



IFSB-ReliefF: یک روش انتخاب نمونه و ویژگی هم زمان بر مبنای ReliefF

زینب عباسی، محسن رحمانی* و حسین غفاریان

گروه کامپیوتر، دانشکده فنی و مهندسی، دانشگاه اراک، ایران

چکیده

افزایش استفاده از اینترنت و برخی از پدیده‌ها مانند شبکه‌های حسگر، منجر به افزایش غیرضروری اطلاعات شده است. اگرچه این امر مزایای بسیاری دارد، اما باعث ایجاد مشکلاتی مانند نیاز به فضای ذخیره‌سازی و پردازنده‌های بهتر و همچنین بالایش اطلاعات برای حذف اطلاعات غیرضروری می‌شود. الگوریتم‌های کاهش داده، روش‌هایی برای انتخاب اطلاعات مفید از مقدار زیادی داده‌های تکراری، ناقص و زائد فراهم می‌کنند. در این مقاله، الگوریتم ReliefF که یک الگوریتم رتبه‌بندی ویژگی است، تغییر داده شده تا به طور همزمان ویژگی‌ها و نمونه‌ها را انتخاب کند. الگوریتم پیشنهادشده می‌تواند بر روی ویژگی‌های اسمی و عددی و مجموعه‌داده‌ها با مقادیر مفقود اجرا و همچنین، می‌تواند به صورت موازی یک پردازنده چند هسته‌ای اجرا شود، که این امر باعث کاهش بسیار چشم‌گیر زمان اجرا و امکان اجرای آن روی مجموعه‌داده‌های بزرگ می‌شود؛ علاوه بر این، در این الگوریتم، انتخاب نمونه از هر رده متناسب با احتمال پیشین رده است و در نتیجه توازن و نسبت اولیه رده‌ها در مجموعه اصلی از بین نخواهد رفت. نتایج آزمایش بر روی چهار مجموعه‌داده نشان‌دهنده موفقیت الگوریتم پیشنهادی در این امر است.

وازگان کلیدی: کاهش داده‌ها، انتخاب نمونه، انتخاب ویژگی، ReliefF

IFSB-ReliefF: A New Instance and Feature Selection Algorithm Based on ReliefF

Zeinab Abbasi, Mohsen Rahmani* & Hossein Ghafarian

Computer Department, faculty of engineering, Arak University, Arak, Iran

Abstract

Increasing the use of Internet and some phenomena such as sensor networks has led to an unnecessary increasing the volume of information. Though it has many benefits, it causes problems such as storage space requirements and better processors, as well as data refinement to remove unnecessary data. Data reduction methods provide ways to select useful data from a large amount of duplicate, incomplete and redundant data. These methods are often applied in the pre-processing phase of machine learning algorithms. Three types of data reduction methods can be applied to data: 1. Feature reduction. 2. Instance reduction. 3. Discretizing feature values.

In this paper, a new algorithm, based on ReliefF, is introduced to decrease both instances and features. The proposed algorithm can run on nominal and numeric features and on data sets with missing values. In addition, in this algorithm, the selection of instances from each class is proportional to the prior probability of classes. The proposed algorithm can run parallel on a multi-core CPU, which decreases the runtime significantly and has the ability to run on big data sets.

One type of instance reduction is instance selection. There are many issues in designing instance selection algorithms such as representing the reduced set, how to make a subset of instances, choosing

* Corresponding author

نویسنده عهده‌دار مکاتبات



سال ۱۳۹۹ شماره ۴ پیاپی ۴۶

• تاریخ ارسال مقاله: ۰۵/۰۷/۱۳۹۷ • تاریخ پذیرش: ۱۱/۰۶/۱۳۹۸ • تاریخ انتشار: ۰۴/۱۲/۱۳۹۹ • نوع مطالعه: پژوهشی

distance function, evaluating designed reduction algorithm, the size of reduced data set and determining the critical and border instances. There are three ways of creating a subset of instances. 1) Incremental. 2) Decremental. 3) Batch. In this paper, we use the batch way for selecting instances. Another important issue is measuring the similarity of instances by a distance function. We use Jaccard index and Manhattan distance for measuring. Also, the decision on how many and what kind of instances should be removed and which must remain is another important issue. The goal of this paper is reducing the size of the stored set of instances while maintaining the quality of dataset. So, we remove very similar and non-border instances in terms of the specified reduction rate.

