آکادمی داده
dataacademy.ir

روش جدیدی برای داده کاوی مبتنی بر مجموعه های دانه درشت و پردازش تکاملی

عبدالرضا سالاری[1] مهران محسن زاده[1] بابک نصیری[1]

۱-دانشگاه آزاد اسلامی ، واحد علوم و تحقیقات

دانشکده فنی - گروه کامپیوتر

abishSalar@yahoo.com
m_mohsenzadeh77@yahoo.com
nasiri_babak@yahoo.com

محمد تشنه لب[2]

۲- دانشگاه خواجه نصیرالدین طوسی، دانشکده برق

teshnehlab@eetd.kntu.ac.ir

چکیده :

درخت تصمیم یکی از روشهای طبقه بندی اطلاعات در داده کاوی می باشد که به دلیل سادگی و قابلیت تفهیم ، بسیار مورد استفاده قرار می گیرد . در این مقاله روش جدیدی برای تولید درخت تصمیم[1] مبتنی بر پردازش تکاملی و مجموعه دانه درشت ارائه شده است . ابتدا برروی مجموعه داده ها ، عملیات پیش پردازش شامل استفاده از مقادیر عمومی به جای مقادیر گمشده[2] و گسسته سازی مشخصه ها[3] انجام ، سپس با بهره گیری از تئوری مجموعه های دانه درشت[4] و پردازش تکاملی ، مشخصه های مناسب استخراج می شوند و در انتها با استفاده از پردازش تکاملی درخت تصمیم ساخته می شود . نتایج بدست آمده بر روی مجموعه داده های مختلف که دارای مشخصه های زیادی هستند ، در مقایسه با نتایج معتبر ارائه شده ، بسیار مطلوب بوده است.

dataacademy.ir

کلمات کلیدی : داده کاوی ، درخت تصمیم ، مجموعه دانه درشت ، پردازش تکاملی ، استخراج مشخصه ها[5]

۱- مقدمه

داده کاوی یکی از مهمترین مراحل در فرآیند اکتشاف دانش بشمار می رود و وظیفه آن استخراج دانش و الگوهایی است که بصورت ضمنی در داده ها وجود دارد . داده کاوی بر روی مجموعه داده های بزرگ هزینه محاسباتی زیادی دارد ، به همین علت نیاز است که مشخصه های اصلی داده ها از بین کل مشخصه ها ، قبل از عمل داده کاوی استخراج شود .

---

[1] Decision Tree
[2] Missing Values
[3] Feature Discretization
[4] Rough Set Theory
[5] Feature Selection

در دهه گذشته روشهای مختلفی برای استخراج مشخصه ها پیشنهاد شده است که به طور کلی می توان آنها را به دو گروه زیر تقسیم نمود . الگوریتم های رتبه بندی مشخصه ها و الگوریتم های یافتن زیرمجموعه مینیمم . در روش رتبه بندی مشخصه ها ، به هر مشخصه یک رتبه نسبت می دهند . در این روش زیرمجموعه مینیمم از مشخصه ها برای آنالیز های بعدی مشخص نمی شود و فقط فاصله بین مشخصه ها بدست می آید اما در روش یافتن زیر مجموعه مینیمم ، این موضوع بصورت معکوس وجود دارد . *[10]*

برای تولید درخت تصمیم نیز روش های مختلفی پیشنهاد شده است که از بین آنها می توان به روشهای *ID3* ، *C4.5* و پردازش تکاملی اشاره نمود .

در این مقاله روش جدیدی برای داده کاوی با استفاده از درخت تصمیم مبتنی بر پردازش تکاملی ارائه شده است . همچنین از پردازش تکاملی برای استخراج مشخصه های داده ها از داده های حجیم استفاده گردیده است . روش پیشنهادی پیاده سازی و نتایج اجرا با سایر روشها مقایسه شده است .

مجموعه داده های استفاده شده برای ارزیابی روش برگرفته از کار گروه های تحقیقاتی بر روی داده کاوی و وب کاوی است که در گردهمایی *COIL 2000 [3]* به مسابقه گذاشته شده است . نتایج بدست آمده از اجرای روش ارائه شده در این مقاله نسبت به نتایج روشهای مشابه بسیار مطلوب بوده است .

