

تشخیص بوها در کد با استفاده از روش های یادگیری عمیق

طه اخلاق‌پسندی و محمدهادی علائیان*

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی

چکیده

بو در کد یک نشانه است که احتمالا نشان دهنده ی یک مشکل عمیق تر در برنامه است، بوی کد مانع نگهداری، توسعه و تکامل برنامه می شود. گاهی اوقات بوی کد ناشی از طراحی و سبک برنامه نویسی ضعیف است. پس تشخیص بوها در کد امری ضروریست. در سال‌های گذشته کارهای زیادی برای طراحی یک مدل یادگیری ماشین یا یادگیری عمیق برای تشخیص نفوذ انجام شده است. این مدل‌ها با استفاده از الگوریتم های یادگیری ماشین، مانند ¹RF، ²SVM، ³Decision tree، ⁴DNN، ⁵ANN، ⁶CNN، ⁷RNN، ⁸LSTM و ⁹GRU با دقت های مختلفی ساخته شده است.

هدف از این پژوهش، ارائه روشی برای بهبود نتیجه تشخیص بوها در کد با استفاده از روش‌های یادگیری ماشین است. چرا که روش‌های یادگیری ماشین، در حال رشد هستند و دائماً با روش‌هایی که از لحاظ عملکردی و قدرت پردازشی، کارایی و دقت بهتری دارند جایگزین می‌شوند.

کلیمات کلیدی: یادگیری ماشین، تشخیص بو در کد، یادگیری عمیق.

مقدمه

واژه بوی کد برای نشان دادن وجود مشکلات کیفیت در کد استفاده می شود. تعداد زیادی از بوها در یک سیستم نرم افزاری با سطح بالایی از مشکلات فنی مرتبط است که تکامل سیستم را مختل می کند. اول اینکه بو همانطور که از نامش پیداست بو می دهد پس چیزی است که سریعاً تشخیص داده می شود. یک متد طولانی یا یک کلاس با داده زیاد ولی بدون رفتار، مثال های خوبی در این مورد هستند. دومین نکته اینکه، بو همیشه نشان از یک مشکل نیست. برای مثال برخی از متد های طولانی بسیار خوب هستند. باید عمیق تر به آنها نگاه کرد تا متوجه شد که آیا یک مشکل اساسی وجود دارد یا خیر. بو ها به ذات خود بد نیستند. آنها اغلب نشان دهنده ی یک مشکل هستند نه خود مشکل. وجود بوی بد در کد قطعاً به این معنی نیست که نرم افزار کار نمی کند، نرم افزار با وجود بوی بد کد باز هم خروجی خود را می دهد. اما این بوی بد ممکن است باعث کند شدن پردازش، افزایش خطر خرابی و خطاهای نرم افزار شود. کد بد بو باعث پایین آمدن کیفیت کد و در نتیجه افزایش بدهی فنی می شود.

چند نمونه بو در کد به صورت زیر است.

Long Methods: متدهایی که شامل خطوط زیادی هستند. معمولاً متدهایی با بیش از ۱۰ خط را متدهای بزرگ می دانیم.

¹ Random Forest

² Support vector machine

³ Deep Neural Network

⁴ Artificial neural network

⁵ Convolutional Neural Networks

⁶ Recurrent Neural Networks

⁷ Long short-term memory

⁸ Gated Recurrent Units

* Corresponding author (m.alaeiyan@kntu.ac.ir)

Feature Envoy: وقتی یک متد، علاوه بر داده‌های خودش، به داده‌های یک **object** دیگر دسترسی پیدا کند.
God Class: کلاسی‌هایی که تعداد زیادی متغییر و تابع را شامل می‌شوند.
Long Parameter List: یک تابع بیش از سه یا چهار پارامتر ورودی داشته باشد.
Statements Switch: وقتی یک عملگر **switch** پیچیده یا دنباله‌ای از دستورات **if** در کد وجود دارد.
Data Class: کلاس داده‌ای فقط شامل فیلدها و متدهایی ساده برای دست‌یابی به این فیلدها هستند. **getters and setters** این کلاس‌ها به سادگی تبدیل به ظرف‌هایی برای داده‌های مورد استفاده توسط کلاس‌های دیگر می‌شوند. این کلاس‌ها هیچ قابلیت دیگری ندارند و نمی‌توانند به طور مستقل از داده‌هایی که متعلق به خود آنهاست استفاده کنند.

کارهای پیشین

در این بخش به مدل‌هایی که قبلاً برای این کار ساخته شده بود، می‌پردازیم.