The other type of data reduction that is performed in our algorithm is feature selection. Feature selection methods divide into three categories: wrapper methods, filter methods, and hybrid methods. Many feature selection algorithms are introduced. According to many parameters, these algorithms are divided into different categories; For example, based on the search type for the optimal subset of the features, they can be categorized into three categories: Exponential Search, Sequential Search, and Random Search. Also, an assessment of a feature or a subset of features is done to measure its usefulness and relevance by the evaluation measures that are categorized into various metrics such as distance, accuracy, consistency, information, etc.

ReliefF is a feature selection algorithm used for calculating a weight for each feature and ranking features. But this paper is used ReliefF for ranking instances and features. This algorithm works as follows: First, the nearest neighbors of each instances are found. Then, based on the evaluation function, for each instance and feature, a weight is calculated, and eventually, the features and instances that are more weighed are retained and the rest are eliminated. IFSB-ReliefF (Instance and Feature Selection Based on ReliefF) algorithm is tested on two datasets and then C4.5 algorithm classifies the reduced data. Finally, the obtained results from the classification of reduced data sets are compared with the results of some instance and feature selection algorithms that are run separately.

Keywords: data reduction, instance selection, feature selection, ReliefF.

کرده و ویژگی‌های جدید تولید می‌کنند. برای مطالعه بیشتر،
به [3, 4, 5] مراجعه شود.

۲. کاهش نمونه: مانند کاهش ویژگی، دو نوع روش وجود دارند که عبارتند از: کاهش نمونه (یا رکورد)، که نمونه‌برداری (یا انتخاب نمونه) نامیده می‌شود و فشرده‌سازی اطلاعات. در نوع نخست نمونه‌ها با روش‌های تصادفی [6] یا روش‌های خاص [7] انتخاب شده و بقیه حذف می‌شوند. مانند روش‌های استخراج ویژگی، در فشرده‌سازی اطلاعات نمونه‌ها با هم ترکیب شده و به شکل محدودتر و فشرده تبدیل می‌شوند. برای مطالعه بیشتر به [8, 9, 10, 11] مراجعه شود.

۳. جداسازی مقادیر ویژگی‌ها: این روش‌ها دامنه مقادیر برای هر ویژگی را با گسترش کردن مقادیر ویژگی‌های عددی پیوسته کاهش می‌دهند [12].

همان‌طور که در قیل گفته شد، یک نوع از روش‌های کاهش داده، کاهش نمونه است. از آنجایی که مجموعه‌داده‌های حجیم، فراتر از قابلیت ذخیره و پردازش سامانه‌های معمولی هستند [13]، این نوع کاهش داده لازم است. یک نوع از کاهش نمونه، انتخاب نمونه‌ها است. دو دلیل برای افزایش دقت الگوریتم‌های یادگیری ماشین با انتخاب نمونه وجود دارد. یکی این که انتخاب نمونه می‌تواند نوافه در داده‌ها را از بین ببرد. و دیگر این‌که، برخی از نمونه‌ها زائد هستند و انتخاب نمونه به حذف آن‌ها کمک

۱- مقدمه

در سال‌های اخیر، بهدلیل استفاده از اینترنت و بهخصوص شبکه‌های اجتماعی، رشد داده‌ها بسیار سریع بوده است. علاوه‌بر این، بعضی از ابزارها مانند حسگرها مقدار زیادی داده تولید می‌کنند؛ بنابراین ذخیره‌سازی و پردازش نیازمند دستگاه‌های ذخیره‌سازی ویژه و رایانه‌های پرقدرت است. یکی از چالش‌های ذخیره و پردازش داده‌های بزرگ، فشرده‌سازی و استخراج اطلاعات مفید است. بسیاری از الگوریتم‌ها و روش‌ها برای حل این مشکل ارائه شده‌اند. روش‌های کاهش داده تلاش می‌کنند داده‌های زائد و بی‌ربط را حذف کنند تا دقت نتایج را افزایش داده و یا زمان و هزینه فرآیندها را کاهش دهند [1]. این روش‌ها اغلب در مرحله پیش‌پردازش الگوریتم‌های یادگیری ماشین استفاده می‌شوند.

