه در داده‌های محدود).
* مدل‌های درختی-تجمیعی مانند درخت تصادفی و Extra Trees نیز عملکرد نزدیک و پایدار ارائه کرده‌اند.
* مدل‌های ساده‌تر مانند رگرسیون خطی و SVM در این مسئله (با الگوهای احتمالاً غیرخطی) در سطح پایین‌تری قرار گرفته‌اند.
* انتخاب StratifiedKFold باعث می‌شود مقایسه مدل‌ها تحت حفظ نسبت کلاس‌ها منصفانه‌تر باشد.

گام 3) نتایج آزمون نهایی روی Test Set (دیتاست اصلی)

پس از اتمام اعتبارسنجی متقاطع، هر مدل روی داده‌های آموزشی آموزش داده شد و روی مجموعه آزمون ارزیابی گردید. جدول 4-4 نتایج آزمون نهایی را نشان می‌دهد.

جدول 4- 4: نتایج آزمون نهایی مدل‌ها روی دیتاست اصلی

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | Accuracy | AUC | Precision | Recall | F1 | Kappa | MCC |
| درخت تصادفی | 0.6750 | 0.6695 | 0.6719 | 0.6667 | 0.6675 | 0.3367 | 0.3385 |
| Gradient Boosting | 0.6250 | 0.6790 | 0.6313 | 0.6313 | 0.6250 | 0.2574 | 0.2626 |
| Extra Trees | 0.5875 | 0.6859 | 0.5918 | 0.5922 | 0.5874 | 0.1811 | 0.1839 |
| KNN | 0.5625 | 0.5508 | 0.5777 | 0.5745 | 0.5608 | 0.1443 | 0.1521 |
| SVM | 0.5500 | 0.4984 | 0.2750 | 0.5000 | 0.3548 | 0.0000 | 0.0000 |
| XGBoost | 0.5375 | 0.5464 | 0.4733 | 0.4937 | 0.3933 | -0.0137 | -0.0260 |
| AdaBoost | 0.5250 | 0.6237 | 0.4922 | 0.4949 | 0.4587 | -0.0106 | -0.0126 |
| Ensemble Model (Soft Voting) | 0.5000 | 0.5988 | 0.4437 | 0.4672 | 0.4182 | -0.0695 | -0.0860 |
| LightGBM | 0.5000 | 0.5477 | 0.4667 | 0.4747 | 0.4505 | -0.0526 | -0.0580 |
| درخت تصمیم | 0.4625 | 0.5215 | 0.7278 | 0.5114 | 0.3353 | 0.0205 | 0.1018 |
| رگرسیون خطی | 0.4500 | 0.6338 | 0.2250 | 0.5000 | 0.3103 | 0.0000 | 0.0000 |
| Stacking Ensemble | 0.4250 | 0.4006 | 0.3777 | 0.4015 | 0.3780 | -0.2042 | -0.2195 |

تفسیر نتایج Test با تکیه بر نتایج آزمایش:

* از منظر معیارهای “متعادل” (به‌ویژه MCC و Kappa) و نیز Accuracy/F1، درخت تصادفی بهترین نتیجه را روی آزمون نهایی ارائه کرده است. این معیارها به دلیل لحاظ کردن همه درایه‌های ماتریس درهم‌ریختگی، تصویر دقیق‌تری از کیفیت طبقه‌بندی ارائه می‌دهند.
* از نظر ROC-AUC، مدل Extra Trees بالاترین مقدار را دارد. تفاوت بین «AUC بالاتر» و «Accuracy پایین‌تر» می‌تواند ناشی از این باشد که AUC توان رتبه‌بندی/تفکیک را می‌سنجد و الزاماً بهترین عملکرد را در آستانه تصمیم ثابت (مثل 0.5) تضمین نمی‌کند.
* مشاهده اختلاف میان بهترین مدل در CV (Soft Voting) و بهترین مدل در Test (درخت تصادفی) نشان‌دهنده اهمیت گزارش هر دو سطح ارزیابی است؛ در داده‌های کوچک، نوسان نمونه‌ها و حساسیت به تقسیم Train/Test می‌تواند باعث تغییر رتبه مدل‌ها شود.

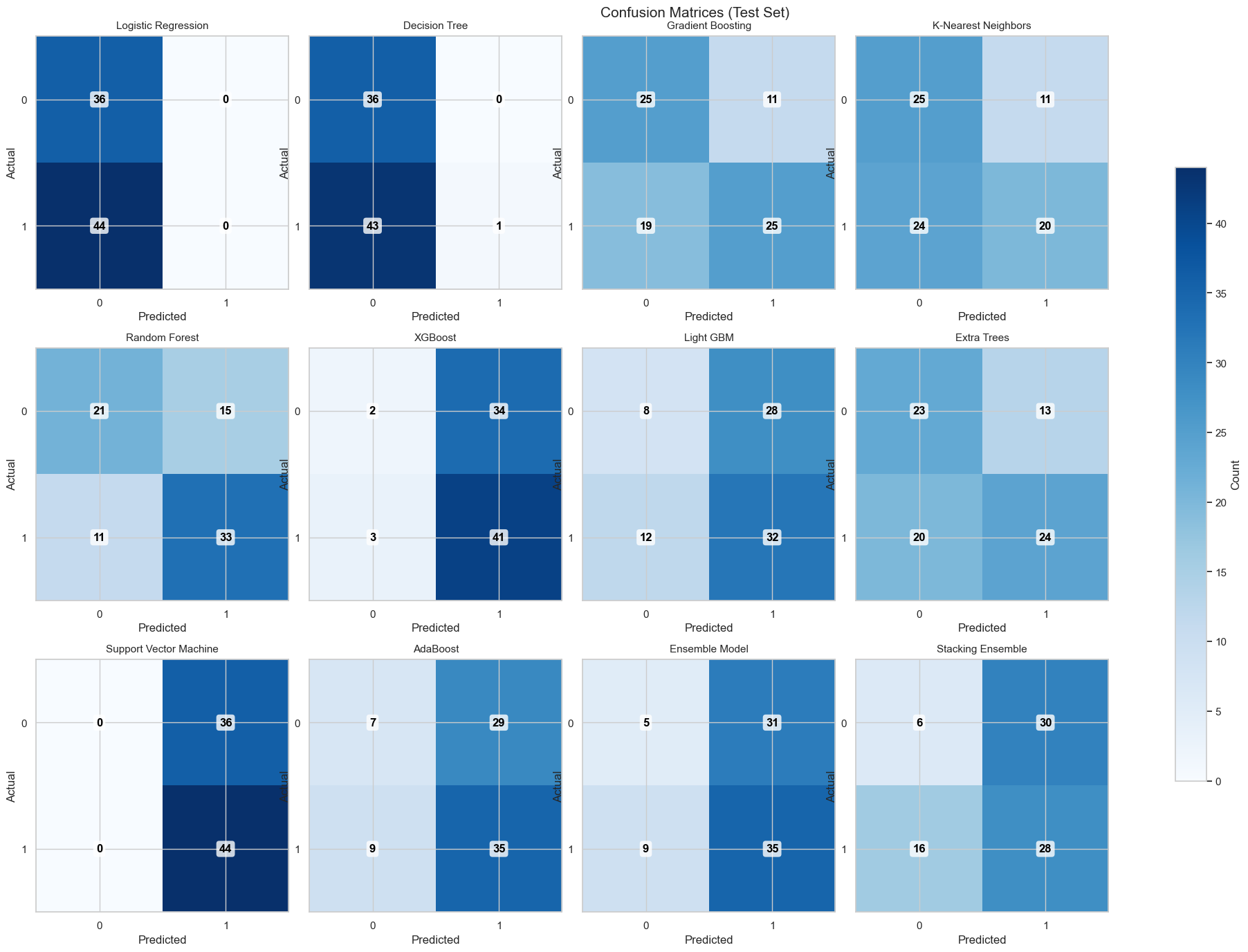
گام 4) تحلیل رفتاری مدل‌ها با ماتریس درهم‌ریختگی (Confusion Matrix)

برای تحلیل دقیق‌تر، ماتریس‌های درهم‌ریختگی مجموعه آزمون (مطابق شکل ارائه‌شده) نشان می‌دهد هر مدل چگونه بین False Positive و False Negative مبادله برقرار کرده است. این تحلیل برای مسئله امنیتی بسیار مهم است، زیرا FN (عدم تشخیص حمله) معمولاً هزینه بالاتری دارد.

خلاصه چهار مؤلفه (TN, FP, FN, TP) برای هر مدل بر اساس نتایج آزمایش:

جدول 4- 5: چهار مؤلفه (TN, FP, FN, TP) برای هر مدل

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| مدل | TN | FP | FN | TP |
| درخت تصادفی | 21 | 15 | 11 | 33 |
| Gradient Boosting | 25 | 11 | 19 | 25 |
| Extra Trees | 23 | 13 | 20 | 24 |
| KNN | 25 | 11 | 24 | 20 |
| XGBoost | 2 | 34 | 3 | 41 |
| LightGBM | 8 | 28 | 12 | 32 |
| Ensemble (Soft Voting) | 5 | 31 | 9 | 35 |
| Stacking Ensemble | 6 | 30 | 16 | 28 |
| AdaBoost | 7 | 29 | 9 | 35 |
| درخت تصمیم | 36 | 0 | 43 | 1 |
| رگرسیون خطی | 36 | 0 | 44 | 0 |
| SVM | 0 | 36 | 0 | 44 |



شکل 4- 1: ماتریس درهم ریختگی مدلها

جمع‌بندی تحلیلی Confusion Matrix:

* درخت تصمیم با FP≈0 (Specificity بالا) اما FN بسیار زیاد، عملاً بسیاری از حملات را از دست می‌دهد (ریسک عملیاتی بالا).
* SVM رفتار کاملاً معکوس دارد (تقریباً همه نمونه‌ها را حمله تشخیص می‌دهد) که FP بسیار بالا ایجاد می‌کند و برای بهره‌برداری عملی می‌تواند منجر به هشدارهای کاذب گسترده شود.
* درخت تصادفی در نتایج آزمایش، نسبتاً متعادل‌تر عمل کرده و همین امر با مقادیر بهتر MCC/Kappa هم‌راستا است (معیارهای متوازن بر پایه کل ماتریس)

بر اساس نتایج آزمایش روی دیتاست اصلی:

1. در سطح اعتبارسنجی متقاطع، مدل Soft Voting بهترین میانگین عملکرد را نشان داده است.
2. در سطح آزمون نهایی، درخت تصادفی بهترین نتیجه را از منظر Accuracy/F1 و به‌ویژه معیارهای متوازن MCC/Kappa ارائه کرده است.
3. تحلیل Confusion Matrix نشان داد برخی مدل‌ها به سمت کاهش FP با افزایش FN یا بالعکس متمایل هستند؛ در کاربرد امنیتی، این مبادله باید متناسب با هزینه خطاها مدیریت شود (مثلاً با تنظیم آستانه تصمیم یا راهبردهای هزینه‌محور).