مدل CNN: معماری مدل **CNN** که برای تشخیص بویا استفاده می‌شود. این معماری از معماری‌های معمولی **CNN** که در طبقه بندی تصاویر استفاده می‌شود الهام گرفته شده است و از یک بخش استخراج ویژگی و به دنبال آن یک بخش طبقه بندی تشکیل شده است. بخش استخراج ویژگی از مجموعه‌ای از لایه‌ها، به طور خاص، کانولوشن، نرمال سازی دسته‌ای و لایه‌های حداکثر ادغام تشکیل شده است. این مجموعه از لایه‌ها لایه‌های پنهان معماری را تشکیل می‌دهند. لایه کانولوشن عملیات کانوالو را بر اساس پارامترهای فیلتر و کرنل مشخص شده انجام می‌دهد و وزن شبکه را به لایه بعدی محاسبه می‌کند، در حالی که لایه حداکثر باعث کاهش ابعاد فضای ویژگی می‌شود. نرمال سازی دسته‌ای اثرات توزیع‌های ورودی متنوع برای هر مینی دسته آموزشی را کاهش می‌دهد، بنابراین آموزش را بهینه می‌کند. برای آزمایش با پیکربندی‌های مختلف، از یک، دو و سه لایه پنهان استفاده شده [2].

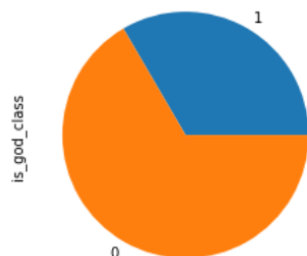
مدل RNN: معماری مدل **RNN** که از مدل‌های پیشرفته در مدل سازی زبان طبیعی الهام گرفته شده است که از یک شبکه **LSTM** به عنوان یک لایه بازگشتی استفاده می‌کند. این مدل از یک لایه جاسازی و به دنبال آن بخش یادگیری ویژگی یک لایه **LSTM** پنهان تشکیل شده است [2].

مدل رمزگذار خودکار: رمزگذارهای خودکار، شبکه‌های عصبی هستند که می‌توانند نمایش‌های معنی دار داده‌ها را به روشی بدون نظارت بیاموزند. هدف رمزگذارهای خودکار این است که یاد بگیرند که یک کپی از ورودی داده شده را بازسازی کنند. برای این هدف، یک قسمت گلوگاه مانند بین لایه‌های ورودی و خروجی، ورودی را در یک نمایش فشرده رمزگذاری می‌کند که به نوبه خود توسط یک قسمت رمزگشایی از حالت فشرده خارج می‌شود. اصل اساسی این است که نمایش کدگذاری شده ویژگی‌های برجسته‌ای را که در خروجی بازسازی شده منعکس می‌شود را به تصویر می‌کشد و سایر موارد کمتر مهم را کنار می‌گذارد، بنابراین قابلیت کاهش ابعاد و کاهش نویز را ارائه می‌دهد [2].

توضیحات مجموعه داده

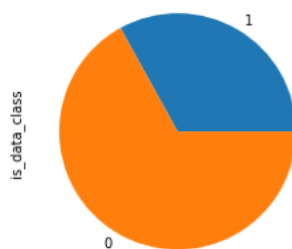
ما شش دیتاست از بوهای **DataClass**، **FeatureEnvoy**، **GodClass**، **LongMethod**، **LongParameterList**، **SwitchStatements** را بدست آوردیم. هر دیتاست ویژگی‌های مربوط به خودش را دارد. همچنین هر رکورد در هر دیتاست، یک برچسب صفر یا یک دارد که معنی آن این است که آیا این سطر دارای بو میباشد یا خیر.

در دیتاست GodClass مجموعاً ۴۱۹ رکورد وجود دارد با ۶۲ ویژگی. همچنین فراوانی تعداد بوهای این مجموعه داده، ۲۷۹ رکورد بدون بو و ۱۴۰ رکورد دارای بو است.



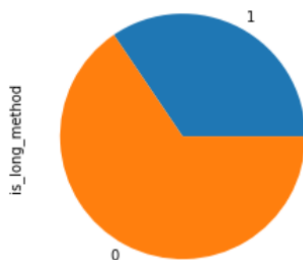
شکل ۱- تعداد بوهای مجموعه داده GodClass

در دیتاست DataClass مجموعاً ۴۱۸ رکورد وجود دارد با ۶۲ ویژگی. همچنین فراوانی تعداد بوهای این مجموعه داده، ۲۸۰ رکورد بدون بو و ۱۳۸ رکورد دارای بو است.



