**بررسی مقاله** Transfer learning for cross-company software defect prediction

در این مقاله(Ma, Luo, Zeng, & Chen, 2012)، آقای ما و همکاران الگوریتمی جدید به نام transfer naïve bayes را ارائه کرده‌اند که برای پیش‌بینی خطای بین پروژه‌ای[[1]](#footnote-1) مورداستفاده قرار می‌گیرد. این الگوریتم با استفاده از داده‌های مقصد[[2]](#footnote-2)، به هریک از نمونه‌های داده‌های آموزشی[[3]](#footnote-3)، وزنی را نسبت داده و سپس یک مدل بیزین ساده[[4]](#footnote-4) را از روی آن‌ها می‌سازد. درنهایت برچسب داده‌های مقصد، از طریق معادلاتی که در مقاله ارائه شده است، پیش‌بینی می‌شود.

برای پیاده‌سازی مقاله از دیتاست های جدول 1(table 1 از مقاله) استفاده شده است. به‌جز دیتاست kc1، بقیه دیتاست­ها دقیقاً همان دیتاست های مشخص‌شده در جدول 1 می‌باشند. از بین نسخه‌های مختلف دیتاست kc1 نسخه‌ای انتخاب شد که بیشترین شباهت را ازنظر تعداد رکورد و درصد خطا دارا باشد (دیتاست موجود در مقاله 1212 رکورد با درصد خطای 26.00 و دیتاست استفاده شده 1162 رکورد با درصد خطای 25.30 را داراست).

محیط پیاده‌سازی مقاله، نرم‌افزار وکا می‌باشد. به دلیل عدم تسلط به این نرم‌افزار و موجود نبودن کد منبع مقاله برای این نرم­افزار، از زبان برنامه‌نویسی R استفاده شده است. بااین‌وجود از کتابخانه Rweka که شامل روش‌های موجود در نرم‌افزار وکا می‌باشد، استفاده شده است. درواقع دسته‌بند استفاده شده مشابه دسته‌بند مقاله اما در یک محیط برنامه‌نویسی متفاوت می‌باشد.

آنچه در ادامه این گزارش می‌آید بدین‌صورت می‌باشد. در بخش اول به مقایسه نتایج موجود در مقاله با نتایج پیاده‌سازی انجام‌شده می‌پردازیم. در بخش دوم به بررسی نتایج بدون استفاده از فیلتر لگاریتم (log-filter) می‌پردازیم. سپس بررسی خواهیم کرد که چه مقدار از بهبود حاصل‌شده به خاطر روش TNB و چه مقدار به خاطر گسسته سازی است و در پایان گزارش به نتیجه‌گیری خواهیم پرداخت.

**مقایسه نتایج مقاله با نتایج پیاده‌سازی انجام‌شده**

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | CC | NN-filter | TNB |
|  | ar3 |  |  |
| AUC | 0.5009±0.0000 | 0.6511±0.0001 | 0.7139±0.0002 |
| pd | 1.0000±0.0000 | 0.8750±0.0000 | 0.8750±0.0000 |
| pf | 0.9982±0.0000 | 0.5727±0.0003 | 0.4473±0.0007 |
| F-measure | 0.2257±0.0000 | 0.3012±0.0000 | 0.3540±0.0002 |
|  | ar4 |  |  |
| AUC | 0.5264±0.0000 | 0.6170±0.0000 | 0.6920±0.0001 |
| pd | 1.0000±0.0000 | 0.8500±0.0000 | 0.8300±0.0006 |
| pf | 0.9471±0.0001 | 0.6161±0.0001 | 0.4460±0.0008 |
| F-measure | 0.3268±0.0000 | 0.3753±0.0000 | 0.4405±0.0001 |
|  | ar5 |  |  |
| AUC | 0.5036±0.0001 | 0.7839±0.0001 | 0.8268±0.0002 |
| pd | 1.0000±0.0000 | 1.0000±0.0000 | 1.0000±0.0000 |
| pf | 0.9929±0.0002 | 0.4321±0.0004 | 0.3464±0.0008 |
| F-measure | 0.3653±0.0000 | 0.5696±0.0001 | 0.6232±0.0004 |

جدول 1 – نتایج تجربی مقاله: مقایسه AUC، pd، pf و F-measure با استفاده از دیتاست های NASA به‌عنوان داده‌های آموزشی و هریک از دیتاست های SOFTLAB به‌عنوان داده مقصد (این جدول معادل جدول 4 از مقاله می‌باشد).