### 4-3-2 عملکرد روی Syn-2000 (Seed = 50% Real)

در این بخش، عملکرد مدل‌ها روی دیتاست Syn-2000 گزارش می‌شود؛ این دیتاست مصنوعی با استفاده از نیمی از داده واقعی (200 رکورد به‌عنوان Seed) تولید شده و سپس برای ارزیابی، مطابق رویه آزمایش، به دو بخش آموزش و آزمون تفکیک گردیده است (در خروجی آزمون، 400 نمونه به‌عنوان Test دیده می‌شود؛ یعنی سناریوی آزمون 20% از 2000 رکورد)

ارزیابی مدل‌ها در این سناریو بر مبنای نتایج آزمایش روی مجموعه آزمون و با اتکا به ماتریس‌های درهم‌ریختگی (Confusion Matrices) انجام شده است. در یک مسئله امنیتی، تحلیل Confusion Matrix اهمیت ویژه‌ای دارد، زیرا:

* FN (False Negative) به معنی «حمله‌ای که تشخیص داده نشده» و از نظر ریسک، معمولاً پرهزینه‌ترین خطاست.
* FP (False Positive) به معنی «هشدار کاذب» است که هزینه عملیاتی و نویز هشدارها را افزایش می‌دهد.

گام 1) استخراج مؤلفه‌های Confusion Matrix

برای هر مدل، چهار مؤلفه زیر از نتایج آزمون استخراج شده است:

* TN: نمونه‌های غیرحمله که درست غیرحمله تشخیص داده شدند
* FP: غیرحمله‌هایی که اشتباهاً حمله تشخیص داده شدند
* FN: حمله‌هایی که اشتباهاً غیرحمله تشخیص داده شدند
* TP: حمله‌هایی که درست حمله تشخیص داده شدند

جدول 4- 6: خلاصه ماتریس درهم‌ریختگی مدل‌ها روی Syn-2000 (Test = 400)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| مدل | TN | FP | FN | TP |
| رگرسیون خطی | 65 | 110 | 57 | 168 |
| درخت تصمیم | 124 | 51 | 57 | 168 |
| Gradient Boosting | 122 | 53 | 34 | 191 |
| نزدیکترین همسایه | 158 | 17 | 18 | 207 |
| درخت تصادفی | 123 | 52 | 33 | 192 |
| XGBoost | 133 | 42 | 27 | 198 |
| LightGBM | 142 | 33 | 25 | 200 |
| Extra Trees | 99 | 76 | 26 | 199 |
| Support Vector Machine | 83 | 92 | 60 | 165 |
| AdaBoost | 104 | 71 | 35 | 190 |
| Ensemble Model (Soft Voting) | 150 | 25 | 18 | 207 |
| Stacking Ensemble | 155 | 20 | 24 | 201 |



شکل 4- 2: کنفیوژن ماتریس مدلها با 2000 دیتا مصنوعی

گام 2) محاسبه معیارهای عملکرد (بر مبنای نتایج آزمون)

برای مقایسه کمی مدل‌ها، معیارهای زیر از Confusion Matrix محاسبه و گزارش شده‌اند:

* Accuracy: نسبت کل پیش‌بینی‌های صحیح
* Precision (macro) و Recall (macro) و F1 (macro): میانگین عملکرد روی هر دو کلاس (حمله/غیرحمله)
* Precision\_pos / Recall\_pos / F1\_pos: عملکرد روی کلاس «حمله» (کلاس مثبت)
* Specificity: توانایی مدل در شناسایی «غیرحمله» (کلاس منفی)

جدول 4- 7: نتایج عملکرد مدل‌ها روی Syn-2000 (Test)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | Accuracy | Precision (macro) | Recall (macro) | F1 (macro) | Precision (Attack) | Recall (Attack) | F1 (Attack) | Specificity | FP | FN |
| نزدیکترین همسایه | 0.9125 | 0.9109 | 0.9114 | 0.9112 | 0.9241 | 0.9200 | 0.9220 | 0.9029 | 17 | 18 |
| Ensemble Model (Soft Voting) | 0.8925 | 0.8925 | 0.8886 | 0.8903 | 0.8922 | 0.9200 | 0.9059 | 0.8571 | 25 | 18 |
| Stacking Ensemble | 0.8900 | 0.8877 | 0.8895 | 0.8885 | 0.9095 | 0.8933 | 0.9013 | 0.8857 | 20 | 24 |
| LightGBM | 0.8550 | 0.8543 | 0.8502 | 0.8519 | 0.8584 | 0.8889 | 0.8734 | 0.8114 | 33 | 25 |
| XGBoost | 0.8275 | 0.8281 | 0.8200 | 0.8228 | 0.8250 | 0.8800 | 0.8516 | 0.7600 | 42 | 27 |
| درخت تصادفی | 0.7875 | 0.7877 | 0.7781 | 0.7810 | 0.7869 | 0.8533 | 0.8188 | 0.7029 | 52 | 33 |
| Gradient Boosting | 0.7825 | 0.7824 | 0.7730 | 0.7758 | 0.7828 | 0.8489 | 0.8145 | 0.6971 | 53 | 34 |
| Extra Trees | 0.7450 | 0.7578 | 0.7251 | 0.7280 | 0.7236 | 0.8844 | 0.7960 | 0.5657 | 76 | 26 |
| AdaBoost | 0.7350 | 0.7381 | 0.7194 | 0.7222 | 0.7280 | 0.8444 | 0.7819 | 0.5943 | 71 | 35 |
| درخت تصمیم | 0.7300 | 0.7261 | 0.7276 | 0.7267 | 0.7671 | 0.7467 | 0.7568 | 0.7086 | 51 | 57 |
| Support Vector Machine | 0.6200 | 0.6112 | 0.6038 | 0.6033 | 0.6420 | 0.7333 | 0.6846 | 0.4743 | 92 | 60 |
| رگرسیون خطی | 0.5825 | 0.5686 | 0.5590 | 0.5529 | 0.6043 | 0.7467 | 0.6680 | 0.3714 | 110 | 57 |

گام 3) تحلیل و مقایسه آکادمیک نتایج (بر اساس آزمون)

1) بهترین مدل از نظر عملکرد کلی (Accuracy و F1 macro)

* KNN با Accuracy=0.9125 و F1(macro)=0.9112 بهترین عملکرد کلی را ثبت کرده است.
* این نتیجه همراه با FP بسیار پایین (17) و FN پایین (18) نشان می‌دهد KNN در این سناریو همزمان توانسته است *هشدار کاذب* را کنترل کند و *حملات از دست‌رفته* را نیز حداقل نگه دارد.

2) بهترین مدل‌ها از نظر تشخیص حمله (Recall روی کلاس Attack)

* KNN و Ensemble (Soft Voting) هر دو Recall(Attack)=0.92 دارند؛ یعنی در نتایج آزمایش، حدود 92% حملات در آزمون شناسایی شده‌اند.
* با این حال، Soft Voting در مقایسه با KNN FP بیشتری تولید کرده است (25 در برابر 17). بنابراین KNN از نظر عملیاتی (کاهش نویز هشدار) گزینه مناسب‌تری است.

3) عملکرد مدل‌های تقویتی/درختی پیشرفته (LightGBM و XGBoost)

* LightGBM با Accuracy=0.855 و F1(Attack)=0.8734 عملکرد قابل توجهی داشته است و نسبت به XGBoost، هم FP و هم FN را کمتر نگه داشته است (FP=33, FN=25).
* XGBoost نیز قوی است (Accuracy=0.8275)، اما FP بیشتری نسبت به LightGBM دارد (42).

4)علت افت مدل‌های ساده‌تر در این سناریو:

* رگرسیون خطی در آزمون دارای FP بسیار بالا (110) و Specificity پایین (0.3714) است؛ یعنی تمایل به برچسب‌زدن «حمله» برای تعداد زیادی از نمونه‌های غیرحمله دارد.
* SVM نیز FP و FN بالایی ثبت کرده (FP=92, FN=60) که نشان می‌دهد در این تنظیمات، مرز تصمیم مناسبی برای داده‌های این سناریو شکل نگرفته است.

جمع‌بندی

بر اساس نتایج آزمایش روی Syn-2000 (Seed=50% Real)

* KNN بهترین مدل از نظر توازن کلی خطاها و بیشترین دقت بوده است (هم FP و هم FN پایین)
* Ensemble (Soft Voting) و Stacking نیز عملکرد رقابتی دارند، اما در این آزمون، یا FP بالاتری نسبت به KNN داشته‌اند یا FN بیشتری ایجاد کرده‌اند.
* در تحلیل امنیتی، مدلی مطلوب‌تر است که FN را پایین نگه دارد و در عین حال FP را کنترل کند؛ در این سناریو، KNN بهترین trade-off را نشان داده است.