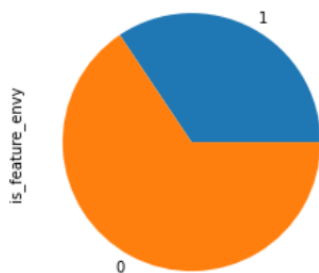
شکل ۲- تعداد بوهای مجموعه داده DataClass

در دیتاست LongMethod مجموعاً ۴۰۶ رکورد وجود دارد با ۸۳ ویژگی. همچنین فراوانی تعداد بوهای این مجموعه داده، ۲۶۶ رکورد بدون بو و ۱۴۰ رکورد دارای بو است.



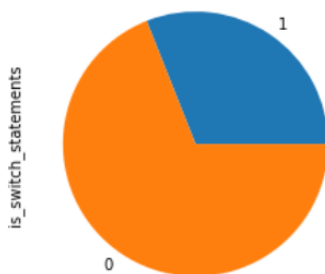
شکل ۳- تعداد بوهای مجموعه داده LongMethod

در دیتاست FeatureEnvy مجموعاً ۴۰۷ رکورد وجود دارد با ۸۳ ویژگی. همچنین فراوانی تعداد بوهای این مجموعه داده، ۲۶۷ رکورد بدون بو و ۱۴۰ رکورد دارای بو است.
در دیتاست SwitchStatements مجموعاً ۴۱۶ رکورد وجود دارد با ۵۶ ویژگی. همچنین فراوانی تعداد بوهای این



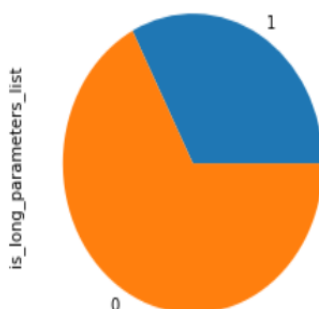
شکل ۴- تعداد بوهای مجموعه داده FeatureEnvy

مجموعه داده، ۲۸۷ رکورد بدون بو و ۱۲۹ رکورد دارای بو است.
در دیتاست LongParameterList مجموعاً ۴۱۳ رکورد وجود دارد با ۵۶ ویژگی. همچنین فراوانی تعداد بوهای این



شکل ۵- تعداد بوهای مجموعه داده SwitchStatements

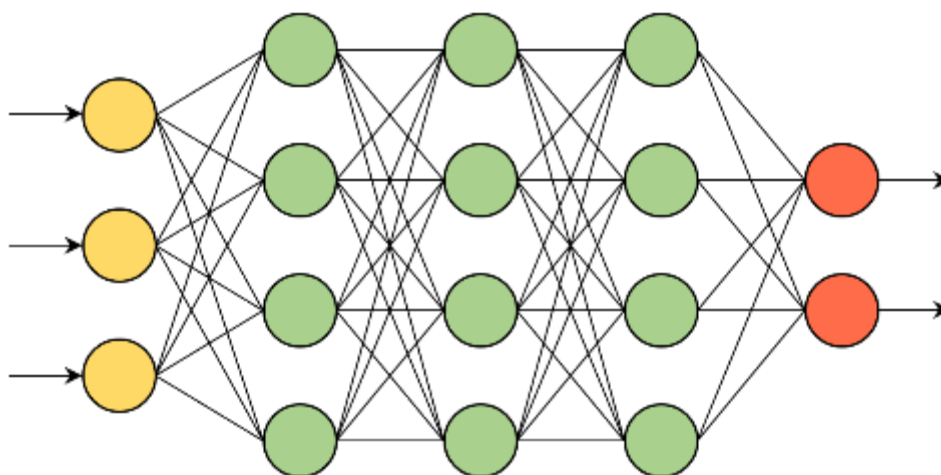
مجموعه داده، ۲۷۸ رکورد بدون بو و ۱۳۵ رکورد دارای بو است.



شکل ۶- تعداد بوهای مجموعه داده
LongParameterList

ساخت مدل

برای ساخت مدل یادگیری عمیق، از شبکه‌های عصبی عمیق استفاده می‌کنیم. شبکه عصبی عمیق یک شبکه عصبی مصنوعی (ANN^1) با چندین لایه پنهان بین لایه ورودی و لایه خروجی یک شبکه عصبی عمیق (DNN^2) است. شبکه عصبی عمیق به دنبال روش ریاضی درستی است، رابطه می‌تواند خطی یا غیر خطی باشد تا ورودی را به خروجی تبدیل کند. DNN ها معمولاً شبکه‌های پیشخور هستند که در آنها داده‌ها از ورودی به لایه خروجی بدون حلقه بازگشتی جریان می‌یابد. DNN نقشه نوروں مجازی را تولید می‌کند و مقادیر عددی دلخواه یا "وزن" را به اتصالات بین نوروں ها اختصاص می‌دهد. ورودی و وزن‌ها ضرب می‌شوند و مقدار بین ۱ و ۰ برمی‌گردد. اگر شبکه دنباله‌ای را به درستی شناسایی نکرده باشد، الگوریتم وزن‌ها را تغییر می‌دهد. این به الگوریتم اجازه می‌دهد تا آن پارامترها را دستکاری کند تا زمانی که دستکاری ریاضی مناسب برای تکمیل پردازش داده‌ها تصمیم‌گیری شود [1].