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | NN | TNB |
|  | ar3 |  |  |
| AUC | 0.5236±0.0156 | 0.6648±0.0000 | 0.7076±0.027 |
| pd | 1.0000±0.0000 | 0.8750±0.0000 | 0.8625±0.0395 |
| pf | 0.9527±0.0311 | 0.5455±0.0000 | 0.4473±0.0563 |
| F-measure | 0.2341±0.0059 | 0.3111±0.0000 | 0.3508±0.0235 |
|  | ar4 |  |  |
| AUC | 0.5603±0.0102 | 0.5514±0.0000 | 0.7305±0.0272 |
| pd | 1.0000±0.0000 | 0.8500±0.0000 | 0.9000±0.0000 |
| pf | 0.8793±0.0205 | 0.7471±0.0000 | 0.4391±0.0544 |
| F-measure | 0.3434±0.0052 | 0.3333±0.0000 | 0.4741±0.0302 |
|  | ar5 |  |  |
| AUC | 0.5393±0.0164 | 0.7679±0.0000 | 0.8295±0.0299 |
| pd | 1.0000±0.0000 | 1.0000±0.0000 | 0.9375±0.0659 |
| pf | 0.9214±0.0328 | 0.4643±0.0000 | 0.2786±0.0151 |
| F-measure | 0.3829±0.0085 | 0.5517±0.0000 | 0.6432±0.0267 |

جدول 2 – نتایج تجربی پیاده‌سازی انجام‌شده: مقایسه AUC، pd، pf و F-measure با استفاده از دیتاست های NASA به‌عنوان داده‌های آموزشی و هریک از دیتاست های SOFTLAB به‌عنوان داده مقصد.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | NN | TNB |
|  | ar3 |  |  |
| AUC | 0.0227 | 0.0137 | -0.0063 |
| pd | 0.0000 | 0.0000 | -0.0125 |
| pf | -0.0455 | -0.0272 | 0.0000 |
| F-measure | 0.0084 | 0.0099 | -0.0032 |
|  | ar4 |  |  |
| AUC | 0.0339 | -0.0656 | 0.0385 |
| pd | 0.0000 | 0.0000 | 0.0700 |
| pf | -0.0678 | 0.1310 | -0.0069 |
| F-measure | 0.0166 | -0.0420 | 0.0336 |
|  | ar5 |  |  |
| AUC | 0.0357 | -0.0160 | 0.0027 |
| pd | 0.0000 | 0.0000 | -0.0625 |
| pf | -0.0715 | 0.0322 | -0.0678 |
| F-measure | 0.0176 | -0.0179 | 0.0200 |

جدول 3 – اختلاف مقادیر میانگین جدول 1 و 2: meantable2-meantable1

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | NN | TNB |
|  | ar3 |  |  |
| AUC | 0.0156 | -0.0001 | 0.0268 |
| pd | 0.0000 | 0.0000 | 0.0395 |
| pf | 0.0311 | -0.0003 | 0.0556 |
| F-measure | 0.0059 | 0.0000 | 0.0233 |
|  | ar4 |  |  |
| AUC | 0.0102 | 0.0000 | 0.0271 |
| pd | 0.0000 | 0.0000 | -0.0006 |
| pf | 0.0204 | -0.0001 | 0.0536 |
| F-measure | 0.0052 | 0.0000 | 0.0301 |
|  | ar5 |  |  |
| AUC | 0.0163 | -0.0001 | 0.0297 |
| pd | 0.0000 | 0.0000 | 0.0659 |
| pf | 0.0326 | -0.0004 | 0.0143 |
| F-measure | 0.0085 | -0.0001 | 0.0263 |

جدول 4 – اختلاف مقادیر انحراف از معیار جدول 1 و 2: stdtable2-stdtable1

جدول 3 اختلاف مقادیر میانگین جدول 1 و 2 را نشان می‌دهد. اعداد صفر، مثبت و منفی به ترتیب نشان نشان‌دهنده یکسان بودن دقت پیاده‌سازی انجام‌شده و مقاله، مقادیر دقت بالاتر و پایین‌تر از مقاله می‌باشد. خلاصه‌ای از مقادیر به‌دست‌آمده در جدول 3 در زیر آمده است:

|  |  |  |  |
| --- | --- | --- | --- |
| AUC | pd | pf | F-measure |
| Min.:-0.065600 | Min.:-0.0625000 | Min.:-0.07150 | Min.:-0.042000 |
| Mean: 0.006589 | Mean:-0.0005556 | Mean:-0.01372 | Mean: 0.004778 |
| Max.: 0.038500 | Max.: 0.0700000 | Max.: 0.13100 | Max.: 0.033600 |

جدول 4 – کمترین، میانگین و بیشترین اختلاف مقادیر به‌دست‌آمده در پیاده‌سازی مقاله با مقادیر موجود در مقاله.