سه نوع روش کاهش داده می‌تواند بر روی داده‌ها اعمال شود:

۱. کاهش ویژگی‌ها: این روش‌ها سعی می‌کنند ویژگی‌های مربوط را پیدا کرده و ویژگی‌های نوفه‌دار و غیر مفید را حذف کنند. آنها به دو دسته انتخاب ویژگی و استخراج ویژگی تقسیم می‌شوند. روش‌های انتخاب ویژگی، زیرمجموعه‌ای از ویژگی‌ها را انتخاب کرده و بقیه را حذف می‌کنند [2]. روش‌های استخراج ویژگی، ویژگی‌ها را ترکیب می‌کنند [3].

فصلنامه



⁷ Exponential⁸ Sequential⁹ IFSB-ReliefF: Instance and Feature Selection Based on ReliefF

مقاله بدين صورت سازماندهی شده است: بخش ۲ مقدمات لازم را توضیح می‌دهد و کارهای پیشین را بررسی می‌کند. در بخش ۳، الگوریتم IFSB-ReliefF که برای

می‌کند [20]. روش‌های فیلتر سریع و بی‌طرفانه هستند [2]. همچنین روش‌های فیلتر نسبت به روش‌های بسته‌بندی در برخورد با مجموعه‌داده‌های حجیم به دلیل انجام محاسبات سریع‌تر ترجیح داده می‌شود. روش‌های ترکیبی دو روش یادشده را ترکیب و از مزایای آنها استفاده می‌کنند [2]. الگوریتم‌های انتخاب تقسیم می‌شوند. به عنوان مثال، بر اساس دسته‌های مختلف تقسیم می‌شوند. به عنوان آنها نوع جستجو برای زیرمجموعه بھینه از ویژگی‌ها، می‌توان آنها را به سه دسته تقسیم کرد: جستجوی نمایی⁷ (جستجو برای هر زیرمجموعه ممکن برای یافتن زیرمجموعه مطلوب)، جستجوی ترتیبی⁸ (اضافه یا حذف ویژگی به مطلوب زیرمجموعه در هر مرحله) و جستجوی تصادفی (انتخاب تصادفی یک زیرمجموعه و ارزیابی آن) [4]. همچنین، ارزیابی یک ویژگی یا زیرمجموعه‌ای از ویژگی‌ها برای سنجش سودمندی و اهمیت آنها با معیارهای مختلف مانند فاصله، دقت، صحت، اطلاعات، و ... اندازه‌گیری می‌شود [2, 4]. نتایج الگوریتم‌های انتخاب ویژگی می‌توانند از یک الگوریتم داده‌کاوی برای مقایسه دقت مجموعه انتخاب شده و مجموعه داده‌های کامل مورد استفاده قرار گیرند [2].

ReliefF یک الگوریتم انتخاب ویژگی است [21]. این الگوریتم برای محاسبه وزن برای هر ویژگی و رتبه‌بندی ویژگی‌ها استفاده، اما در این مقاله، ReliefF برای رتبه‌بندی و انتخاب نمونه‌ها و ویژگی‌ها استفاده می‌شود. الگوریتم پیشنهادشده، ویژگی‌ها و نمونه‌ها را به طور هم‌زمان کاهش می‌دهد. همچنین این الگوریتم به گونه‌ای طراحی شده است که بتواند با استفاده از با برنامه‌نویسی موازی، هم‌زمان روی نمونه‌ها کار انجام دهد و با اجرای آن روی یک پردازنده چند هسته‌ای، زمان اجرا تا حد بسیار زیادی کاهش می‌یابد. الگوریتم IFSB-ReliefF (انتخاب نمونه و ویژگی بر اساس ReliefF⁹) بر روی چهار مجموعه داده آزمایش شده است و سپس الگوریتم C4.5 [22] داده‌های کاهش یافته را رتبه‌بندی می‌کند؛ در نهایت، نتایج بدست آمده از رتبه‌بندی مجموعه‌داده‌های کاهش داده شده با نتایج برخی از الگوریتم‌های انتخاب نمونه و ویژگی که به طور جداگانه اجرا می‌شوند، مقایسه می‌شود.