شکل ۷- شبکه عصبی عمیق

¹ Artificial Neural Network

² Deep Neural Network

نمای کلی مدل به صورت زیر است.

```
(layers): Sequential(
  (0): Linear(in_features=61-82-55, out_features=128, bias=True)
  (1): Tanh()
  (2): Linear(in_features=128, out_features=64, bias=True)
  (3): Tanh()
  (4): Linear(in_features=64, out_features=32, bias=True)
  (5): Sigmoid()
  (6): Linear(in_features=32, out_features=16, bias=True)
  (7): Sigmoid()
  (8): Linear(in_features=16, out_features=2, bias=True)
)
```

پارامترهای مدل DNN در جدول ۱ توضیح داده شده است.

جدول ۱- پارامترهای شبکه عصبی عمیق

پارامتر	نوع / مقدار
تعداد لایه های پنهان	۴
تعداد نورون ها در هر لایه	۱۶-۳۲-۶۴-۱۲۸
نرخ یادگیری	۰.۰۲
تابع بهینه ساز	Adam
توابع فعال ساز	Tanh - Sigmoid
دوره	۱۵۰
اندازه دسته	۱۵

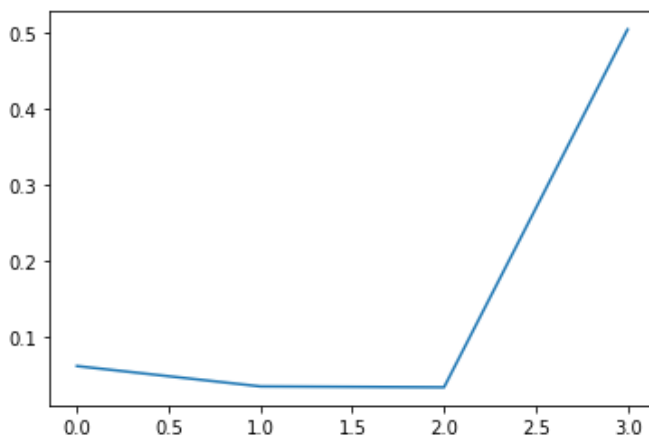
نتایج آزمایش مدل یادگیری عمیق

نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده GodClass در جدول ۲ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برجسب گذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۹۶.۸۰٪ است. در بخش آموزش خطا ۰.۰۵۰۵۰ بود. همچنین مدت زمان آموزش ۱۴۴۶۹ میلی ثانیه است.

جدول ۲- نتایج مدل شبکه عصبی عمیق برای مجموعه داده GodClass

نتایج	معیار
۹۶.۸۰٪	دقت آزمایش
۰.۰۵۰۵۰	خطای آموزش
۱۴۴۶۹ ms	مدت زمان آموزش

همچنین نمودار خطی آموزشی به صورت زیر است.



شکل ۸- نمودار خطی GodClass

معیارهای ارزیابی Precision - Recall و F1-score

معیار Recall یا یادآوری

حداکثر مقدار این معیار یک و یا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که ما انتظار داشتیم پیش بینی شوند ولی برنامه پیش‌بینی نکرده‌است که به آن False Negative می‌گوییم نسبت به پیش‌بینی‌های درست یا True Positive بیشتر باشد مقدار Recall کمتر خواهد شد.

فرمول محاسبه Recall

در فرمول زیر TP مخفف True Positive و FN مخفف False Negative است.

$$\text{Recall} = \frac{TP}{FN+TP}$$

معیار Precision یا دقت

حداکثر مقدار این معیار یک و یا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که برنامه به غلط پیش‌بینی کرده‌است که به آن False Positive می‌گوییم نسبت به پیش‌بینی‌های درست یا True Positive بیشتر باشد مقدار Precision کمتر خواهد شد.

فرمول محاسبه Precision

در فرمول زیر TP مخفف True Positive و FP مخفف False Positive است.

$$\text{Precision} = \frac{TP}{TP+FP}$$

معیار f1-score

زمانی که می‌خواهید معیار ارزیابی شما میانگینی از دو مورد قبلی باشد یعنی همان Precision یا Recall می‌توانید از میانگین هارمونیک این دو معیار استفاده کنید که به آن معیار f1-score می‌گویند.