برای مثال مقادیر مربوط به ستون AUC در جدول 4 نشان می‌دهد، مقادیری که برای AUC در پیاده‌سازی به‌دست‌آمده در بدترین حالت حدود 6.6 درصد پایین‌تر، در بهترین حالت حدود 3.9 درصد بالاتر و به‌صورت میانگین 0.7 درصد بالاتر از مقاله بوده است. میانگین مقادیر موجود در جدول 3، 0.07 درصد و میانگین قدر مطلق این مقادیر، 2.8 درصد می‌باشد که مقدار معقولی است. تفاوت مقادیر پیاده‌سازی با مقادیر مقاله به دلیل انتخاب تصادفی 90 درصدی داده‌های آموزشی (جزئیات بیشتر در قسمت 4.3 مقاله موجود است.)، تفاوت زبان برنامه‌نویسی و تفاوت دیتاست kc1 که بخشی از دیتاست آموزشی است، می‌باشد.

**پیش‌بینی خطا بدون استفاده از فیلتر لگاریتم**

در این قسمت به بررسی نتایج بدون استفاده از فیلتر لگاریتم خواهیم پرداخت. در ابتدا به مقایسه روش‌های CC، NN و TNB بدون استفاده از این فیلتر پرداخته سپس نتایج را با حالتی که از فیلتر لگاریتم استفاده شده است مقایسه می‌کنیم.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | NN | TNB |
|  | ar3 |  |  |
| AUC | 0.7259±0.0047 | **0.7386±0.0000** | 0.7031±0.0299 |
| pd | 0.7500±0.0000 | 0.7500±0.0000 | **0.8625±0.0395** |
| pf | 0.2982±0.0094 | **0.2727±0.0000** | 0.4564±0.0638 |
| F-measure | 0.3948±0.0067 | **0.4138±0.0000** | 0.3468±0.0263 |
|  | ar4 |  |  |
| AUC | 0.7273±0.0065 | 0.7080±0.0000 | **0.7439±0.0269** |
| pd | 0.5650±0.0242 | 0.6000±0.0000 | **0.8900±0.0316** |
| pf | **0.1103±0.0111** | 0.1839±0.0000 | 0.4023±0.0539 |
| F-measure | **0.5524±0.0039** | 0.5000±0.0000 | 0.4905±0.0292 |
|  | ar5 |  |  |
| AUC | **0.8536±0.0086** | 0.8482±0.0000 | 0.8062±0.0270 |
| pd | 0.8750±0.0000 | 0.8750±0.0000 | **0.9125±0.0604** |
| pf | **0.1679±0.0173** | 0.1786±0.0000 | 0.3000±0.0452 |
| F-measure | **0.7111±0.0178** | 0.7000±0.0000 | 0.6166±0.0289 |

جدول 5 - نتایج تجربی پیاده‌سازی مقاله: مقایسه AUC، pd، pf و F-measure بدون اعمال فیلتر لگاریتم (log\_filter). مقادیر Bold شده بهترین دقت‌ها را نشان می‌دهد.

جدول 5 نتایج به‌دست‌آمده بدون استفاده از فیلتر لگاریتم[[5]](#footnote-5) را نشان می‌دهد. آقای منزیس و همکاران استفاده از این فیلتر را به‌منظور نرمال‌سازی مقادیر متریک‌هایی که توزیع نمایی دارند، پیشنهاد دادند(Menzies, Greenwald, & Frank, 2007). با مقایسه این جدول با جدول 1 مشاهده می‌کنیم که برای دو روش CC و NN، معیارهای AUC، pf و F-measure بدون استفاده از فیلتر لگاریتم نتایج بهتری خواهند داشت اما برای معیار pd با استفاده از فیلتر لگاریتم به نتایج بهتری خواهیم رسید. برای روش TNB نیز استفاده از فیلتر لگاریتم باعث بهبود حدود یک‌درصدی خواهد شد.

با مقایسه جدول 5 با جدول 2 که به ترتیب نتایج بدون استفاده از فیلتر لگاریتم و با استفاده از آن می‌باشد می‌بینیم که در حالت کلی دقت روش CC (به‌جز برای معیار pd) با استفاده از این فیلتر کاهش قابل‌توجهی خواهد داشت و بخشی از بهبودی که در مقاله برای روش TNB به آن اشاره‌شده به این خاطر می‌باشد.