### 4-3-3 عملکرد روی Syn-2000 (Seed = All Real-400)

در این سناریو، مجموعه‌داده مصنوعی Syn-2000 با استفاده از کل دیتاست واقعی (400 رکورد) به‌عنوان Seed تولید شده است. سپس مطابق طراحی آزمایش، داده‌ها به آموزش/آزمون تفکیک شده‌اند؛ با توجه به مجموع مقادیر هر ماتریس درهم‌ریختگی، اندازه مجموعه آزمون در این سناریو 320 نمونه است. ارزیابی عملکرد مدل‌ها بر پایه نتایج آزمون و تحلیل دقیق Confusion Matrix انجام می‌شود، زیرا در مسائل امنیتی، توازن بین False Negative (حملهِ کشف‌نشده) و False Positive (هشدار کاذب) تعیین‌کننده کیفیت عملیاتی سامانه است.

گام 1) استخراج مؤلفه‌های ماتریس درهم‌ریختگی

چهار مؤلفه استاندارد عبارت‌اند از:

TN (غیرحمله درست)، FP (هشدار کاذب)، FN (حمله از دست‌رفته)، TP (حمله درست).

جدول 4- 8: خلاصه ماتریس‌های درهم‌ریختگی مدل‌ها روی Syn-2000 (Seed=All Real-400) در مجموعه آزمون

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| مدل | TN | FP | FN | TP | N(Test) |
| رگرسیون خطی | 81 | 67 | 71 | 101 | 320 |
| درخت تصمیم | 113 | 35 | 58 | 114 | 320 |
| Gradient Boosting | 106 | 42 | 30 | 142 | 320 |
| نزدیکترین همسایه | 113 | 35 | 44 | 128 | 320 |
| درخت تصادفی | 113 | 35 | 30 | 142 | 320 |
| XGBoost | 116 | 32 | 19 | 153 | 320 |
| LightGBM | 116 | 32 | 17 | 155 | 320 |
| Extra Trees | 103 | 45 | 38 | 134 | 320 |
| Support Vector Machine | 103 | 45 | 56 | 116 | 320 |
| AdaBoost | 108 | 40 | 61 | 111 | 320 |
| Ensemble Model (Soft Voting) | 118 | 30 | 26 | 146 | 320 |
| Stacking Ensemble | 110 | 38 | 14 | 158 | 320 |

گام 2) محاسبه شاخص‌های عملکرد روی مجموعه آزمون

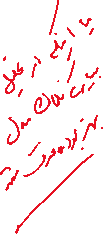
برای مقایسه کمی، شاخص‌های زیر از Confusion Matrix استخراج شده‌اند:

* Accuracy: سهم پیش‌بینی‌های صحیح از کل نمونه‌ها
* Precision/Recall/F1 (macro): میانگین عملکرد روی هر دو کلاس (حمله و غیرحمله)
* Precision/Recall/F1 برای کلاس حمله (Attack): شاخص‌های حساس برای ارزیابی توان تشخیص حمله
* Specificity: توان مدل در شناسایی صحیح «غیرحمله» (کنترل هشدار کاذب)  
  این نوع تحلیل با چارچوب متریک‌های طبقه‌بندی در scikit-learn هم‌راستا است.

جدول 4- 9: نتایج عملکرد مدل‌ها روی Syn-2000 (Seed=All Real-400) در مجموعه آزمون

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | Accuracy | Precision (macro) | Recall (macro) | F1 (macro) | Precision (Attack) | Recall (Attack) | F1 (Attack) | Specificity | FP | FN |
| LightGBM | 0.8469 | 0.8505 | 0.8425 | 0.8446 | 0.8289 | 0.9012 | 0.8635 | 0.7838 | 32 | 17 |
| XGBoost | 0.8406 | 0.8431 | 0.8367 | 0.8385 | 0.8270 | 0.8895 | 0.8571 | 0.7838 | 32 | 19 |
| Stacking Ensemble | 0.8375 | 0.8466 | 0.8309 | 0.8338 | 0.8061 | 0.9186 | 0.8587 | 0.7432 | 38 | 14 |
| Ensemble Model (Soft Voting) | 0.8250 | 0.8245 | 0.8231 | 0.8237 | 0.8295 | 0.8488 | 0.8391 | 0.7973 | 30 | 26 |
| درخت تصادفی | 0.7969 | 0.7962 | 0.7945 | 0.7952 | 0.8023 | 0.8256 | 0.8138 | 0.7635 | 35 | 30 |
| Gradient Boosting | 0.7750 | 0.7756 | 0.7709 | 0.7721 | 0.7717 | 0.8256 | 0.7978 | 0.7162 | 42 | 30 |
| نزدیکترین همسایه | 0.7531 | 0.7525 | 0.7539 | 0.7526 | 0.7853 | 0.7442 | 0.7642 | 0.7635 | 35 | 44 |
| Extra Trees | 0.7406 | 0.7396 | 0.7375 | 0.7382 | 0.7486 | 0.7791 | 0.7635 | 0.6959 | 45 | 38 |
| درخت تصمیم | 0.7094 | 0.7130 | 0.7132 | 0.7094 | 0.7651 | 0.6628 | 0.7103 | 0.7635 | 35 | 58 |
| Support Vector Machine | 0.6844 | 0.6841 | 0.6852 | 0.6839 | 0.7205 | 0.6744 | 0.6967 | 0.6959 | 45 | 56 |
| AdaBoost | 0.6844 | 0.6871 | 0.6875 | 0.6843 | 0.7351 | 0.6453 | 0.6873 | 0.7297 | 40 | 61 |
| رگرسیون خطی | 0.5688 | 0.5670 | 0.5673 | 0.5671 | 0.6012 | 0.5872 | 0.5941 | 0.5473 | 67 | 71 |

گام 3) تحلیل آکادمیک نتایج (مبتنی بر آزمون)



* 1. بهترین عملکرد کلی (Accuracy و F1 macro)

بر اساس نتایج آزمون، LightGBM بالاترین Accuracy و F1(macro) را ثبت کرده است. این موضوع نشان می‌دهد داده مصنوعی تولیدشده با Seed کاملِ واقعی، برای LightGBM ساختاری ایجاد کرده که هم در تشخیص حمله و هم در کنترل خطاهای کلاس غیرحمله، تعادل مناسبی برقرار می‌شود.

2) حساسیت به حمله (Recall برای کلاس Attack) و کنترل حمله‌های از دست‌رفته (FN)

* کمترین FN مربوط به Stacking Ensemble است (FN=14) و در نتیجه بالاترین Recall(Attack)=0.9186 را دارد. از منظر امنیتی، کاهش FN اهمیت ویژه‌ای دارد چون «حمله‌ای که تشخیص داده نشود» مستقیماً ریسک نفوذ را افزایش می‌دهد.
* با این حال، Stacking در مقایسه با LightGBM/XGBoost، FP بالاتری دارد (38 در برابر 32) و این به معنی افزایش هشدارهای کاذب و هزینه عملیاتی بالاتر است.

3)بهترین توازن عملیاتی بین FP و FN

* LightGBM توازن بسیار خوبی ایجاد کرده است: FN پایین (17) در کنار FP کنترل‌شده (32).
* Soft Voting نیز Specificity بالاتری دارد (0.7973) و FP کمتری ثبت کرده (30) اما FN آن از LightGBM بیشتر است (26). در سامانه‌های امنیتی، ترجیح بین این دو به سیاست سازمان وابسته است: اگر هزینه FN بالاتر باشد، LightGBM ارجح است؛ اگر کنترل FP اولویت داشته باشد، Soft Voting می‌تواند گزینه قابل دفاعی باشد.

4) رفتار مدل‌های ساده‌تر

* رگرسیون خطی در این سناریو ضعیف‌ترین عملکرد کلی را دارد (Accuracy≈0.569) و همزمان FP و FN نسبتاً بالا ثبت کرده است. این الگو نشان می‌دهد مرز تصمیم خطی برای داده‌های این مسئله کافی نبوده و مدل در جداسازی الگوهای حمله/غیرحمله محدودیت دارد.

جمع‌بندی

بر اساس نتایج آزمایش روی Syn-2000 (Seed = All Real-400):

* LightGBM بهترین عملکرد کلی را ارائه کرده (بالاترین Accuracy و F1 macro) و در عین حال FN و FP را در سطح متعادل نگه داشته است.
* Stacking Ensemble کمترین FN را دارد و برای سناریوهایی که «عدم کشف حمله» پرهزینه‌ترین خطا است، گزینه بسیار جدی محسوب می‌شود؛ هرچند با افزایش FP همراه است.
* Soft Voting FP کمتری دارد و از منظر کنترل هشدار کاذب بهتر است، اما با افزایش FN نسبت به LightGBM همراه می‌شود.

### 4-3-4 مقایسه نهایی سه سناریو (Real-400 vs Syn-2000(50%) vs Syn-2000(100%))

در این بخش، نتایج سه سناریوی اصلی آزمایش به‌صورت یکپارچه مقایسه می‌شوند تا مشخص شود تغییر منبع/کیفیت داده (واقعی محدود در برابر داده مصنوعی با Seed متفاوت) چگونه باعث جابجایی رتبه مدل‌ها و تغییر الگوی خطا (FP/FN) می‌شود. از آنجا که ارزیابی نهایی در هر سناریو بر پایه مجموعه آزمون مستقل انجام شده است، این مقایسه تصویر روشن‌تری از توان تعمیم و ریسک عملیاتی (به‌ویژه FN در امنیت) ارائه می‌دهد. همچنین در تفسیر نتایج باید توجه داشت که در داده‌های کوچک، نوسان ناشی از نمونه‌گیری و تقسیم‌بندی می‌تواند محسوس باشد؛ به همین دلیل استفاده از Cross-Validation در مرحله ارزیابی میانی رایج و توصیه‌شده است.