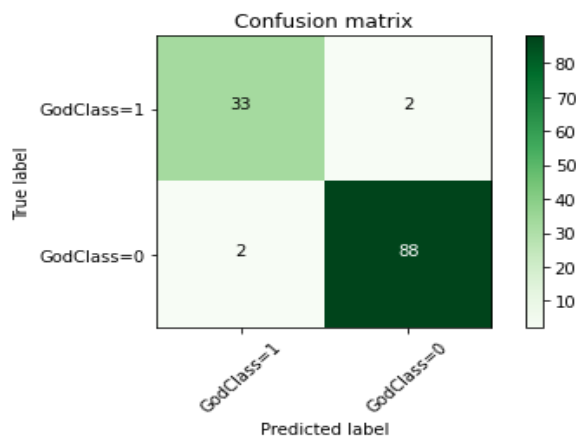
$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

معیارهای ارزیابی Precision - Recall و F1-score به صورت زیر است.

جدول ۳- معیارهای ارزیابی Recall - Precision و F1-score برای مجموعه داده GodClass

کلاس	precision	recall	f1-score
۰	0.98	0.98	0.98
۱	0.94	0.94	0.94

و ماتریس درهم ریختگی آن نیز به صورت زیر است.
نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده DataClass در جدول ۴ نشان داده شده است. جدول



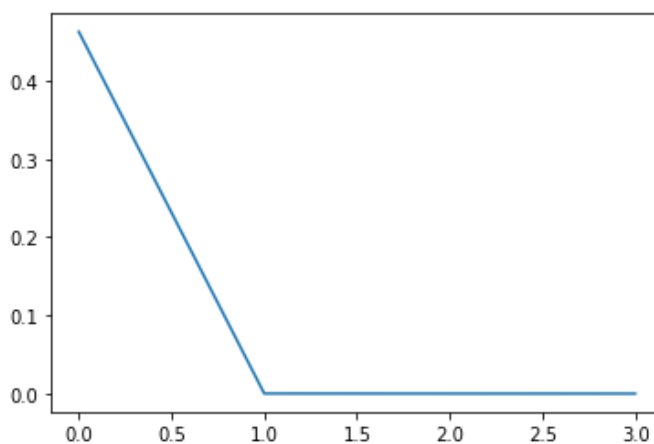
شکل ۹- ماتریس درهم ریختگی GodClass

نتایج شبکه عصبی مصنوعی برای برچسب‌گذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۹۸.۴۰٪ است. در بخش آموزش خطا ۰.۰۰۰۱ بود. همچنین مدت زمان آموزش ۱۰۰۱۱ میلی ثانیه است.

جدول ۴- نتایج مدل شبکه عصبی عمیق برای مجموعه داده DataClass

نتایج	معیار
۹۸.۴۰٪	دقت آزمایش
۰.۰۰۰۱	خطای آموزش
۱۰۰۱۱ ms	مدت زمان آموزش

همچنین نمودار خطای آموزش به صورت زیر است.



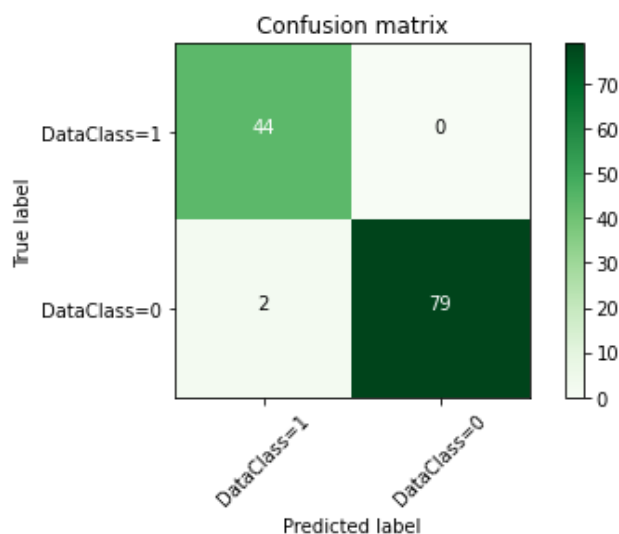
شکل ۱۰- نمودار خطای DataClass

معیارهای ارزیابی Recall - Precision و F1-score به صورت زیر است.