روش ارائه شده دارای سه قدم اصلی است . پیش پردازش ، استخراج مشخصه ها و ایجاد درخت تصمیم . هر یک از این قدم ها تشریح شده است . در بخش دوم مقاله عملیات مورد نیاز در پیش پردازش داده ها مورد بررسی قرار گرفته است . در بخش سوم روشی برای بهبود استخراج مشخصه ها ، با استفاده از تئوری مجموعه های دانه درشت و پردازش تکاملی مورد استفاده قرار گرفته است . در بخش چهارم روشی برای تولید درخت تصمیم با استفاده از پردازش تکاملی توضیح داده شده است . در بخش پنجم نتایج حاصل از پیاده سازی روش پیشنهادی بر روی مجموعه داده های مختلف مورد ارزیابی قرار گرفته است و بخش آخر به نتیجه گیری و پیشنهادهایی برای بهبود تخصیص دارد .

۲- پیش پردازش

در این بخش پیش پردازش های لازم بر روی داده ها ، شامل عملیاتی برای پاکسازی داده ها[6] نظیر استفاده از مقدار عمومی به جای مقادیر گمشده و گسسته سازی مشخصه ها تشریح شده است .

روشهای مختلفی برای مدیریت نمودن مقادیر کمشده وجود دارد که می توان به حذف رکورد مربوطه (در صورتیکه مشخصه گمشده ، مشخصه کلاس باشد) ، پرکردن مقدار برای مشخصه گمشده به صورت دستی ، جایگزین نمودن یک مقدار عمومی به جای مقدار مشخصه گمشده ، جایگزین نمودن میانگین مقادیر موجود در ستون مشخصه گمشده به جای مقدار مشخصه گمشده اشاره نمود . در روش پیشنهادی از جایگزین نمودن یک مقدار عمومی به جای مقدار مشخصه گمشده استفاده شده است . *[10]*

موضوع بعدی گسسته سازی مشخصه ها - چگونگی تبدیل داده های با بازه بزرگتر به داده های با بازه کوچکتر است . برای حل این مسئله در روش پیشنهادی از الگوریتم *ChiMerge* [2] که از توزیع آماری $\chi 2$ برای گسسته سازی بازه های عددی استفاده می کند ، استفاده شده است . این الگوریتم شامل یک مرحله آغازین و یک فرآیند پائین به بالای اتصال می باشد ، که بازه ها به طور پیوسته تا هنگامی به هم متصل می شوند که شرط پایانی برقرار گردد . الگوریتم فوق در ابتدا با مرتب

[6] Data Cleansing

سازی داده های آموزشی که شامل مقادیری است که باید گسسته شوند ، شروع به کار می کند و اولین مجموعه گسسته شده با قرار گرفتن هر مقدار در بازه خود شکل می گیرد . فرمول محاسبه مقدار ۲χ به شرح زیر می باشد :

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{(A_{ij}-E_{ij})^2}{E_{ij}}$$

که در آن

m = ۲ (دو بازه با هم مقایسه می شوند)

k = تعداد کلاس ها

A_{ij} = تعداد نمونه ها در i امین بازه jامین کلاس

R_i = تعداد نمونه ها در i امین بازه = $\sum_{j=1}^{k} A_{ij}$

C_i = تعداد نمونه ها در jامین کلاس = $\sum_{j=1}^{m} A_{ij}$

N = مجموع تعداد نمونه ها = $A_{ij}\sum_{j=1}^{k} C_j$

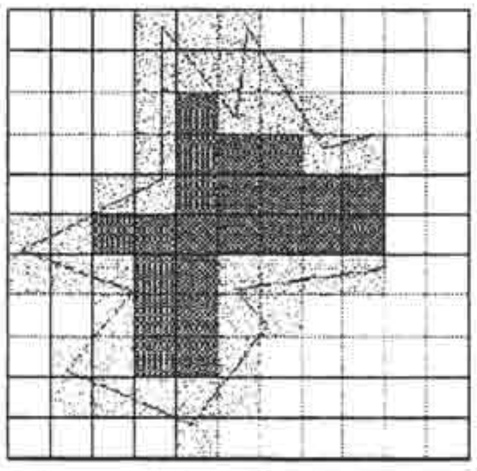
E_{ij} = فرکانس مورد نظر از $A_{ij} = \frac{R_i \times C_j}{N}$

می باشند.

۳- استخراج مشخصه ها با استفاده از مجموعه های دانه درشت و پردازش تکاملی

dataacademy.ir

تئوری مجموعه های دانه درشت بر پایه تقریب مفاهیم (مجموعه) با استفاده از دانش موجود در یک سیستم اطلاعاتی می باشد . از نقاط قوت این تئوری می توان به کارایی آن در مورد مقادیر گمشده اشاره نمود . تئوری مجموعه های دانه درشت برای پردازش داده های غیر کامل ، دو مجموعه تقریب بالا و تقریب پائین را تعریف می کند .