در قسمتی از مقاله(Ma et al., 2012)، آقای ما و همکاران در مورد نتایج موجود در جدول 4 از مقاله‌شان که معادل جدول 1 از این گزارش می‌باشد، چنین توضیح می‌دهند که برای همه دیتاست های مقصد، TNB به‌صورت قابل‌توجهی مقادیر AUC، F-measure و pf بالاتری را نسبت به روش NN به دست می‌آورد و به ازای معیار pd نیز تنها در یک مورد از سه مورد ضعیف‌تر از NN عمل می‌کند. این درصورتی است که اگر از فیلتر لگاریتم بر روی دیتاست استفاده نشود، باوجود مقادیر بهتر pd برای هر 3 دیتاست مقصد و یک مقدار بهتر AUC، روش TNB مقادیر F-measure و pf پایین‌تری نسبت به NN خواهد داشت.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | NN | TNB |
| AUC mean | 0.**7689** | 0.7650 | 0.7511 |
| pd mean | 0.7300 | 0.7417 | 0.**8883** |
| pf mean | 0.**1921** | 0.2117 | 0.3862 |
| F-measure mean | 0.**5528** | 0.5379 | 0.4846 |

جدول 6 – میانگین مقادیر موجود در جدول 5. مقادیر Bold شده بهترین دقت‌ها را نشان می‌دهد.

همان‌طور که در جدول 6 مشخص است درصورتی‌که از فیلتر لگاریتم استفاده نشود، TNB تنها مقدار pd بهتری را نسبت به 2 روش NN و CC برای 3 دیتاست ar3، ar4 و ar5 به دست می‌آورد. درواقع این روش تعداد بیشتری از رکوردها را برچسب خطادار زده و موجب بالا رفتن مقدار pd و pf می‌شود. حتی اگر از فیلتر لگاریتم برای روش TNB استفاده شود، دقت‌ها در بهترین حالت 2 درصد بهتر شده و باز مقدار AUC، pf و F-measure کمتری نسبت به دو روش NN و CC حاصل می‌شود. می‌توان چنین نتیجه گرفت که استفاده از روش TNB تنها باعث بهبود معیار pd می‌شود و آنچه در مقاله نتیجه گرفته‌شده بدون در نظر گرفتن عدم اعمال فیلتر لگاریتم بوده و در حالت کلی دارای ایراد است.

**مقایسه روش TNB با روش CC به همراه گسسته سازی**

یکی از مراحل روش TNB، گسسته سازی است که مقدار زیادی از بهبود مشاهده‌شده در این روش به خاطر آن می‌باشد. برای مشخص کردن اینکه چه مقدار از بهبود حاصل‌شده به خاطر معادلات تعریف‌شده برای روش TNB و چه مقدار به خاطر گسسته سازی است، به مقایسه روش TNB و Disc\_CC می‌پردازیم. در روش Disc\_CC ابتدا داده‌ها را گسسته سازی کرده سپس روش CC را اعمال می‌کنیم. درواقع تفاوت دو روش در این است که روش TNB پس از اعمال گسسته سازی از طریق محاسبه وزن نمونه‌های آموزشی و معادلات تعریف‌شده در مقاله(Ma et al., 2012) به پیش‌بینی می‌پردازد اما روش Disc\_CC پس از گسسته سازی از طریق بیزین ساده برچسب‌ها را پیش‌بینی می‌کند.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | Disc\_CC | TNB |
|  | ar3 |  |  |
| AUC | 0.5236±0.0156 | 0.6811±0.0153 | **0.7076±0.0270** |
| pd | 1.0000±0.0000 | **0.8750±0.0000** | 0.8625±0.0395 |
| pf | 0.9527±0.0311 | 0.5127±0.0307 | **0.4473±0.0563** |
| F-measure | 0.2341±0.0059 | 0.3245±0.0131 | 0.3508±0.0235 |
|  | ar4 |  |  |
| AUC | 0.5603±0.0102 | **0.7609±0.0057** | 0.7305±0.0272 |
| pd | **1.0000±0.0000** | 0.9000±0.0000 | 0.9000±0.0000 |
| pf | 0.8793±0.0205 | **0.3782±0.0114** | 0.4391±0.0544 |
| F-measure | 0.3434±0.0052 | **0.5078±0.007** | 0.4741±0.0302 |
|  | ar5 |  |  |
| AUC | 0.5393±0.0164 | 0.7911±0.0121 | **0.8295±0.0299** |
| pd | **1.0000±0.0000** | **1.0000±0.0000** | 0.9375±0.0659 |
| pf | 0.9214±0.0328 | 0.4179±0.0241 | **0.2786±0.0151** |
| F-measure | 0.3829±0.0085 | 0.5779±0.0147 | **0.6432±0.0267** |

جدول 7 – مقایسه CC، Disc\_CC و TNB. مقادیر Bold شده بهترین دقت‌ها را نشان می‌دهند.