جدول ۵- معیارهای ارزیابی Recall - Precision و F1-score برای مجموعه داده DataClass

کلاس	precision	recall	f1-score
۰	1.00	0.98	0.99
۱	0.96	1.00	0.98

و ماتریس درهم ریختگی آن نیز به صورت زیر است.



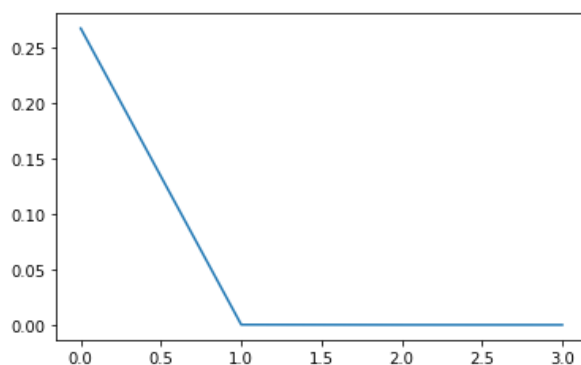
شکل ۱۱- ماتریس درهم ریختگی DataClass

نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده LongMethod در جدول ۶ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برچسب‌گذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۹۷.۵۲٪ است. در بخش آموزش خطا ۰.۰۰۰۲ بود. همچنین مدت زمان آموزش ۱۶۱۲۳ میلی ثانیه است.

جدول ۶- نتایج مدل شبکه عصبی عمیق برای مجموعه داده LongMethod

نتایج	معیار
۹۷.۵۲٪	دقت آزمایش
۰.۰۰۰۲	خطای آموزش
۱۶۱۲۳ ms	مدت زمان آموزش

همچنین نمودار خطای آموزش به صورت زیر است. معیارهای ارزیابی Recall - Precision و F1-score به صورت زیر است.

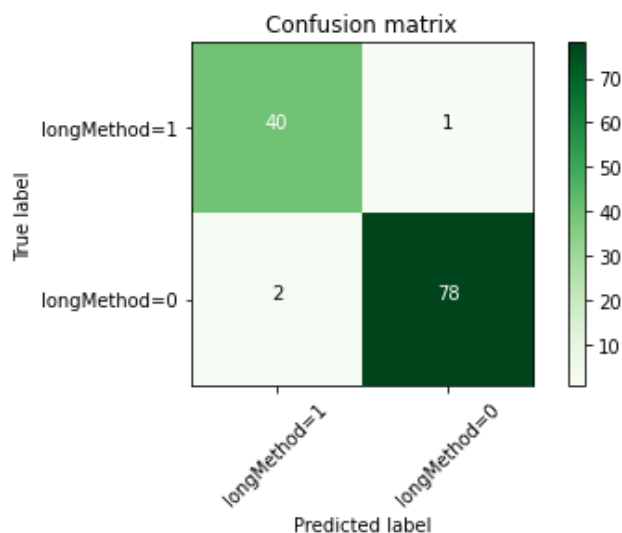


شکل ۱۲- نمودار خطای LongMethod

جدول ۷- معیارهای ارزیابی **Recall - Precision** و **F1-score** برای مجموعه داده **LongMethod**

کلاس	precision	recall	f1-score
۰	0.99	0.97	0.98
۱	0.95	0.98	0.96

و ماتریس درهم ریختگی آن نیز به صورت زیر است.



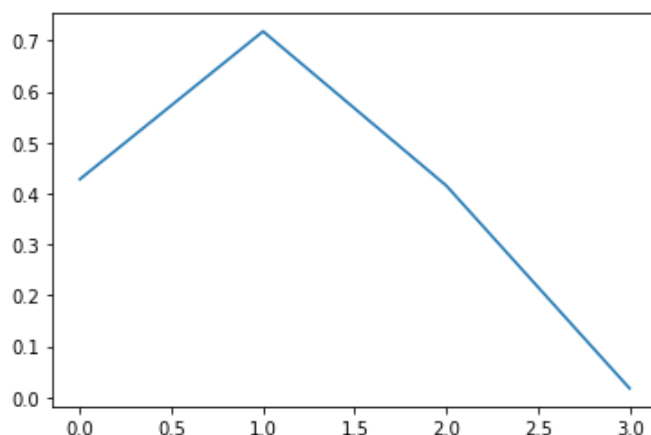
شکل ۱۳- ماتریس درهم ریختگی **LongMethod**

نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده **FeatureEnvy** در جدول ۸ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برچسب‌گذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۸۷.۷۰٪ است. در بخش آموزش خطا ۰.۰۱۷۳ بود. همچنین مدت زمان آموزش ۷۸۵۵ میلی ثانیه است.