شکل (۱) : شمای کلی مجموعه های دانه درشت.

چنانچه X را مجموعه واقعی در نظر بگیریم ، تقریب بالا شامل اشیائی است که احتمالاً عضو X می باشند ، درحالیکه تقریب پائین شامل اشیائی است که قطعاً عضو X هستند . در صورتیکه تقریب بالا و پائین برای یک زیرمجموعه از سیستم اطلاعاتی برابر باشند آنگاه آن زیرمجموعه را قابل تعریف و در غیر این صورت آن مجموعه را بسختی[7] قابل تعریف می نامند.

[7] Rough

سیستم اطلاعاتی عبارت از یک زوج مرتب $S=(U,A\cup\{d\})$ که U یک مجموعه متناهی و غیر تهی (به نام مجموعه جهانی) و A یک مجموعه متناهی و غیر تهی از مشخصه های شرطی و d مشخصه تصمیم می باشد [4,5].

ماتریس تمایز نیز یک ماتریس متقارن با ابعاد $|U|\times|U|$ با درایه های C_{ij} می باشد که بصورت زیر تعریف می شود :

$$C_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \quad for \quad i, j = 1,..., n\}$$

تابع تمایز نیز با ترکیب فصلی کلیه مشخصه های موجود در C_{ij} و سپس ترکیب عطفی کلیه C_{ij} ها با یکدیگر ساخته می شود :

$$f_A(a_1,..., a_m) = \wedge\{\vee c_{ij} \mid 1 <= j <= i <= n, c_{ij} \neq 0\}$$

برای استخراج مشخصه های مناسب ، با استفاده از تئوری مجموعه های دانه درشت ، لازم است که ابتدا ماتریس تمایز ساخته شود و سپس تابع تمایز از روی این ماتریس بدست آید. با ساده سازی این تابع ، مجموعه کاهش یافته (شامل مشخصه های مناسب) بدست می آید. هنگامیکه حجم داده ها زیاد باشد ، ساده سازی تابع تمایز بدست آمده دارای هزینه محاسباتی بسیار بالایی می باشد. از اینرو بجای ساده سازی تابع تمایز ، از توانایی الگوریتم های ژنتیکی که جستجوی بهینه در فضاهای بزرگ است ، استفاده میشود.

ساختار کروموزوم ها بصورت یک رشته دودویی در نظر گرفته شده و تعداد ژن ها برابر با تعداد مشخصه ها در مجموعه داده می باشد . برای مثال ، اگر اشیاء یک سیستم اطلاعاتی دارای ۶ مشخصه(با مشخصه تصمیم) باشد ، کروموزوم مورد استفاده توسط این روش شامل ۶ ژن می باشد .

تابع تطابق استفاده شده در این روش ، برای انتخاب بهترین ها ، از ماتریس تمایز استفاده می کند. با فرض اینکه $b(i,(k,n))$ یک درایه از ماتریس تمایز مربوط به مشخصه i ام و دوتایی (o_k,o_n) باشد ، ماتریس تمایز بصورت زیر ساخته می شود [6,7] :

$$for \quad i \in \{1,.., N\}: \qquad b(i,(k,n)) = \begin{cases} 1 & a_i(o_k) \neq a_i(o_n) \\ 0 & a_i(o_k) = a_i(o_n) \end{cases}$$

$$b(N+1,(k,n)) = \begin{cases} 0 & d_i(o_k) \neq d_i(o_n) \\ 1 & d_i(o_k) = d_i(o_n) \end{cases}$$

تابع تطابق نیز بصورت زیر می باشد که در آن بخش اول تعداد یک های موجود در کروموزوم $r(N_r)$ و بخش دوم شامل تعداد سطرهای پوشش داده شده از ماتریس تمایز $B(C_r)$ می باشد .

$$F(r) = (1-\alpha) * Min\ (\varepsilon_1, \frac{N-N_r}{N}) + \alpha * Min\ (\varepsilon_2, \frac{C_r}{K}) \qquad where$$

$$K = \frac{m^2 - m}{2}$$

همچنین N تعداد مشخصه های اولیه ، m تعداد اشیاء در سیستم اطلاعاتی ، K تعداد سطرهای ماتریس تمایز ، ε_1 حداقل درصد مشخصه های حذف شده قابل قبول ، ε_2 حداقل درصد پوشش های داده شده مورد قبول و α میزان اهمیت به هر یک از دو بخش تابع می باشد .