جدول 4- 10: جدول تجمیعی «بهترین مدل هر سناریو» بر اساس نتایج آزمون - معیار انتخاب بهترین مدل : Accuracy و F1(macro) به‌عنوان شاخص عملکرد کلی، و FN/FP به‌عنوان شاخص‌های ریسک امنیتی/هشدار کاذب.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| سناریو | اندازه آزمون | بهترین مدل از نظر عملکرد کلی | Accuracy | F1 (macro) | FP | FN | تفسیر ریسک |
| Real-400 | 80 | درخت تصادفی | 0.6750 | 0.6675 | 15 | 11 | توازن نسبتاً مناسب؛ اما به‌علت کمبود داده، عملکرد کلی محدود است. |
| Syn-2000 (Seed=50% Real) | 400 | KNN | 0.9125 | 0.9112 | 17 | 18 | بهترین عملکرد کلی؛ همزمان FP پایین و FN پایین‌تر از اکثر مدل‌ها. |
| Syn-2000 (Seed=All Real-400) | 320 | LightGBM | 0.8469 | 0.8446 | 32 | 17 | بهترین مدل کلی در این سناریو؛ FN پایین‌تر و F1 متوازن‌تر. |

نکته امنیتی تکمیلی: اگر معیار تصمیم‌گیری سازمان «حداقل‌سازی FN» باشد (یعنی از دست نرفتن حمله حتی به قیمت افزایش FP)، در سناریوی Syn-2000(100%) مدل Stacking کمترین FN را نشان داده است (FN=14)؛ ولی FP بالاتری نسبت به LightGBM دارد. (این انتخاب، تصمیم سیاستی/عملیاتی است.)

تحلیل علت تغییر رتبه‌ها (چرا بهترین مدل‌ها بین سناریوها عوض می‌شوند؟)

1) اثر اندازه نمونه و کاهش واریانس ارزیابی

* در سناریوی Real-400 تعداد داده محدود است؛ در چنین شرایطی، نوسان نمونه‌گیری و حساسیت به تقسیم Train/Test زیاد می‌شود و مدل‌هایی که واریانس کمتر دارند یا به‌طور طبیعی Robust هستند (مثل درخت تصادفی) معمولاً پایدارتر ظاهر می‌شوند. استفاده از Cross-Validation دقیقاً برای کاهش همین وابستگی به یک تقسیم‌بندی و بهبود برآورد عملکرد توصیه می‌شود.
* با بزرگ شدن داده (Syn-2000)، ارزیابی روی آزمون بزرگ‌تر انجام می‌شود و مقایسه مدل‌ها «کم‌نوسان‌تر» و قابل اتکاتر می‌شود.

2) نقش «فیدلیتی» و «Utility» داده مصنوعی در تغییر رفتار مدل‌ها

* کیفیت داده مصنوعی فقط به تعداد رکورد وابسته نیست؛ میزان شباهت توزیعی/ساختاری به داده واقعی (Fidelity) تعیین می‌کند که داده مصنوعی چقدر برای آموزش و ارزیابی مفید است (Utility). پژوهش‌های اخیر نشان می‌دهند Utility می‌تواند حتی با فاصله توزیعی مشخص تغییر کند و باید آن را به‌صورت task-based (یعنی با عملکرد مدل روی وظیفه) سنجید.
* در Syn-2000(50%) داده مصنوعی با Seed نصف داده واقعی ساخته شده است. اگر این تولید باعث تراکم مناسب نمونه‌ها و پوشش بهتر نواحی تصمیم شده باشد، مدل‌های «نمونه‌محور» مثل KNN می‌توانند جهش عملکردی داشته باشند.

3) چرا KNN در Syn-2000(50%) بهترین شد؟

* KNN یک روش instance-based / lazy learning است و کیفیت آن شدیداً به تراکم نمونه‌ها و «قابل تفکیک بودن» نواحی محلی داده وابسته است. وقتی داده بیشتر و متوازن‌تر می‌شود، KNN معمولاً تصمیم‌گیری محلی دقیق‌تری دارد و خطا کاهش می‌یابد.
* در نتایج شما، KNN همزمان FP پایین و FN پایین دارد؛ این الگو معمولاً زمانی رخ می‌دهد که داده آموزشی، هم پوشش کافی از نمونه‌های حمله و هم از غیرحمله فراهم کرده باشد (یعنی مرزهای محلی بهتر شکل گرفته‌اند).

4) چرا LightGBM در Syn-2000(100%) بهترین شد؟

* LightGBM از خانواده Gradient Boosting Tree است و به‌طور خاص برای مدل‌سازی روابط غیرخطی و تعامل ویژگی‌ها بسیار توانمند است؛ با افزایش داده (و در بسیاری مسائل، با افزایش کیفیت/تنوع داده)، مدل‌های boosting معمولاً بهتر می‌توانند «ساختار واقعی مسئله» را یاد بگیرند و تعمیم دهند. نسخه LightGBM نیز به دلیل طراحی کارا (از جمله روش‌های بهینه‌سازی خاص) در مقیاس بزرگ/متوسط عملکرد خوبی دارد.
* تفاوت Syn-2000(100%) با Syn-2000(50%) این است که Seed کامل واقعی می‌تواند باعث شود داده مصنوعی از نظر الگوهای حمله/غیرحمله نماینده‌تر باشد (Fidelity بالاتر) و در نتیجه مدل‌های boosting بهتر از «نویز مصنوعی» عبور کنند و مرز تصمیم پایدارتر بسازند.

5) چرا مدل‌های Ensemble (Soft Voting / Stacking) همیشه بهترین نیستند؟

* Ensembleها معمولاً وقتی بهترین می‌شوند که (الف) مدل‌های پایه متنوع باشند و (ب) داده کافی برای جلوگیری از «یادگیری الگوهای تصادفی/ناپایدار» وجود داشته باشد. در داده کوچک (Real-400)، ممکن است مزیت ensemble روی آزمون کوچک پایدار نماند؛ در داده بزرگ‌تر، بهبود مشاهده می‌شود اما همچنان «هزینه FP/FN» می‌تواند به‌صورت سیاستی انتخاب را تغییر دهد. (این منطق با مبانی ارزیابی و پایداری مدل‌ها سازگار است.)

جمع‌بندی نهایی

* Real-400: بهترین عملکرد کلی به درخت تصادفی رسید؛ تفسیر محتمل آن، پایداری بیشتر مدل‌های bagging در داده‌های محدود و کاهش ریسک بیش‌برازش است.
* Syn-2000(50%): بهترین عملکرد کلی به KNN رسید؛ افزایش تراکم/توازن داده باعث تقویت تصمیم‌گیری محلی شده است.
* Syn-2000(100%): بهترین عملکرد کلی به LightGBM رسید؛ با Seed واقعی کامل، Fidelity داده مصنوعی احتمالاً بالاتر بوده و boosting توانسته الگوهای غیرخطی را دقیق‌تر یاد بگیرد.

اگر مرحله بعدی را می‌خواهی، من یک جدول مقایسه‌ای رتبه‌بندی Top-3 برای هر سناریو هم می‌گذارم (با ستون‌های FN/FP/Accuracy/F1) و بعد یک پاراگراف «راهنمای انتخاب مدل بر اساس سیاست امنیتی» (FN محور vs FP محور) اضافه می‌کنم.

**فصل 5: بحث، نتیجه گیري و پیشنهادات**



**5-1 مرور اجمالی نتایج تحقیق**

این پژوهش با هدف افزایش دقت تشخیص حملات تزریق در پایگاه‌داده‌های غیررابطه‌ای و هم‌زمان کنترل هشدارهای کاذب طراحی و اجرا شد. راهبرد کلی پژوهش بر دو محور استوار بود: (1) استفاده از الگوریتم‌های یادگیری ماشین و مقایسه نظام‌مند چندین مدل، و (2) تقویت داده‌های آموزشی از طریق تولید داده مصنوعی با مدل‌های زبانی بزرگ (نسخه GPT-5.2) به‌منظور جبران محدودیت اندازه و تنوع داده واقعی. ادبیات اخیر نیز تأکید می‌کند که تولید داده مصنوعی توسط LLMها می‌تواند در شرایط کمبود داده برچسب‌دار، «افزایش پوشش داده» و «بهبود کارایی مدل» را ممکن کند، مشروط به اینکه کیفیت داده مصنوعی با معیارهای وظیفه‌محور سنجیده شود.

1) جمع‌بندی سناریوهای آزمایشی و شاخص‌های ارزیابی

برای ارزیابی دقیق، آزمایش‌ها در سه سناریوی داده‌ای اجرا شدند:

* سناریوی 1: Real-400 (داده واقعی با حجم محدود)
* سناریوی 2: Syn-2000 (Seed=50% Real) (تولید 2000 نمونه مصنوعی با اتکا به 50% داده واقعی به‌عنوان Seed)
* سناریوی 3: Syn-2000 (Seed=All Real-400) (تولید 2000 نمونه مصنوعی با اتکا به کل داده واقعی به‌عنوان Seed)

ارزیابی عملکرد بر مبنای خروجی آزمون و مجموعه‌ای از معیارهای استاندارد انجام شد؛ به‌ویژه Accuracy، Precision، Recall و F1 و همچنین تحلیل Confusion Matrix برای کنترل دو خطای کلیدی در امنیت یعنی FP (هشدار کاذب) و FN (حمله کشف‌نشده). در چارچوب‌های ارزیابی یادگیری ماشین، Confusion Matrix مبنای اصلی استخراج این شاخص‌هاست.