جدول ۸- نتایج مدل شبکه عصبی عمیق برای مجموعه داده **FeatureEnvy**

نتایج	معیار
۸۷.۷۰٪	دقت آزمایش
۰.۰۱۷۳	خطای آموزش
۷۸۵۵ ms	مدت زمان آموزش

همچنین نمودار خطای آموزش به صورت زیر است.



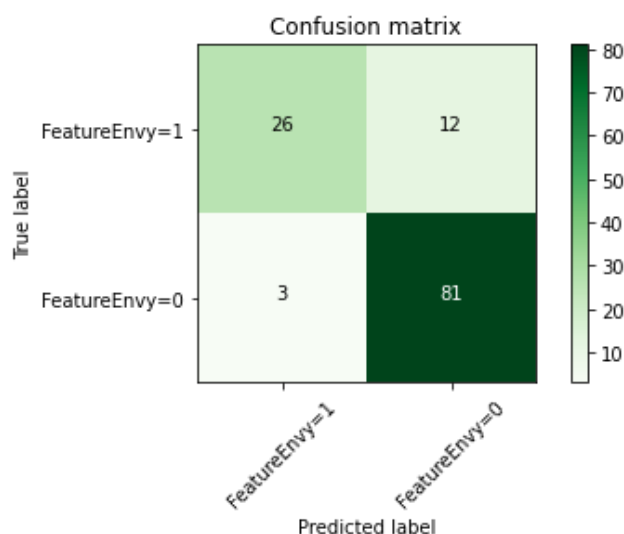
شکل ۱۴- نمودار خطی FeatureEnvy

معیارهای ارزیابی Recall - Precision و F1-score به صورت زیر است.

جدول ۹- معیارهای ارزیابی Recall - Precision و F1-score برای مجموعه داده FeatureEnvy

کلاس	precision	recall	f1-score
۰	0.87	0.96	0.92
۱	0.90	0.68	0.78

و ماتریس درهم ریختگی آن نیز به صورت زیر است.



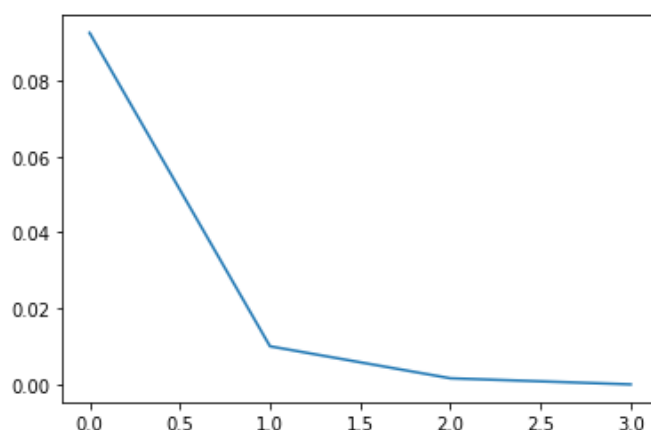
شکل ۱۵- ماتریس درهم ریختگی FeatureEnvy

نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده SwitchStatements در جدول ۱۰ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برجسبگذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۸۰.۶۵٪ است. در بخش آموزش خطا ۰.۰۰۰۱ بود. همچنین مدت زمان آموزش ۷۳۰۲ میلی ثانیه است.

جدول ۱۰- نتایج مدل شبکه عصبی عمیق برای مجموعه داده SwitchStatements

نتایج	معیار
۸۰.۶۵٪	دقت آزمایش
۰.۰۰۰۱	خطای آموزش
۷۳۰۲ ms	مدت زمان آموزش

همچنین نمودار خطای آموزش به صورت زیر است.



شکل ۱۶- نمودار خطای SwitchStatements

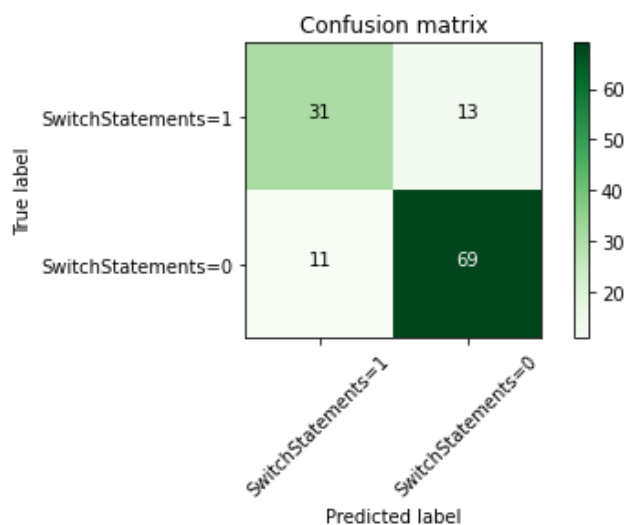
معیارهای ارزیابی Recall - Precision و F1-score به صورت زیر است.