مقاله بدين صورت سازماندهی شده است: بخش ۲ مقدمات لازم را توضیح می‌دهد و کارهای پیشین را بررسی می‌کند. در بخش ۳، الگوریتم IFSB-ReliefF که برای

می‌کند. بسیاری از الگوریتم‌های انتخاب نمونه به دلیل کارایی کم، کمبود منابع و تعیین ناپذیری نمی‌توانند از عهده مجموعه‌داده‌های بزرگ برآیند [12].

مسائل بسیاری در طراحی الگوریتم‌های انتخاب نمونه وجود دارند، مانند چگونگی نمایش مجموعه کاهش یافته، نحوه ایجاد یک زیرمجموعه از نمونه‌ها، انتخاب تابع فاصله مناسب، ارزیابی عملکرد الگوریتم انتخاب نمونه طراحی شده [14]، شرایط حذف یا نگهداری نمونه‌ها (اندازه مجموعه‌داده‌های کاهش داده شده) و تعیین نمونه‌های بحرانی و نقاط مرزی [15].

سه راه برای ایجاد زیرمجموعه‌ای از نمونه‌ها وجود دارد: ۱) افرایشی¹: در ابتدا، زیرمجموعه خالی است و سپس بعضی از نمونه‌ها به آن افزوده می‌شوند. ۲) کاهشی²: ابتدا تمام مجموعه‌داده به عنوان زیرمجموعه نهایی انتخاب شده و سپس برخی از نمونه‌ها از آن حذف می‌شوند. ۳) دسته‌ای³: ابتدا تمام نمونه‌ها مورد تجزیه و تحلیل قرار می‌گیرند و برای هر یک از آنها یک وزن یا رتبه محاسبه و سپس تمام نمونه‌ها با وزن کم به یک باره حذف می‌شود [16]. در این مقاله، از روش دسته‌ای برای انتخاب نمونه‌ها استفاده می‌شود. یکی دیگر از مسائل مهم، اندازه‌گیری شباهت نمونه‌ها با یک تابع فاصله است. در روش پیشنهادی از شاخص ژاکارد⁴ [17] و فاصله منهتن⁵ برای اندازه‌گیری شباهت نمونه‌ها استفاده می‌شود. همچنین، تصمیم‌گیری درباره تعداد نمونه‌هایی که باید حذف شوند یا باقی بمانند، یکی دیگر از مسائل مهم است. هدف این مقاله کاهش حجم نمونه‌های ذخیره شده در عین حفظ کیفیت مجموعه‌داده‌ها است؛ بنابراین، موارد بسیار مشابه و نقاط غیرمرزی بر حسب نرخ کاهش مشخص شده حذف می‌شود.

نوع دیگری از کاهش داده‌ها که در الگوریتم پیشنهادی انجام می‌شود، انتخاب ویژگی است. روش‌های انتخاب ویژگی به سه دسته تقسیم می‌شود: روش‌های بسته‌بندی⁶، روش‌های فیلتر و روش‌های ترکیبی. روش‌های بسته‌بندی از یک الگوریتم داده‌کاوی برای ارزیابی ویژگی‌ها استفاده می‌کنند [18]. این روش‌ها مفید است؛ اما در هنگام استفاده برای داده‌های حجیم کارایی چندانی ندارد [19]. روش‌های فیلتر از الگوریتم‌های داده‌کاوی استفاده نمی‌کنند و از طریق یک تابع ارزیابی، ویژگی‌های مربوطه را جستجو

¹ Incremental² Decremental³ Batch⁴ Jaccard index⁵ Manhattan distance⁶ Wrapper

چندردهای و مجموعه‌دادهها با مقادیر مفقود کار کند [21]. به منظور غلبه بر این مشکلات، الگوریتم Relief اصلاح شده است و الگوریتم جدیدی به نام ReliefF ارائه شد [21]. ReliefF می‌تواند بر روی مجموعه‌دادهها با مقادیر مفقود و مجموعه‌دادهها با بیش از دو رده داده کار کند. به جای انتخاب یکی از نزدیکترین همسایه‌ها، که در الگوریتم Relief می‌شود، ReliefF تعدادی از نزدیکترین ReliefF همسایگان یک نمونه انتخابی را می‌یابد؛ علاوه بر این، از یکتابع متفاوت برای محاسبه وزن ویژگی‌ها استفاده می‌کند. این کار برای مدیریت مجموعه‌داده‌های ناقص انجام می‌شود. همان‌طور که در شکل (۱) نشان داده شده است، الگوریتم ReliefF به شرح زیر عمل می‌کند:

در خط ۵، pc احتمال پیشین رده است. در خط ۶.a الگوریتم یک نمونه تصادفی را انتخاب می‌کند. خطوط ۶.b تا ۶.d، تعداد B تا از نزدیکترین همسایگان همراه با نمونه انتخاب شده به نام Hits و B تا از نزدیکترین همسایگان رده‌های دیگر را که نامیده می‌شود، می‌یابند. در خط ۶.e یک تابع ارزیابی (۸) وزن ویژگی‌ها را محاسبه می‌کند. این تابع تفاوت بین نمونه انتخاب شده و مجموعه‌های Hits و Misses را محاسبه می‌کند. ReliefF اصلی از فاصله منهتن در تابع (۸) استفاده می‌کند. معادله $(pc)(1-p(class(x_t))$ در فرمول وزن ویژگی، برای استفاده از ReliefF در مسائل چند رده استفاده می‌شود [25]. درنهایت، الگوریتم فهرستی از وزن ویژگی‌ها را به دست می‌آورد. بر اساس این فهرست، ویژگی‌های با وزن بیشتری انتخاب شده و مابقی حذف می‌شوند [24].

1. /* X =the set of training examples */

2. /* L =the number of random examples to draw */

3. /* B =the number of nearest neighbours to compute*/

4. For each feature $k \{w_k=0.0\}$

{pc:=

5. For each class $c \{pc:=the fraction of X belonging to class c\}$

6. For $l:=1$ to L do

a) Randomly select an instance (x_i, y_i)

$y_i=y_t$ که مجموعه hits

b) Let hits be the set of B examples (x_i, y_i) nearest to x_t such that $y_i=y_t$

c) For each class $c \neq y_t$

انتخاب ویژگی و نمونه از Relief استفاده می‌کند، ارائه می‌شود. بخش ۴ نتایج آزمایش‌ها و سرانجام در بخش ۵ خلاصه و نتیجه کار ارائه می‌شود.

۲- کارهای مرتبه

در این بخش، ابتدا، الگوریتم Relief توضیح داده شده است. برخی از الگوریتم‌های معرفی شده هم نمونه و هم ویژگی‌ها را انتخاب می‌کنند. برخی از مقالات در مورد این نوع الگوریتم‌ها در این بخش مورد بررسی قرار گرفته است.

ReliefF - ۲-۱

یک راه برای کاهش حجم داده‌ها انتخاب ویژگی [2] است. الگوریتم‌های انتخاب ویژگی می‌توانند به حذف ویژگی‌های زائد و یا دارای نوفه کمک کنند. همچنین می‌توانند زمان اجرا را کاهش داده و وقت الگوریتم‌های داده‌کاوی را افزایش دهند.

[23] یک روش انتخاب ویژگی فیلتری است که از یادگیری مبتنی بر نمونه الهام گرفته شده است. این الگوریتم یک روش پیش پردازش معروف است که می‌تواند در بسیاری از مسائل داده‌کاوی مورد استفاده قرار گیرد [24]. ویژگی‌ها را بر اساس کیفیت آن‌ها به طور مؤثر رتبه‌بندی می‌کند. این الگوریتم می‌تواند بر روی مجموعه‌داده‌های اسمی و عددی کار کند [21]. Relief درجه اهمیت ویژگی‌ها را با محاسبه تفاوت بین ویژگی‌ها تخمین می‌زند [24]؛ اما Relief بخلاف همه محاسن، معایبی نیز دارد. این الگوریتم نمی‌تواند در مجموعه‌داده‌های

Algorithm ReliefF (input:X, L, B)

* / X =مجموعه داده آموزشی

2. * / تعداد نمونه‌هایی که باید به طور تصادفی انتخاب شوند=L *

3. * / تعداد نزدیکترین همسایه‌هایی که باید محاسبه شوند=B *

4. برای هر ویژگی $k \{w_k=0.0\}$:

5. برای کلاس c کسری از مجموعه‌داده که متعلق به این رده است

6. حلقه تکرار از ۱ تا L

a. یک نمونه تصادفی انتخاب کن (x_i, y_i)

b. برای هر رده $c \neq y_i$

c. برای هر داده x_t که مجموعه hits

سال ۱۳۹۹ شماره ۴ پیاپی ۴۶





مجموعه misses که مجموعه‌ای از نزدیکترین نمونه‌ها (x_i, y_i) به نمونه x_t است را بباب به گونه‌ای که $y_j = c$

Let misses be the set of B examples (x_j, y_j) Nearest to x_t such that $y_j = c$

d. پایان حلقه

d) End for

e) For each feature k

$$w_k = w_k - \frac{1}{LB} \sum_{(x_i, y_i) \in Hit} \delta(x_{t,k}, x_{i,k}) + \sum_{c \neq y_t} \frac{pc}{(1 - p(class(x_t)))LB} \sum_{(x_i, y_i)} \delta(x_{t,k}, x_{i,k})$$

e. برای هر ویژگی k

f. پایان حلقه

f) End for

7. End for

7. پایان حلقه

8. مجموعه مقادیر وزن‌ها (W) را به عنوان خروجی برگردان.

8. Return the set of weights (W)

9. پایان Relief

(شکل-۱): شبیه کد الگوریتم ReliefF

(Figure-1): Pseudo code of ReliefF

اطلاعات برای انتخاب نمونه‌ها و ویژگی‌ها و استفاده از آن‌ها در یک سامانه فیلترینگ تطبیقی^۱ معرفی شده است. دلیل استفاده از انتخاب نمونه در این مقاله این است که تعداد مشتریان (نمونه‌ها) در سامانه پالایش گروهی افزایش می‌یابد؛ بنابراین، کاهش تعداد مشتریان، دقت و مقیاس‌پذیری سامانه را افزایش می‌دهد. همچنین با توجه به نوع داده‌های خاصی مسئله، از آنجایی که شناسایی ویژگی‌های مرتبط از ویژگی‌های نامناسب دشوار است، آن‌ها به جای انتخاب ویژگی، از وزن‌گذاری ویژگی‌ها استفاده می‌کنند؛ بنابراین، اطلاعات متقابل برای وزن‌دهی ویژگی‌ها استفاده می‌شود. برای انتخاب نمونه، ضریب همبستگی هر نمونه به کمک سایر نمونه‌ها محاسبه می‌شود. این مقاله به کاهش داده‌ها در یک حوزه خاص می‌پردازد و روش استفاده شده قابل تعمیم نیست. نتایج مقاله نیز فقط بر روی مجموعه‌داده EachMovie آزمایش شده است. همچنین، اگرچه این روش دقت را افزایش می‌دهد، زمان لازم برای اجرای الگوریتم بسیار بیشتر از سایر روش‌ها است.

در [27] یک مدل کاهش جدید برای بهبود کارایی سامانه‌های تشخیص نفوذ پیشنهاد شده که در آن از OneR برای کاهش ویژگی و از انتشار وابستگی^۲ برای انتخاب نمونه استفاده شده است. آنها از چارچوب برنامه‌نویسی موازی MapReduce [28] برای پیاده‌سازی الگوریتم خود برای افزایش سرعت کاهش داده‌ها استفاده کردند. برای آزمایش نتایج کاهش داده‌ها، داده‌های کاهش‌یافته بهوسیله الگوریتم‌های KNN و SVM رده‌بندی شدند. این مقاله بر