2) یافته‌های کلیدی پژوهش بر اساس نتایج آزمایش

نتایج نشان داد که نوع داده (واقعی محدود در برابر مصنوعی تقویت‌شده با Seed متفاوت) می‌تواند رتبه‌بندی مدل‌ها را به‌طور معنادار تغییر دهد:

الف) عملکرد روی Real-400:

در سناریوی داده واقعی محدود، مدل درخت تصادفی بهترین عملکرد کلی را ارائه داد. این نتیجه با رفتار شناخته‌شده روش‌های bagging سازگار است؛ زیرا چنین مدل‌هایی معمولاً در نمونه‌های کوچک پایدارتر عمل کرده و حساسیت کمتری به نوسانات داده دارند.

ب) عملکرد روی Syn-2000 (Seed=50% Real)

در این سناریو، KNN بهترین عملکرد کلی را ثبت کرد و هم‌زمان توانست مقدار FP و FN را در سطح پایین نگه دارد. این الگو نشان می‌دهد افزایش تراکم نمونه‌ها و متنوع‌تر شدن فضای ویژگی‌ها (در نتیجه تولید داده مصنوعی) می‌تواند به نفع مدل‌های نمونه‌محور (instance-based) تمام شود.

ج) عملکرد روی Syn-2000 (Seed=All Real-400)

در این سناریو، LightGBM بهترین عملکرد کلی را ارائه داد و از نظر توازن خطاها نیز وضعیت مطلوبی داشت. از منظر نظری، LightGBM به‌عنوان یک پیاده‌سازی کارا از Gradient Boosting درخت تصمیمs، برای یادگیری روابط غیرخطی و تعامل ویژگی‌ها مناسب است و در داده‌های بزرگ‌تر/غنی‌تر می‌تواند مزیت بیشتری نشان دهد.

همچنین مشاهده شد که در برخی سناریوها، مدل‌های ترکیبی مانند Stacking می‌توانند FN را به حداقل برسانند (یعنی ریسک «از دست رفتن حمله» را کمتر کنند)، اما گاهی با افزایش FP همراه می‌شوند. بنابراین «بهترین مدل» به سیاست عملیاتی نیز وابسته است: اگر اولویت سازمان کاهش FN باشد، انتخاب ممکن است متفاوت از حالتی باشد که کاهش FP اولویت دارد.

3) نقش داده مصنوعی تولیدشده با GPT-5.2 در بهبود نتایج

یکی از خروجی‌های مهم پژوهش، تولید داده‌های مصنوعی هدفمند برای سناریوهای حمله/غیرحمله بود که عملاً باعث شد مدل‌ها در سناریوهای Syn-2000 عملکرد بهتری نسبت به حالت داده واقعی محدود داشته باشند. این نتیجه با گزارش‌های پژوهشی هم‌راستا است که نشان می‌دهند LLMها می‌توانند داده‌های مصنوعی «وظیفه‌محور» تولید کنند و در شرایط کمبود داده واقعی، به بهبود آموزش مدل‌ها کمک نمایند.

4) جمع‌بندی نهایی این بخش

در مجموع، نتایج آزمایش‌ها نشان داد:

1. تقویت داده با تولید مصنوعی می‌تواند عملکرد مدل‌ها را به شکل محسوسی بهبود دهد، به‌خصوص وقتی داده واقعی محدود است.
2. بهترین مدل در هر سناریو ثابت نیست و با تغییر کیفیت/منبع داده، مدل‌های متفاوتی در رتبه اول قرار می‌گیرند (Real-400 → درخت تصادفی، Syn-2000(50%) → KNN، Syn-2000(100%) → LightGBM).
3. در ارزیابی امنیتی، تحلیل همزمان FP و FN ضروری است، زیرا FN به معنی عبور حمله و FP به معنی افزایش هشدار کاذب و هزینه عملیاتی است.
4. استفاده از مدل‌های تقویتی پیشرفته (به‌ویژه LightGBM) در داده مصنوعیِ Seed کامل، با انتظار نظری از GBDTهای کارا و توانمند در روابط غیرخطی سازگار است.

## 5-2 تحلیل نتایج در چارچوب پیشینه پژوهش

نتایج این پژوهش در امتداد مسیر تحقیقات پیشینِ مقابله با تزریق (Injection) قرار می‌گیرد، اما از چند جهت «گام جلوتر» است: (1) تمرکز مستقیم بر تزریق در پایگاه‌داده‌های غیررابطه‌ای (NoSQL Injection) که در مقایسه با SQL Injection کمتر پوشش داده شده، و (2) استفاده از تولید داده مصنوعی مبتنی بر مدل‌های زبانی بزرگ برای رفع محدودیت جدی داده‌های برچسب‌دار در این حوزه. در ادبیات موضوع، هم‌زمانیِ ماهیت نوظهور NoSQL و تنوع بالای الگوهای حمله، باعث شده بسیاری از مطالعات به مجموعه‌داده‌های کوچک یا سناریوهای محدود متکی باشند و قابلیت تعمیم در محیط واقعی با چالش مواجه شود.

### 5-2-1 جایگاه پژوهش حاضر نسبت به رویکردهای رایج (ابزارمحور/تحلیل‌کد/یادگیری ماشین)

در کارهای پیشین، بخش قابل توجهی از راهکارها برای کشف NoSQL Injection ماهیت ابزارمحور یا تحلیل ایستا/پویا داشته‌اند. برای نمونه، ابزار NoSQL Racket ایده اصلی خود را بر مقایسه ساختار کوئری NoSQL در سطح کد (تحلیل ایستا) و سطح زمان اجرا (تحلیل پویا) بنا می‌کند و با همین مکانیزم، انحراف ساختاری ناشی از تزریق را شناسایی می‌کند. این نوع رویکردها از منظر مهندسی امنیت، ارزشمندند زیرا به جای یادگیری از داده، مستقیماً رفتار اجرای کوئری را پایش می‌کنند.

با این حال، یافته‌های پژوهش حاضر نشان داد که یک چارچوب داده‌محور و مدل‌محور می‌تواند مکمل/جایگزین موثری برای رویکردهای ابزارمحور باشد؛ به‌ویژه وقتی مسئله با کمبود داده‌های واقعی و تنوع پایین مواجه است. در واقع، در این پژوهش، مدل‌های مختلف یادگیری ماشین روی سه سناریوی داده‌ای ارزیابی شدند و مشاهده شد که با تغییر کیفیت و ترکیب داده (واقعی محدود در برابر مصنوعی تقویت‌شده)، رتبه‌بندی مدل‌ها تغییر می‌کند؛ این موضوع نشان می‌دهد «ماهیت داده» در NoSQL Injection یک عامل تعیین‌کننده است و صرفاً تکیه بر یک مدل منفرد یا یک سناریوی ثابت، می‌تواند نتیجه‌گیری را شکننده کند.

### 5-2-2 چرایی تفاوت نتایج نسبت به برخی مطالعات قبلی

الف) استفاده از داده مصنوعی مبتنی بر LLM به جای تولیدکننده‌های کلاسیک

در بخشی از ادبیات امنیت، تولید داده مصنوعی با روش‌های کلاسیک (از جمله خانواده GANها) مطرح بوده است؛ اما داده‌های امنیتی متنی/ساختاریافته (مانند payloadها، query fragmentها و الگوهای تزریق) اغلب به تولیدکننده‌ای نیاز دارند که «ساختار و معنا» را همزمان حفظ کند. در این پژوهش از GPT-5.2 برای تولید داده مصنوعی استفاده شد؛ رویکردی که با روندهای جدید تحقیقاتی در امنیت همسو است، زیرا مطالعات نشان می‌دهند LLMها می‌توانند برای تولید داده مصنوعی در کاربردهای امنیتی (از جمله تشخیص ناهنجاری و شبیه‌سازی تهدید) مفید باشند به شرط کنترل کیفیت و ارزیابی وظیفه‌محور.

ب) تمرکز مسئله بر NoSQL Injection و شکاف داده/ارزیابی

بر اساس مرورهای حوزه، هرچند SQL Injection بسیار مطالعه شده، اما NoSQL Injection هم به دلیل تفاوت در نحو کوئری‌ها و هم به دلیل تنوع پایگاه‌ها و APIها، به داده‌های متنوع‌تر و ارزیابی دقیق‌تری نیاز دارد. پژوهش حاضر دقیقاً روی این شکاف تمرکز کرده و نشان می‌دهد تقویت داده می‌تواند به «بهبود تعمیم» کمک کند.

ج) اثر داده بر انتخاب مدل و تغییر رتبه‌ها

یافته‌های تجربی شما نشان دادند که بهترین مدل در سناریوی داده واقعیِ محدود با بهترین مدل در سناریوهای مصنوعی یکسان نیست. این موضوع از منظر روش‌شناسی مهم است، چون نشان می‌دهد برخی نتایج مطالعات قبلی ممکن است «وابسته به داده» باشند و با تغییر سناریو داده، نتیجه تغییر کند. از منظر فنی نیز این رفتار قابل انتظار است: مدل‌های مبتنی بر boosting (مثل LightGBM) معمولاً در داده‌های بزرگ‌تر/غنی‌تر فرصت بیشتری برای یادگیری الگوهای غیرخطی و تعامل ویژگی‌ها پیدا می‌کنند؛ و LightGBM نیز به‌عنوان یک پیاده‌سازی کارا از GBDT برای همین هدف طراحی شده است.