آنچه در جدول 7 آمده است برای زمانی است که فیلتر لگاریتم اعمال شود. هنگامی‌که فیلتر لگاریتم اعمال می‌شود دقت CC کاهش پیداکرده اما دقت TNB حدود 1 درصد افزایش پیدا می‌کند درنتیجه جدول 7 به‌صورت خوش‌بینانه در نظر گرفته شده است وگرنه همان‌طور که در قسمت قبل اشاره شد روش TNB به‌صورت کلی باعث بهبود دقت CC نخواهد شد. با بررسی جدول 7 مشاهده می‌کنیم که روش TNB تنها در 5 مورد از 12 مورد بهترین دقت‌ها را داراست و در دو مورد حتی دقت پایین‌تری را نسبت به روش CC به دست می‌آورد.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CC | Disc\_CC | TNB |
| AUC | 0.5411 | 0.7444 | **0.7558** |
| pd | **1.0000** | 0.9250 | 0.9000 |
| pf | 0.9178 | 0.4362 | **0.3883** |
| F-measure | 0.3201 | 0.4701 | **0.4894** |

جدول 8 – میانگین مقادیر موجود در جدول 7. مقادیر Bold شده بهترین دقت‌ها را نشان می‌دهد.

همان‌طور که در جدول 8 مشخص است TNB مقادیر بهتری را برای سه معیار AUC، pf و F-measure برای 3 دیتاست ar3، ar4 و ar5 به دست می‌آورد و تنها برای معیار pd به‌صورت میانگین 10 درصد کاهش را داراست.

|  |  |  |
| --- | --- | --- |
|  | Disc\_CC | TNB |
| AUC | +0.2033(94.69%) | +0.2147(05.31%) |
| pd | -0.075(\*) | -0.1000(\*) |
| pf | +0.4816(90.95%) | +0.5295(09.05%) |
| F-measure | +0.1500(88.60%) | +0.1693(11.40%) |

جدول 9 –میزان بهبود روش‌های Disc\_CC و TNB نسبت به روش CC.

اعداد مثبت و منفی موجود در جدول 9 به ترتیب نشان‌دهنده دقت بالاتر و پایین‌تر آن روش نسب به CC می‌باشد. اعداد داخل پرانتز مشخص می‌کند که چه درصدی از بهبود به خاطر روش موردنظر است. برای مثال 94.69 درصد بهبود به‌دست‌آمده برای معیار AUC به خاطر گسسته سازی (Disc\_CC) و 5.31 درصد از بهبود به خاطر روش TNB می‌باشد. همان‌طور که در جدول مشخص است بیشتر بهبود به‌دست‌آمده برای روش TNB به خاطر استفاده از گسسته سازی بوده و وزن دهی و معادلات تعریف‌شده در مقاله درصد کمی از بهبود را به خود اختصاص می‌دهند (برای حالاتی که دقت کاهش‌یافته است از \* استفاده شده است).

**نتیجه‌گیری**

با توجه به مواردی که در این گزارش ذکر شد، مشخص شد که نتایجی که آقای ما و همکاران در مقاله به آن اشاره کرده‌اند بدون در نظر گرفتن عدم اعمال فیلتر لگاریتم بوده و درصورتی‌که از این فیلتر استفاده نشود و داده‌ها بدون تغییر به دسته‌بند بیزین ساده داده شوند، مقادیر مربوط به AUC، pf و F-measure بهتر خواهند شد و در این حالت روش TNB، چه با استفاده از فیلتر لگاریتم و چه بدون استفاده از آن در بیشتر موارد نخواهد توانست روش NN و CC را شکست دهد و نتایج به آن خوبی که در مقاله اشاره شده‌اند نخواهد بود. علاوه بر این، بخش قابل‌توجهی از بهبود به‌دست‌آمده به خاطر گسسته سازی می‌باشد و خود روش TNB سهم کمی از بهبود را به خود اختصاص می‌دهد.

Ma, Y., Luo, G., Zeng, X., & Chen, A. (2012). Transfer learning for cross-company software defect prediction. *Information and Software Technology*, *54*(3), 248–256. https://doi.org/10.1016/j.infsof.2011.09.007

Menzies, T., Greenwald, J., & Frank, A. (2007). Data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering*, (1), 2–13.

1. cross-project defect prediction [↑](#footnote-ref-1)
2. target [↑](#footnote-ref-2)
3. train [↑](#footnote-ref-3)
4. naive bayes [↑](#footnote-ref-4)
5. Log filter [↑](#footnote-ref-5)