۲-۲- انتخاب هم‌زمان ویژگی‌ها و نمونه‌ها

بعضی از الگوریتم‌ها به جای انتخاب، فقط ویژگی‌ها یا نمونه‌ها می‌توانند هم ویژگی‌ها و هم نمونه‌ها را انتخاب کنند. این انتخاب می‌تواند در یک یا چند مرحله انجام شود. اگر ویژگی‌ها و نمونه‌ها به طور هم‌زمان انتخاب شوند، انتخاب در یک مرحله است؛ در غیر این صورت، در چند مرحله خواهد بود. به عنوان مثال، در [1] یک الگوریتم تکاملی برای انتخاب ویژگی و نمونه مورد استفاده قرار می‌گیرد. در این الگوریتم، هر کروموزوم دارای $n + m$ ژن است (اگر مجموعه‌داده دارای n نمونه و m ویژگی باشد). اگر نمونه / ویژگی در مجموعه‌داده‌ایی که کروموزوم آن را نشان می‌دهد باقی بماند، مقدار ژن برابر با یک است، و اگر از مجموعه‌داده حذف شود، مقدار ژن برابر با صفر خواهد بود. برای هر ترکیب ممکن از نمونه‌ها و ویژگی‌ها، یک کروموزوم وجود دارد. یک رده‌بند برای ارزیابی کروموزوم‌ها استفاده می‌شود. اگر چه الگوریتم بهترین نمونه‌ها و ویژگی‌ها را در یک مرحله و با دقت زیادی انتخاب می‌کند، به دلیل استفاده از یک رده‌بند برای ارزیابی هر کروموزوم، برای مجموعه‌داده‌های بزرگ، این روش بسیار زمان بر یا غیر قابل استفاده خواهد بود. این مقاله بر روی هجده مجموعه‌داده اجرا شده که این ادعای شده شش عدد از آنها در مقیاس بزرگ هستند؛ اما بزرگترین آن‌ها (Satimage) شامل ۶۴۳۵ نمونه و ۳۶ ویژگی است که از مجموعه‌داده‌ای استفاده شده در این مقاله بسیار کوچک‌تر است.

برخی از الگوریتم‌ها ویژگی‌ها را در چند مرحله انتخاب می‌کنند. در [26] یک راه‌کار تئوری مبتنی بر



روی دو مجموعه داده KDD99 و CDMC2012 آزمایش و اجرا شد. نویسنده‌گان ادعا کردند که سرعت تشخیص حمله ۱۸۴ بار افزایش یافته است، در حالی‌که دقت تشخیص تفاوت معنی‌داری ندارد.

یک روش برای ایجاد یک سامانه تشخیص خودکار مبتنی بر تصویر در [29] معرفی شد. ایجاد یک سامانه تفسیر خودکار برای تشخیص بیماری‌ها نیازمند یک کاتالوگ تصویری است که به عنوان پایه‌ای برای توسعه سامانه استفاده می‌شود؛ بنابراین، نویسنده‌گان این مقاله با استفاده از روش‌های انتخاب ویژگی و انتخاب نمونه، نمونه‌ای از تصاویر را ایجاد کردند. آنها روش خود را بر روی چهار کاتالوگ تصویر اجرا کردند. برای کاهش نمونه‌ها، الگوریتم پیشنهادی هر نمونه را با همسایه‌های هم‌کلاسیش ترکیب و یک نمونه جدید ایجاد می‌کند. عمل ادغام نمونه‌ها تا زمانی ادامه می‌یابد که نمونه‌های جدید دقت رده‌بندی را کاهش ندهند. برای کاهش ویژگی‌ها، الگوریتم از یک روش بسته‌بندی برای تعیین وزن و انتخاب ویژگی‌ها استفاده می‌کند. در این بخش، هر ویژگی با رده‌بند KNN ارزیابی و ویژگی‌های خوب تعیین می‌شود. در واقع، در هر دو روش انتخاب نمونه و ویژگی، الگوریتم‌های رده‌بندی به عنوانتابع ارزیابی استفاده می‌شود. شاید این روش دقت سامانه را افزایش دهد، اما زمان زیادی طول می‌کشد و برای مجموعه‌داده‌های حجمی قابل استفاده نیست.

در [30] یک الگوریتم ژنتیک برای انتخاب ویژگی‌ها و نمونه‌ها مورد استفاده قرار می‌گیرد. در این مقاله، انتخاب ویژگی و نمونه به ترتیب، نه همزمان انجام می‌شود. نویسنده‌گان از روش‌های کاهش داده موجود بر پایه الگوریتم ژنتیک برای کاهش ویژگی‌ها و نمونه‌ها استفاده کردند. بسته به اینکه آیا ویژگی‌ها یا نمونه‌ها و با چه ترتیبی انتخاب شوند، پنج روش مختلف به مجموعه‌داده‌ها اعمال شده است. این روش‌ها دو استراتژی را بر اساس نتایج به دست آمده از رده‌بندی داده‌های کاهش‌یافته با دو رده‌بند KNN و SVM پیشنهاد کردند. نتیجه کلی این است که انجام انتخاب ویژگی و سپس انتخاب نمونه‌ها، راه حل بهینه برای پیش‌پردازش داده‌ها است؛ اما برای پایگاه داده‌های بزرگ، این همیشه بهترین حالت نیست، و گاهی اوقات انجام انتخاب نمونه و سپس انتخاب ویژگی بهتر است. البته این نتیجه‌گیری بر اساس نتایج به دست آمده بر روی هشت مجموعه داده، به دست آمده است؛ اما نمی‌توان آن را قطعی دانست و به عنوان یک قانون کلی در مورد تمام مجموعه داده‌ها اعمال کرد.