٤- ساخت درخت تصمیم با استفاده از پردازش تکاملی

الگوریتم های متعددی برای ساخت درخت تصمیم همانند *ID3* ، *C4.5* وجود دارند . اینگونه الگوریتم ها هنگامیکه تعداد مشخصه ها زیاد باشند هزینه محاسباتی بالایی دارند . یکی از راه های ساخت درخت تصمیم استفاده از الگوریتم ژنتیک می باشد.

در روش پیشنهادی برای ساخت درخت تصمیم از الگوریتم های ژنتیک استفاده می شود . برای این منظور از الگوریتم ارائه شده در *[9]* استفاده شده است . برای محاسبه تابع تطابق ، هر درخت تولید شده توسط الگوریتم ، به طبقه بندی مجموعه داده های آزمایشی می پردازد . بدین ترتیب هر درختی که بتواند تعداد بیشتری از داده های آموزشی را بطور درست طبقه بندی کند ، دارای مقدار تابع تطابق بالاتری خواهد بود .

در ابتدا تعدادی درخت تصادفی تولید می شود ، سپس درخت هایی که دارای مقدار تابع تطابق بالاتری هستند ، انتخاب شده و با یکدیگر ترکیب می شوند ، تا نتیجه بهتری حاصل شود . همچنین برای همبری از یک نوع خاص استفاده شده است بدین ترتیب که ابتدا یک نقطه قطع انتخاب می شود و سپس زیر درخت از آن گره مشترک در دو درخت با یکدیگر تعویض می شوند (در صورتیکه در زیر درخت ها گره تکراری موجود نباشد) .

٥- پیاده سازی و ارزیابی روش پیشنهادی

در این بخش از مقاله نتایج حاصل از پیاده سازی روش پیشنهادی بر روی مجموعه داده های مختلف ارائه و مورد ارزیابی و مقایسه با سایر روشها قرار گرفته است.

مجموعه داده های انتخاب شده برای ارزیابی روش پیشنهادی شامل دو مجموعه *Adeater* و *Insurance* می باشند . مجموعه داده *Adeater* برگرفته از مشخصه های بکار گرفته شده در حذف عکس های تبلیغاتی در صفحات اینترنت می باشد *[5]*. این مجموعه داده دارای ۳ مشخصه عددی و ۱۵۵۵ مشغصه دودویی است . مجموعه داده *Insurance* نیز دارای ۸۵ مشخصه عددی می باشد ، که این داده ها برگرفته شده از اطلاعات بیمه گذاران می باشد و برای پیش بینی سرمایه گذاری در یک نوع بیمه خاص مورد استفاده قرار گرفته است . هر دو مجموعه مذکور همچنین ، در گردهمایی *COIL 2000 [3]* به مسابقه گذاشته شده است. در آن مسابقات روش های مختلف داده کاوی مورد ارزیابی قرار می گیرند.

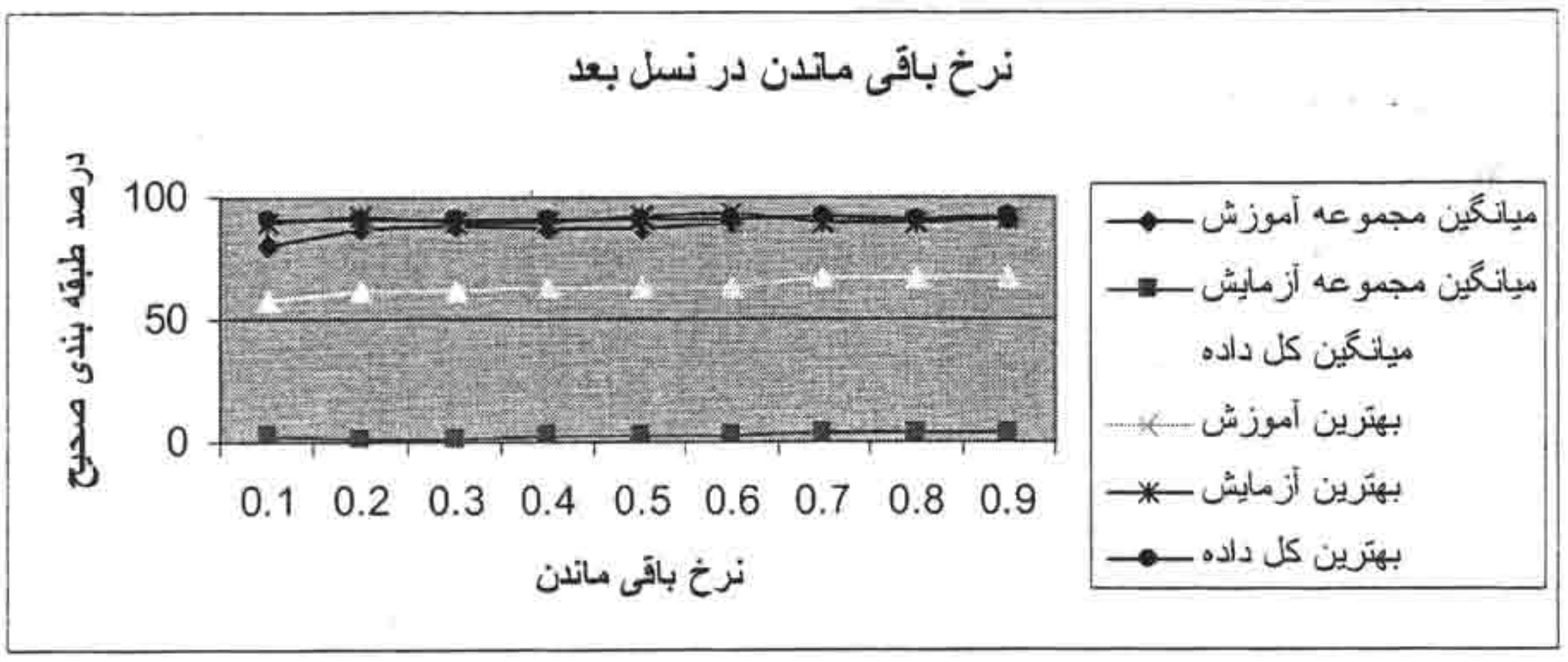
در جدول (۱) تعداد مشخصه ها در حالت اولیه ، پس از گسسته سازی و پس از انتخاب مشخصه ها نشان داده شده است.