جدول ۱۱- معیارهای ارزیابی Recall - Precision و F1-score برای مجموعه داده SwitchStatements

کلاس	precision	recall	f1-score
۰	0.84	0.86	0.85
۱	0.74	0.70	0.72

و ماتریس درهم ریختگی آن نیز به صورت زیر است.

نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای مجموعه داده LongParameterList در جدول ۱۲ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برچسب‌گذاری دودویی را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی ۸۸.۶۲٪ است. در بخش آموزش خطا ۰.۰۳۸۸ بود. همچنین مدت زمان آموزش ۸۹۸۸ میلی ثانیه است.

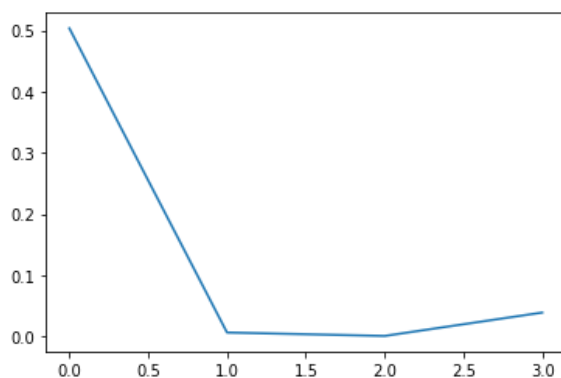


شکل ۱۷- ماتریس درهم ریختگی SwitchStatements

جدول ۱۲- نتایج مدل شبکه عصبی عمیق برای مجموعه داده LongParameterList

نتایج	معیار
۸۸.۶۲٪	دقت آزمایش
۰.۰۳۸۸	خطای آموزش
۸۹۸۸ ms	مدت زمان آموزش

همچنین نمودار خطای آموزش به صورت زیر است.



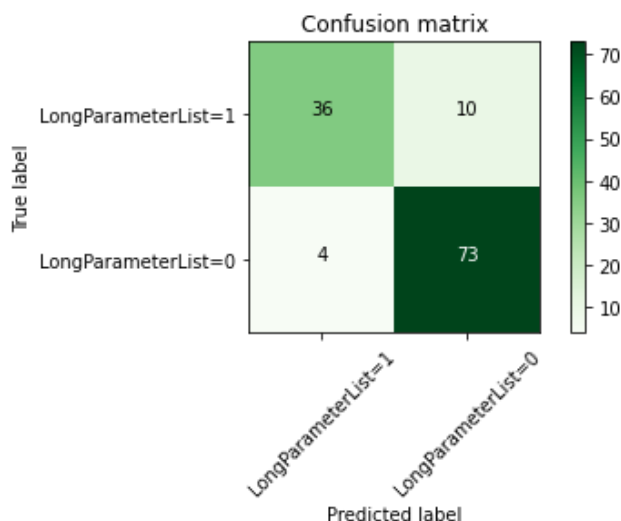
شکل ۱۸- نمودار خطای LongParameterList

معیارهای ارزیابی Recall - Precision و F1-score به صورت زیر است.

جدول ۱۳- معیارهای ارزیابی **Recall - Precision** و **F1-score** برای مجموعه داده **LongParameterList**

کلاس	precision	recall	f1-score
۰	0.88	0.95	0.91
۱	0.90	0.78	0.84

و ماتریس درهم ریختگی آن نیز به صورت زیر است.



شکل ۱۹- ماتریس درهم ریختگی **LongParameterList**

نتیجه‌گیری

این تحقیق مدل یادگیری عمیق مبتنی بر DNN را به عنوان یک سیستم تشخیص بود در کد پیشنهاد می‌کند. در این کار از شش مجموعه داده که شامل بوهای مختلف بود استفاده کردیم تا از آن‌ها با مدل یادگیری عمیق برای شناسایی الگوهای غیرعادی استفاده کنیم. نتایج نشان می‌دهد که دقت بدست آمده از این روش، مطلوب و قابل قبول است.

محل پروژه

<https://github.com/tahaa1377/smell-detection-with-deep-learning>

منابع

- [1] Aleesa, A.; Zaidan, B.; Zaidan, A.; and Sahar, N.M. (2020). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, 32(14), 9827-9858.
- [2] Sharma, T., Efstathiou, V., Louridas, P., & Spinellis, D. (2021). Code smell detection by deep direct-learning and transfer-learning. *Journal of Systems and Software*, 176, 110936.