### 5-2-3 نقاط قوت و محدودیت‌ها

نقاط قوت

* بهبود عملکرد و تعمیم‌پذیری از طریق سناریوهای داده‌ای متنوع: طراحی سه سناریو باعث شد پایداری انتخاب مدل و حساسیت آن به داده، شفاف شود.
* تولید داده مصنوعی با LLM و افزایش پوشش سناریوهای حمله/غیرحمله: مطابق ادبیات، استفاده از LLM برای تولید داده امنیتی می‌تواند هزینه و زمان تولید داده را کاهش دهد و تنوع را افزایش دهد.
* ارزیابی دقیق خطاهای امنیتی (FP/FN) و نه صرفاً Accuracy: این رویکرد باعث می‌شود نتایج برای استقرار عملی قابل اتکاتر باشند.

محدودیت‌ها

* وابستگی عملکرد به کیفیت داده مصنوعی و ریسک سوگیری/آرتیفکت‌های تولیدی: اگر داده مصنوعی الگوهای غیرواقعی یا سرنخ‌های ساده ایجاد کند، برخی مدل‌ها ممکن است به‌صورت ناپایدار «بیش‌برازش» کنند.
* پیچیدگی عملیاتی مدل‌های قوی‌تر/ترکیبی: برخی مدل‌ها و ensembleها ممکن است هزینه محاسباتی و هزینه نگهداشت بیشتری داشته باشند (به‌خصوص در محیط‌های عملیاتی محدود).
* نیاز به ارزیابی میدانی روی جریان واقعی ترافیک/کوئری در مقیاس سازمانی: ادبیات نیز بر چالش تعمیم از آزمایشگاه به محیط واقعی تأکید دارد.

## 5-3 بررسی کاربردهای عملی نتایج در دنیای واقعی (بازنویسی‌شده و همسو با نتایج شما)

یافته‌های این پژوهش چند مسیر کاربردی مشخص برای صنعت و سازمان‌ها ایجاد می‌کند؛ به‌ویژه برای سامانه‌هایی که بر پایگاه‌داده‌های NoSQL و APIهای مبتنی بر JSON/Document تکیه دارند.

### 5-3-1 کاربردهای مستقیم در سامانه‌های امنیتی

1. افزودن لایه تشخیص مبتنی بر یادگیری ماشین به کنار کنترل‌های کلاسیک

* خروجی پژوهش نشان داد مدل‌ها بسته به سناریوی داده، توانایی متفاوتی در کنترل FN (حمله‌های از دست‌رفته) و FP (هشدارهای کاذب) دارند. بنابراین سازمان می‌تواند یک سیاست عملیاتی تعریف کند:
  + محیط‌های حساس (بانکی/سلامت): انتخاب مدل/پیکربندی با اولویت کاهش FN
  + محیط‌های پرحجم (سرویس‌های ابری/پلتفرم‌ها): موازنه FP برای کاهش هزینه عملیاتی  
    این منطق انتخاب بر پایه تحلیل Confusion Matrix در ارزیابی‌های استاندارد توصیه می‌شود.

1. بهبود WAF/IDS برای سرویس‌های NoSQL محور

* خروجی مدل می‌تواند به‌صورت یک ماژول تشخیصی در کنار WAF یا IDS قرار گیرد و درخواست‌های مشکوک را با احتمال حمله برچسب‌گذاری کند. ادبیات LLM در امنیت نیز نشان می‌دهد سیستم‌های جدید در حال حرکت به سمت روش‌های ترکیبی (یادگیری ماشین + تحلیل معنایی/متنی) هستند.

1. کمک به تیم‌های SecOps در اولویت‌بندی رخدادها

* با داشتن یک طبقه‌بند که ریسک را امتیازدهی می‌کند، رخدادهای امنیتی به جای صف‌های طولانی هشدار، قابلیت رتبه‌بندی و تریاژ پیدا می‌کنند. این موضوع به‌صورت مستقیم هزینه FP را کاهش می‌دهد.

### 5-3-2 کاربردهای غیرمستقیم: تولید داده و تمرین/آموزش امنیتی

1. ساخت مجموعه‌داده آموزشی برای تیم امنیت و تست نفوذ

* داده مصنوعی تولید شده با GPT-5.2 می‌تواند برای آموزش نیروها، ساخت سناریوهای تست و حتی طراحی تمرین‌های قرمز/آبی (Red/Blue Team) به‌کار رود؛ ادبیات جدید نیز کاربرد LLMها در تولید داده‌های امنیتی و شبیه‌سازی تهدید را گزارش کرده است.

1. پشتیبانی از پژوهش‌های آینده در نبود دیتاست استاندارد NoSQL Injection

* یکی از چالش‌های شناخته‌شده، کمبود داده‌های جامع در NoSQL Injection است. خروجی این پژوهش می‌تواند مبنایی برای مقایسه روش‌ها، تست الگوریتم‌های جدید، و استانداردسازی ارزیابی‌ها باشد.

### 5-3-3 پیامدها برای صنعت، جامعه و مسیر تحقیقات آینده

1. افزایش امنیت داده‌های حساس و کاهش سطح ریسک نفوذ

* کاهش FN یعنی احتمال عبور حمله کمتر می‌شود و این موضوع برای سامانه‌های حاوی داده حساس، اثر مستقیم دارد.

1. شتاب‌دهی به نوآوری در سامانه‌های NoSQL محور

* سازمان‌ها با اطمینان بیشتر می‌توانند از معماری‌های مبتنی بر NoSQL استفاده کنند؛ چون یک راهکار داده‌محور برای کنترل تزریق در اختیار دارند.

1. گسترش پژوهش روی «کیفیت داده مصنوعی» و کنترل سوگیری

* ادبیات جدید همزمان هشدار می‌دهد که LLMها ابزار قدرتمندی هستند اما کیفیت داده مصنوعی باید با معیارهای دقیق کنترل شود؛ لذا مسیر آینده پژوهش می‌تواند شامل سنجش fidelity/utility داده و ارزیابی میدانی باشد.

## 5-4 محدودیت‌های پژوهش

در اجرای این پژوهش، مجموعه‌ای از محدودیت‌های عملی و فنی وجود داشت که می‌تواند مسیر بهبود در مطالعات آینده را روشن‌تر کند:

1. محدودیت دسترسی به داده‌های واقعی و برچسب‌دار NoSQL Injection

کمبود دیتاست‌های استاندارد و در دسترس عمومی برای حملات تزریق NoSQL یکی از موانع شناخته‌شده این حوزه است و بسیاری از کارهای داده‌محور نیز به این شکاف اشاره کرده‌اند. این مسئله باعث شد بخش قابل توجهی از تقویت داده از طریق داده مصنوعی انجام شود.

1. وابستگی نتایج به کیفیت و «فیدلیتی/یوتیلیتی» داده مصنوعی

هرچند تولید داده با LLM (در این پژوهش GPT-5.2) توانست پوشش سناریوهای حمله/غیرحمله را افزایش دهد، اما داده مصنوعی ذاتاً با خطر «ایجاد آرتیفکت»، «سوگیری» یا «فاصله توزیعی» همراه است. ادبیات synthetic data نیز روی ضرورت سنجش همزمان فیدلیتی، یوتیلیتی و ریسک تأکید دارد.

1. محدودیت ارزیابی میدانی و استقرار در محیط واقعی

ارزیابی‌ها در چارچوب آزمایشگاهی/شبیه‌سازی انجام شده است. در عمل، رفتار کاربران، تغییر الگوهای ترافیک و تنوع پیاده‌سازی‌های NoSQL می‌تواند الگوی خطاها (FP/FN) را تغییر دهد؛ لذا تعمیم نتایج به محیط عملیاتی نیازمند آزمایش میدانی و پایش مستمر است.

1. هزینه محاسباتی و پیچیدگی عملیاتی مدل‌ها

آموزش چند مدل و انجام ارزیابی چندمرحله‌ای (از جمله مدل‌های boosting و ensemble) نیازمند منابع محاسباتی بیشتر است. هرچند مدل‌هایی مانند LightGBM برای کارایی طراحی شده‌اند، اما همچنان اجرای چندمدلی و تکرار آزمایش‌ها در محیط‌های محدود می‌تواند چالش ایجاد کند.

1. محدودیت دامنه تهدید و پوشش انواع پایگاه‌داده/پیاده‌سازی

NoSQL Injection صرفاً به MongoDB محدود نیست و رفتار حمله در اکوسیستم‌های مختلف (مانند Cassandra یا DynamoDB) می‌تواند متفاوت باشد. حتی پژوهش NoSQL Racket نیز چالش «نبود زبان استاندارد کوئری» در NoSQL و تنوع بسترها را مطرح می‌کند که تعمیم‌پذیری ابزارها/مدل‌ها را پیچیده‌تر می‌سازد.

## 5-5 محدودیت‌های ذاتی روش تحقیق یا داده‌های استفاده‌شده

1. حساسیت رتبه‌بندی مدل‌ها به سناریوی داده

نتایج شما نشان داد «بهترین مدل» میان سناریوهای Real-400، Syn-2000(50%) و Syn-2000(100%) ثابت نمی‌ماند. این موضوع بیانگر آن است که برآورد عملکرد، به کیفیت/ساختار داده و نحوه تقویت آن وابسته است و نتیجه‌گیری باید در قالب چند سناریو گزارش شود (کاری که در این پژوهش انجام شده است). (این نکته با توصیه‌های ارزیابی وظیفه‌محور در تولید داده مصنوعی هم‌راستاست.)

1. پوشش محدود الگوهای حمله نسبت به فضای تهدید واقعی

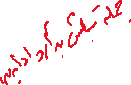
حملات تزریق NoSQL در عمل طیف گسترده‌ای از payloadها و مسیرهای اکسپلویت (مانند bypass احراز هویت، دستکاری شرط‌ها، و انواع تزریق در JSON/query operators) دارند. بنابراین حتی با تولید داده مصنوعی، احتمال باقی‌ماندن شکاف نسبت به تهدیدات «کاملاً نوظهور» وجود دارد.