جدول ۱ – تغییرات تعداد مشخصه ها پس از گسسته سازی و انتخاب مشخصه های مناسب

نام مجموعه داده	تعداد مشخصه اولیه	تعداد مشخصه پس از گسسته سازی	تعداد مشخصه پس از انتخاب	درصد کاهش تعداد مشخصه ها
Adeater	۱۵۵۸	۱۹۹۶	۴۴۵	٪۷۸
Insurance	۸۵	۲۸۲	۷۸	٪۷۲

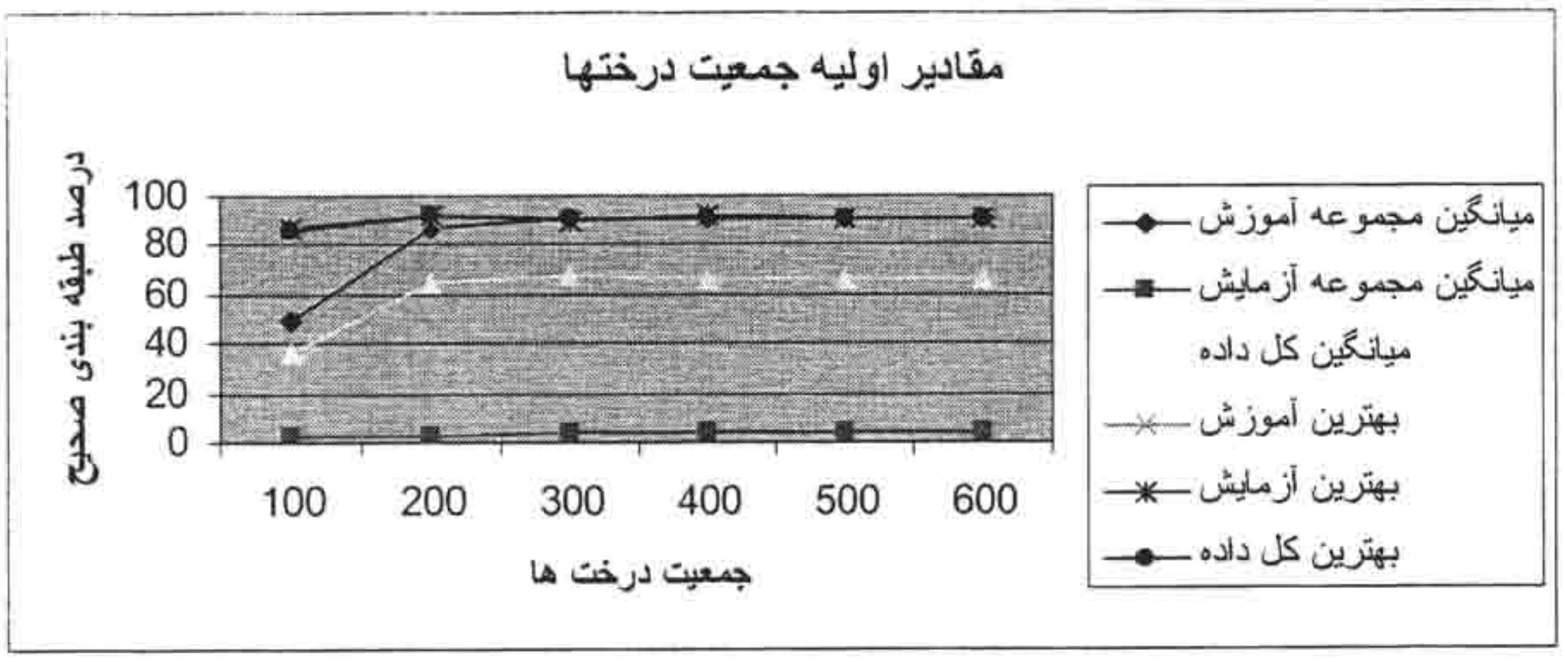
همانطور که مشاهده می شود تعداد مشخصه ها پس از انتخاب مشخصه های مناسب ، کاهش چشمگیری داشته است .

نمودارهـای مقایسه کارایی پردازش تکاملی بر روی پارامتر های نرخ باقی ماندن در نسل بعد ، مقادیر اولیه جمعیت درختها ، نقطـه قطـع و انـدازه مجموعـه آزمـایش بـر روی دو مجموعـه داده ، قبل و بعد از انتخاب مشخصه های مناسب نشان داده شده است.



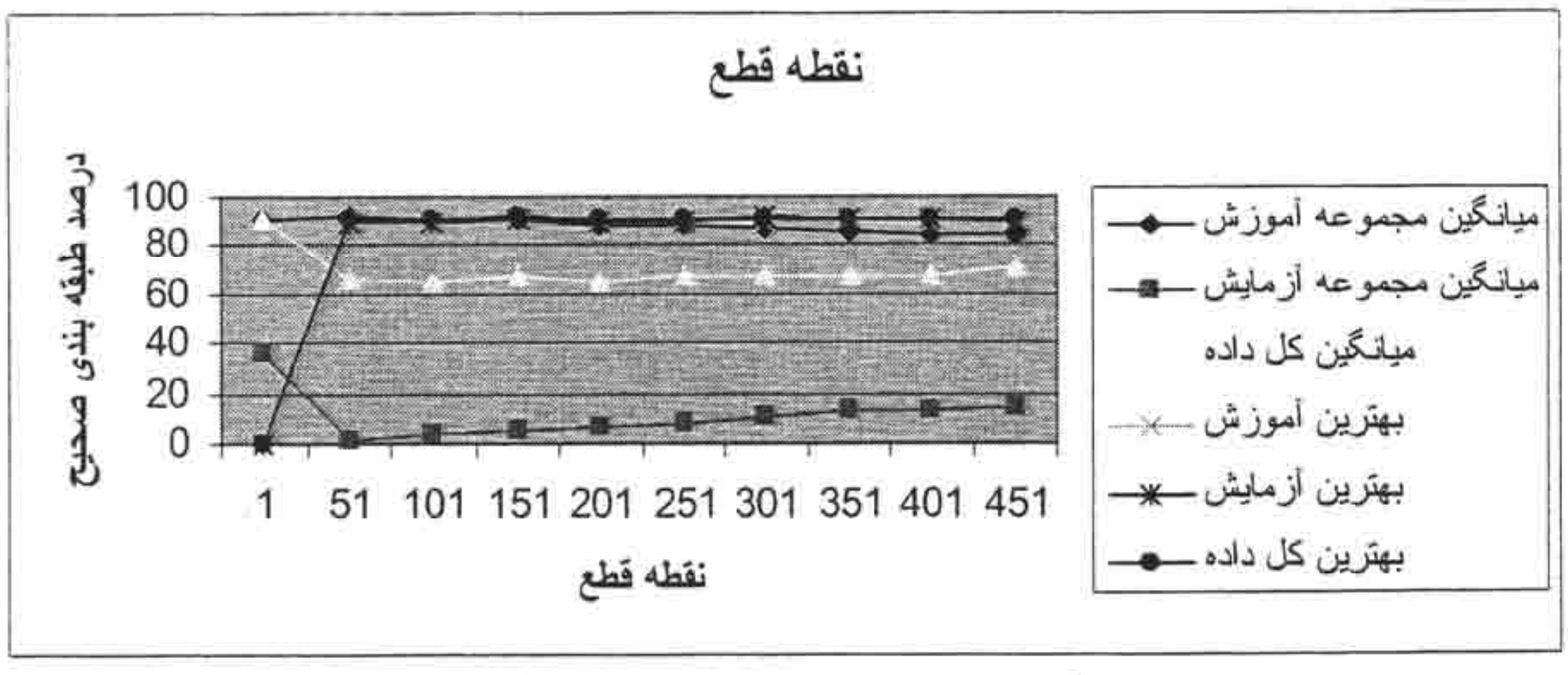
شکل ۲ – نرخ باقی ماندن در نسل بعد برای مجموعه داده Adeater