1. عدم ارزیابی طولانی‌مدت و Drift مفهومی (Concept Drift)

الگوهای حمله و رفتار سامانه‌ها در طول زمان تغییر می‌کند؛ در نتیجه مدل‌ها بدون پایش و بازآموزی دوره‌ای ممکن است افت عملکرد داشته باشند. این محدودیت در بسیاری از سامانه‌های امنیتی داده‌محور وجود دارد و در پژوهش حاضر نیز به‌عنوان محدودیت ذاتی مطرح است.

## 5-6 پیشنهادها برای تحقیقات آینده

1. استانداردسازی و توسعه دیتاست ترکیبی چندسناریویی



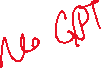
پیشنهاد می‌شود دیتاست‌های آینده به‌صورت رسمی با سناریوهای مشخص (واقعی محدود، مصنوعی با Seed جزئی، مصنوعی با Seed کامل، و ترکیبی) منتشر شوند تا مقایسه روش‌ها پایدارتر گردد. وجود دیتاست‌های تخصصی MongoDB Injection نیز نشان می‌دهد مسیر استانداردسازی در حال شکل‌گیری است و می‌تواند گسترش یابد.



1. کنترل کیفیت داده مصنوعی با چارچوب‌های ارزیابی Fidelity/Utility/Privacy  
   برای کاهش ریسک آرتیفکت و افزایش اتکا، پیشنهاد می‌شود تولید داده با LLM همراه با سنجش‌های سیستماتیک باشد (Utility وظیفه‌محور، Fidelity آماری/ساختاری و در صورت لزوم ارزیابی ریسک).
2. ارزیابی میدانی در مقیاس واقعی و تحلیل هزینه خطا (Cost-Sensitive)  
   در محیط واقعی، هزینه FN (از دست رفتن حمله) معمولاً بسیار بیشتر از هزینه FP (هشدار کاذب) است. پیشنهاد می‌شود ارزیابی آینده علاوه بر معیارهای کلاسیک، با معیارهای هزینه‌محور و سیاست‌های عملیاتی (FN-محور/FP-محور) انجام شود.
3. گسترش دامنه به چند پایگاه‌داده و چند API/چارچوب وب

با توجه به تنوع NoSQL و نبود زبان استاندارد مشترک (چالشی که NoSQL Racket نیز برجسته می‌کند)، پیشنهاد می‌شود آزمایش‌ها روی چند DBMS و چند چارچوب توسعه (Node/Python/Java) تکرار شود.

1. ادغام رویکرد داده‌محور با روش‌های ابزارمحور/تحلیل ایستا-پویا



یک مسیر آینده‌دار، ترکیب طبقه‌بندی یادگیری ماشین با سیگنال‌های ابزارمحور (مانند تحلیل ساختار query در runtime) است تا هم قدرت تعمیم داده‌محور حفظ شود و هم کشف‌های ساختاری دقیق‌تر گردد.



## 5-7 جمع‌بندی نهایی

این پژوهش یک چارچوب داده‌محور برای تشخیص حملات تزریق در پایگاه‌داده‌های غیررابطه‌ای ارائه کرد که دو مؤلفه کلیدی داشت: (1) ارزیابی مقایسه‌ای چند مدل یادگیری ماشین و مدل‌های ترکیبی، و (2) تقویت داده از طریق تولید نمونه‌های مصنوعی با GPT-5.2 در دو سطح Seed (۵۰٪ و ۱۰۰٪ داده واقعی). نتایج آزمایش‌ها نشان دادند که با تغییر سناریوی داده، «بهترین مدل» ثابت نمی‌ماند و این امر اهمیت طراحی چندسناریویی و تحلیل همزمان FP/FN را برجسته می‌کند. همچنین، استفاده از داده مصنوعی به‌عنوان راهکار جبران کمبود داده واقعی، با ادبیات جدیدِ تولید داده توسط LLMها هم‌سو است؛ مشروط به اینکه کیفیت داده مصنوعی به‌صورت چارچوب‌مند ارزیابی شود.

در نهایت، با وجود محدودیت‌هایی مانند نیاز به ارزیابی میدانی و کنترل کیفی دقیق داده مصنوعی، پژوهش حاضر یک گام عملی در جهت پر کردن شکاف داده و تقویت تشخیص NoSQL Injection محسوب می‌شود و می‌تواند به‌عنوان مبنایی برای توسعه ابزارهای امنیتی داده‌محور و مطالعات آینده در این حوزه به کار رود.

فهرست مراجع

[1] A. A. Shairgojri and S. A. Dar, “Emerging Cyber Security India’s Concern and Threats,” *Journal of Technology Innovations and Energy*, vol. 1, no. 2, 2022, doi: 10.56556/jtie.v1i2.201.

[2] W. S. Admass, Y. Y. Munaye, and A. A. Diro, “Cyber security: State of the art, challenges and future directions,” 2024. doi: 10.1016/j.csa.2023.100031.

[3] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” 2023. doi: 10.3390/electronics12061333.

[4] D. Nyale and S. M. Angolo, “A Survey of Artificial Intelligence in Cyber Security,” *International Journal of Computer Applications Technology and Research*, 2022, doi: 10.7753/ijcatr1112.1014.

[5] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, “Data-Driven Cybersecurity Incident Prediction: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, 2019, doi: 10.1109/COMST.2018.2885561.

[6] L. Yan, L. Yao, Q. Zhao, M. Xiao, Y. Li, and S. Min, “Risk Prediction Models for Inadvertent Intraoperative Hypothermia: A Systematic Review,” 2021. doi: 10.1016/j.jopan.2021.02.011.

[7] O. Ben Fredj, O. Cheikhrouhou, M. Krichen, H. Hamam, and A. Derhab, “An OWASP Top Ten Driven Survey on Web Application Protection Methods,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021. doi: 10.1007/978-3-030-68887-5\_14.

[8] A. C. T. Santos, J. L. S. Filho, Á. Í. S. Silva, V. Nigam, and I. E. Fonseca, “BLE injection-free attack: a novel attack on bluetooth low energy devices,” *J Ambient Intell Humaniz Comput*, vol. 14, no. 5, 2023, doi: 10.1007/s12652-019-01502-z.

[9] A. Sadeghian, M. Zamani, and S. Ibrahim, “SQL injection is still alive: A Study on SQL injection signature evasion techniques,” in *Proceedings - 2013 International Conference on Informatics and Creative Multimedia, ICICM 2013*, 2013. doi: 10.1109/ICICM.2013.52.

[10] S. K. Selvaraj, A. Raj, R. Rishikesh Mahadevan, U. Chadha, and V. Paramasivam, “A Review on Machine Learning Models in Injection Molding Machines,” 2022. doi: 10.1155/2022/1949061.

[11] D. Chen, Q. Yan, C. Wu, and J. Zhao, “SQL Injection Attack Detection and Prevention Techniques Using Deep Learning,” in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1757/1/012055.

[12] A. Jain *et al.*, “Overview and Importance of Data Quality for Machine Learning Tasks,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020. doi: 10.1145/3394486.3406477.

[13] J. Dean, D. Patterson, and C. Young, “A New Golden Age in Computer Architecture: Empowering the Machine-Learning Revolution,” *IEEE Micro*, vol. 38, no. 2, 2018, doi: 10.1109/MM.2018.112130030.

[14] M. Takamoto *et al.*, “PDEBENCH: An Extensive Benchmark for Scientific Machine Learning,” in *Advances in Neural Information Processing Systems*, 2022.

[15] R. D․ L․, A. H․ V․, P. B. Honnavalli, and N. S․, “The MongoDB injection dataset: A comprehensive collection of MongoDB - NoSQL injection attempts and vulnerabilities,” *Data Brief*, vol. 54, 2024, doi: 10.1016/j.dib.2024.110289.

[16] N. W. Paton and O. Díaz, “Active Database Systems,” *ACM Comput Surv*, vol. 31, no. 1, 1999, doi: 10.1145/311531.311623.

[17] M. Miftakul Amin, A. Maseleno, K. Shankar, E. Perumal, R. M. Vidhyavathi, and S. K. Lakshmanaprabu, “Active database system approach and rule based in the development of academic information system,” *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 2.26 Special Issue  26, 2018, doi: 10.14419/ijet.v7i2.26.14361.

[18] PortSwigger, “What is cross-site scripting (XSS) and how to prevent it? | Web Security Academy,” portswigger.net.

[19] M. Bravenboer, E. Dolstra, and E. Visser, “Preventing injection attacks with syntax embeddings,” *Sci Comput Program*, vol. 75, no. 7, 2010, doi: 10.1016/j.scico.2009.05.004.

[20] E. F. Codd, “Relational Database: A Practical Foundation for Productivity,” *Commun ACM*, vol. 25, no. 2, 1982, doi: 10.1145/358396.358400.

[21] W. Zhou *et al.*, “Attack sample generation algorithm based on data association group by GAN in industrial control dataset,” *Comput Commun*, vol. 173, 2021, doi: 10.1016/j.comcom.2021.04.014.

[22] Internet World Stats, “World Internet Users Statistics. World Population Stats,” Usage and Population Statistics.

[23] S. N. F. N. B. Mustaffa and M. Farhan, “Detection of False Data Injection Attack using Machine Learning approach,” *Mesopotamian Journal of CyberSecurity*, vol. 2022, 2022, doi: 10.58496/MJCS/2022/005.

[24] S. Lockey, N. Gillespie, D. Holm, and I. A. Someh, “A review of trust in artificial intelligence: Challenges, vulnerabilities and future directions,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2021. doi: 10.24251/hicss.2021.664.

[25] Y. Himeur, S. S. Sohail, F. Bensaali, A. Amira, and M. Alazab, “Latest trends of security and privacy in recommender systems: A comprehensive review and future perspectives,” 2022. doi: 10.1016/j.cose.2022.102746.