همـانطور که در نمودار فوق مشاهده می شود در نمودار نرخ باقی ماندن در نسل بعد ، کارایی با افزایش نرخ باقی ماندن در نسل بعد به آرامی افزایش پیدا میکند .



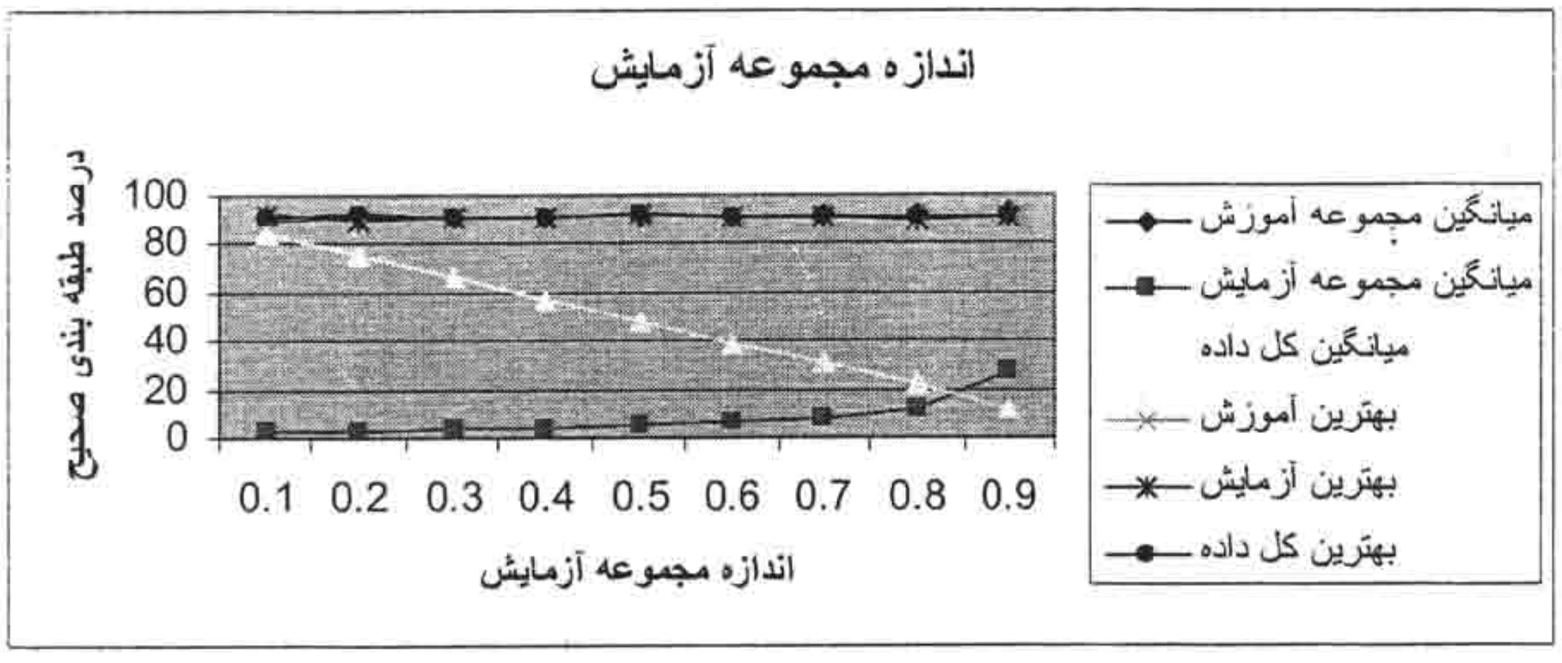
شکل ۳ – مقادیر اولیه جمعیت درخت ها برای مجموعه داده Adeater

در نمودار فوق نتیجه می شود با افزایش جمعیت اولیه درخت ها درصد طبقه بندی افزایش پیدا می کند .



شکل ۴ – نقطه قطع برای مجموعه داده Adeater

نتیجه نمودار فوق این است که با افزایش نقاط قطع در الگوریتم با وجود آنکه از میزان کارایی داده مجموعه آموزشی تا حدود کمی کاسته می شود اما به میزان کارایی مجموعه آزمایش و کل داده بصورت محسوس اضافه می گردد .



شکل ۵ – اندازه مجموعه آزمایش برای مجموعه داده Adeater

از نمودار فوق نتیجه می شود که با افزایش اندازه مجموعه آزمایش ، میزان طبقه بندی صحیح بر روی مجموعه آموزشی کاسته می گردد و با کاهش اندازه مجموعه آزمایش بر میزان طبقه بندی صحیح بر روی مجموعه آموزشی افزوده می گردد و این موضوع منطقی به نظر می رسد زیرا افزایش نمونه های آموزشی باعث تطبیق بیش از حد درخت تصمیم بر روی داده های آموزشی می گردد .

جدول ۲- مقایسه تغییرات طبقه بندی بر روی مجموعه داده های *Adeater* و *Insurance*

نام مجموعه داده	تعداد مشخصه اولیه	*درصد طبقه بندی حاصل از اجرای الگوریتم* C4.5 بر روی تمام مشخصه ها	تعداد مشخصه های انتخابی	درصد طبقه بندی حاصل از اجرای الگوریتم پیشنهادی بر روی مشخصه های انتخابی
Adeater	۱۹۹۶	٪۹۷	۴۴۵	٪۹۴
Insurance	۲۸۲	٪۹۶	۷۸	٪۹۴

همانطور که مشاهده می شود با کاهش دادن تعداد مشخصه ها و استفاده از روش پیشنهادی فقط مقدار کمی از درصد طبقه بندی کاسته شده است و این بیانگر این حقیقت است که الگوریتم پیشنهادی بر روی مجموعه داده های بسیار بزرگ بسیار بهینه تر از روش *C4.5* عمل خواهد نمود .

٦- نتیجه گیری

در این مقاله یک روش جدید برای تولید درخت تصمیم با استفاده از تئوری مجموعه های دانه درشت و پردازش تکاملی ارائه شد . مراحل مختلف شامل پیش پردازش داده ها ، انتخاب مشخصه های مناسب و ساخت درخت تصمیم به تفصیل بیان شد و نتایج حاصل از این کار با نتایج و الگوریتم های معتبر مورد مقایسه قرار گرفت و نشان داد که این روش در مقایسه با آنها بسیار بهینه عمل می کند . از مزایای این روش می توان به کار ، بر روی مجموعه داده های بزرگ با مجموعه مشخصه های فراوان اشاره نمود و از معایب این روش نیز می توان به کم بودن درصد طبقه بندی آن اشاره نمود . از کارهای آینده برای بهینه نمودن این الگوریتم می توان به جایگزین نمودن روش مورچگان به جای الگوریتم ژنتیک در انتخاب مشخصه ها و ساخت

درخت تصمیم اشاره نمود و از روشهای بهینه نمودن نتیجه طبقه بندی نظیر *Bagging* و *Boosting* برای رسیدن به نتیجه بهتر استفاده نمود .

۷- منابع

1) Nicholas kushmerick , *Learning to remove Internet advertisements* ,3rd Int. Conf. on Autonomous Agents 1999.

2) Lio,H. And Motoda H., eds., *Instance Selection and Construction for Data Mining* , Kluwer Academic Boston : MA , 2001

3) *Coil Challenge 2000* - http://www.dcs.napier.ac.uk/coil/challenge/.

4) Lian-Yin Zhai*, Li-Pheng Khoo, Sai-Cheong Fok , *Feature extraction using rough set theory and genetic algorithms an application for the simplification of product quality evaluation* , Computers & Industrial Engineering 43 (2002) 661–676

5) Komorowski, J., Pawlak, Z., Polkowski, L. & Skowron, A. (1999). *Rough sets: A tutorial. In Rough Set Theory*, 3–98.

6) D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley , Reading, Mass, 1989.

7) Jakub Wroblerwski, *Finding minimal reducts using genetic algorithm* , Instiute of Mathematics University of Warsaw,1996.

8) *روش نوین برای بهینه سازی شبکه عصبی با استفاده ازپردازش تکاملی*. عبدالرضا سالاری ، سعید سهیلی خواه ، پنجمین کنفرانس سیستم های هوشمند ، دانشگاه فردوسی مشهد ۱۳۸۲ (بخش۲) .

9) Grefenstette,J.J(1986). *Optimization of control parameters for genetic algorithms*. IEEE Transactions on Systems, Man, and Cybernetics, 16, 122-128.

10)M. Kantardzic , *Data Mining Concepts , Models , Method and Algorithms* , Wiley-InterScience , 2003 (Chapter 2)