[26] D. Van Landuyt, V. Wijshoff, and W. Joosen, “A study of NoSQL query injection in Neo4j,” *Comput Secur*, vol. 137, 2024, doi: 10.1016/j.cose.2023.103590.

[27] D. Das, U. Sharma, and D. K. Bhattacharyya, “Defeating SQL injection attack in authentication security: an experimental study,” *Int J Inf Secur*, vol. 18, no. 1, 2019, doi: 10.1007/s10207-017-0393-x.

[28] M. Haššo, O. Sarakhman, S. Đurđić, D. Stanković, and Ľ. Švorc, “Advanced electrochemical platform for simple and rapid quantification of tannic acid in beverages using batch injection analysis with amperometric detection,” *Journal of Electroanalytical Chemistry*, vol. 942, 2023, doi: 10.1016/j.jelechem.2023.117578.

[29] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, “SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review,” 2023. doi: 10.3390/bdcc7020097.

[30] J. K. Chen and W. Z. Lee, “An introduction of NoSQL databases based on their categories and application industries,” *Algorithms*, vol. 12, no. 5, 2019, doi: 10.3390/a12050106.

[31] F. Chen *et al.*, “Research Progress on Soil Security Assessment in Farmlands and Grasslands Based on Bibliometrics over the Last Four Decades,” *Sustainability (Switzerland)*, vol. 16, no. 1, 2024, doi: 10.3390/su16010404.

[32] J. Brownlee, “Basic Concepts in Machine Learning,” 2015.

[33] A. S. Lea and D. S. Jones, “Mind the Gap — Machine Learning, Dataset Shift, and History in the Age of Clinical Algorithms,” *New England Journal of Medicine*, vol. 390, no. 4, 2024, doi: 10.1056/nejmp2311015.

[34] J. Hegde and B. Rokseth, “Applications of machine learning methods for engineering risk assessment – A review,” 2020. doi: 10.1016/j.ssci.2019.09.015.

[35] C. Virmani, T. Choudhary, A. Pillai, and M. Rani, “Applications of machine learning in cyber security,” in *Research Anthology on Machine Learning Techniques, Methods, and Applications*, 2022. doi: 10.4018/978-1-6684-6291-1.ch033.

[36] U. Tank, S. Arirangan, A. R. Paduri, and N. Darapaneni, “A Study Towards Building Content Aware Models in NLP using Genetic Algorithms,” *EAI Endorsed Transactions on AI and Robotics*, vol. 2, 2023, doi: 10.4108/airo.4078.

[37] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A survey of transformers,” *AI Open*, vol. 3, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[38] A. Saka *et al.*, “GPT models in construction industry: Opportunities, limitations, and a use case validation,” 2024. doi: 10.1016/j.dibe.2023.100300.

[39] K. Han *et al.*, “A Survey on Vision Transformer,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 1, 2023, doi: 10.1109/TPAMI.2022.3152247.

[40] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient Transformers: A Survey,” *ACM Comput Surv*, vol. 55, no. 6, 2022, doi: 10.1145/3530811.

[41] V. O. Jithin and N. Subramanian, “SECURE-D:Framework For Detecting and Preventing Attacks in SQL and NoSQL Databases,” in *10th International Symposium on Digital Forensics and Security, ISDFS 2022*, 2022. doi: 10.1109/ISDFS55398.2022.9800805.

[42] V. Sachdeva and S. Gupta, “Basic NOSQL Injection Analysis and Detection on MongoDB,” in *2018 International Conference on Advanced Computation and Telecommunication, ICACAT 2018*, 2018. doi: 10.1109/ICACAT.2018.8933707.

[43] A. M., O. H., H. M., and A. S., “NoSQL Racket: A Testing Tool for Detecting NoSQL Injection Attacks in Web Applications,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017, doi: 10.14569/ijacsa.2017.081178.

[44] S. Priyadharshini and R. Rajmohan, “Analysis on Database Security Model Against NOSQL Injection,” 2017.

[45] L. Song, Y. Li, and N. Lu, “ProfileSR-GAN: A GAN Based Super-Resolution Method for Generating High-Resolution Load Profiles,” *IEEE Trans Smart Grid*, vol. 13, no. 4, 2022, doi: 10.1109/TSG.2022.3158235.

[46] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, “Trustworthy Artificial Intelligence: A Review,” 2023. doi: 10.1145/3491209.

[47] D. Ray and J. Ligatti, “Defining code-injection attacks,” in *Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, 2012. doi: 10.1145/2103656.2103678.

**Abstract**

NoSQL injection is a critical web-application vulnerability that typically emerges from insufficient input validation and unsafe construction of dynamic database queries, enabling attackers to manipulate NoSQL queries to bypass authentication, exfiltrate data, or alter sensitive records. With the increasing adoption of NoSQL databases in modern applications, the need for robust, data-driven detection mechanisms has become more pressing—especially given the limited availability of reliable, labeled datasets for NoSQL injection research.

This thesis proposes a data-centric approach to improving NoSQL injection detection accuracy while reducing false alarms. A key contribution is addressing the data scarcity challenge through dataset augmentation using a large language model. In addition to a real dataset (Real-400), two synthetic datasets (Syn-2000) were generated using GPT-5.2 under two seeding strategies: (i) Seed=50% Real and (ii) Seed=All Real-400. Recent literature highlights that LLM-driven synthetic data generation can provide task-relevant augmentation when real labeled data is scarce, provided that the generated data is evaluated in a task-based manner.

Multiple machine learning models were trained and evaluated, including رگرسیون خطی, درخت تصادفی, XGBoost, LightGBM, نزدیکترین همسایه (KNN), and ensemble configurations based on soft voting and stacking. Model assessment followed systematic evaluation procedures that preserve class distributions (e.g., Stratified K-Fold validation) to mitigate biased estimates under class imbalance. Experimental results demonstrate that the best-performing model is scenario-dependent: درخت تصادفی achieved the strongest overall performance on Real-400; KNN performed best on Syn-2000 (Seed=50% Real); and LightGBM provided the best overall balance on Syn-2000 (Seed=All Real-400). Moreover, ensemble methods were observed to reduce false negatives in certain settings, sometimes at the cost of increased false positives—indicating that operational deployment should be guided by the organization’s risk preference (FN-sensitive vs. FP-sensitive).

Overall, the findings confirm that GPT-5.2–based synthetic data augmentation can materially enhance NoSQL injection detection and improve generalization, while joint analysis of FP and FN yields a more operationally meaningful view of security risk. The results also suggest that tree-based boosting methods such as LightGBM—designed for efficient learning of complex non-linear patterns—are particularly effective under richer, more representative datasets.

**Keywords:** NoSQL injection, NoSQL databases, synthetic data, large language models (GPT-5.2), machine learning, ensemble learning, attack detection, false positive reduction.



In the Name of God



**دانشگاه صنعتی مالک‌اشتر**

Malek Ashtar University of Technology

Faculty of Electrical and Computer Engineering

M.Sc. Thesis

Title

**Improving the Detection of پایگاه‌داده غیررابطه‌ای Injection Attacks Based on Machine Learning**

By

Amirhossein Ghasemi

Supervised by

[Supervisor's name(s)]

Advised by

[Advisor's name(s)]

**Dec. 23, 2024**

1. NoSQL Injection [↑](#footnote-ref-2)
2. Data Set [↑](#footnote-ref-3)
3. Machin Learning [↑](#footnote-ref-4)
4. NoSQL [↑](#footnote-ref-5)
5. Malicious Payloads [↑](#footnote-ref-6)
6. Legitimate Payloads [↑](#footnote-ref-7)
7. Web scraping [↑](#footnote-ref-8)
8. indexing over array [↑](#footnote-ref-9)
9. embedded objects [↑](#footnote-ref-10)
10. Javascript Injections [↑](#footnote-ref-11)
11. Piggy-backed Queries [↑](#footnote-ref-12)
12. Escape Sequences [↑](#footnote-ref-13)
13. Carriage Return [CR] [↑](#footnote-ref-14)
14. Line Feed [LF] [↑](#footnote-ref-15)
15. Origin Violation [↑](#footnote-ref-16)
16. Supervised Learning [↑](#footnote-ref-17)
17. Unsupervised Learning [↑](#footnote-ref-18)
18. Reinforcement Learning [↑](#footnote-ref-19)
19. Decession tree [↑](#footnote-ref-20)
20. K-Nearest Neighbors [↑](#footnote-ref-21)
21. Transformer [↑](#footnote-ref-22)
22. Generator [↑](#footnote-ref-23)
23. Discriminator [↑](#footnote-ref-24)
24. Parser Module [↑](#footnote-ref-25)
25. Detection Module [↑](#footnote-ref-26)
26. Log Module [↑](#footnote-ref-27)
27. Parameterized Statements [↑](#footnote-ref-28)
28. Data-Centric Security Model [↑](#footnote-ref-29)
29. Information Gain [↑](#footnote-ref-30)
30. Correlation [↑](#footnote-ref-31)
31. Piggybacked Queries [↑](#footnote-ref-32)
32. 10-fold Cross Validation [↑](#footnote-ref-33)
33. Recall [↑](#footnote-ref-34)
34. Precision [↑](#footnote-ref-35)
35. F1-Score [↑](#footnote-ref-36)
36. ROC-AUC [↑](#footnote-ref-37)
37. Confusion Matrix [↑](#footnote-ref-38)
38. Mean Squared Error [↑](#footnote-ref-39)
39. Mean Absolute Error [↑](#footnote-ref-40)
40. Mean Absolute Percentage Error [↑](#footnote-ref-41)
41. Accuracy [↑](#footnote-ref-42)
42. Area Under the Curve [↑](#footnote-ref-43)
43. Sensitivity [↑](#footnote-ref-44)
44. Specificity [↑](#footnote-ref-45)
45. Sensitivity [↑](#footnote-ref-46)