یک روش انتخاب ویژگی و نمونه به طور همزمان برای ارائه یک صفحه داده فشرده و دقیق برای طبقه‌بندی متن در [31] استفاده می‌شود. الگوریتم پیشنهادشده در دو مرحله عمل می‌کند: در مرحله نخست، ویژگی‌هایی که به طور دقیق رده هدف را پیش‌بینی می‌کنند، به ترتیب انتخاب، سپس تمام استنادی که دست‌کم یکی از این ویژگی‌ها را ندارند از مجموعه آموزش حذف می‌شوند. در مرحله دوم، تعدادی از ویژگی‌ها از میان ویژگی‌های انتخاب شده در مرحله قبلی، انتخاب می‌شوند. آنها تمایل دارند مکمل رده هدف را پیش‌بینی کنند. مجموع استناد و ویژگی‌های انتخاب شده در این دو مرحله، مجموعه آموزشی را ایجاد می‌کند. نویسنده‌گان از مجموعه‌ای از داده‌های متی که شامل بیست‌هزار پست خبری بود، برای پیاده‌سازی و مقایسه روش پیشنهادشده با روش‌های دیگر استفاده می‌کنند.

در [32] پژوهش‌گران از کاهش همزمان داده‌ها برای طراحی یک مدل پیش‌بینی ورشکستگی در موضوعات اقتصادی استفاده کردند. آنها از استدلال مبتنی بر مورد (CBR) برای پیش‌بینی ورشکستگی استفاده کردند. با وجود تمام محاسن، کارایی آن بسیار کم است. آنها از انتخاب ویژگی‌ها و موارد بر اساس الگوریتم ژنتیک برای بهبود عملکرد CBR استفاده کردند. این روش فقط برای یک برنامه خاص طراحی شده است. همچنین، به دلیل استفاده از الگوریتم ژنتیک، این روش بسیار وقت‌گیر است و نیاز به منابع سخت‌افزاری خاص دارد. همچنین اندازه جمعیت و تعداد نسل‌هایی که برای الگوریتم ژنتیک تعریف می‌شود، باید کوچک باشد؛ در غیر این صورت فضای جستجو بسیار بزرگ خواهد بود. مجموعه داده آن‌ها شامل اطلاعات وضعیت اقتصادی مربوط به ۲۶۷۰ شرکت است.

به طور خلاصه، کاهش همزمان ویژگی‌ها و نمونه‌ها می‌تواند یک مسئله مهم باشد و به افزایش کارایی و صحت رده‌بندها کمک کند. همان‌طور که در این بخش مشاهده شد، پژوهش‌های زیادی در این زمینه صورت گرفته است، اما متأسفانه راه حل‌های پیشنهادی دچار برخی مشکلات هستند. این مشکلات شامل زمان برپویدن، قابل استفاده نبودن در مجموعه داده‌های بزرگ، مناسب‌بودن برای برخی از مشکلات خاص و تعیین ناپذیری است؛ بنابراین، این مقاله تلاش می‌کند تا با برطرف کردن این مشکلات، با ارائه یک الگوریتم مناسب برای داده‌های بزرگ و از انواع مختلف تلاش کند. همچنین از ساختارهای برنامه‌نویسی موازی برای



ویژگی‌ها وزنی اختصاص می‌دهد. در این الگوریتم، یک نمونه تصادفی در هر تکرار انتخاب و تفاوت آن با مجموعه Hit و Miss اندازه‌گیری می‌شود. یک ویژگی خوب، ویژگی‌ای است که بیشتر شبیه به نزدیک‌ترین نمونه از همان رده است و بیشترین تفاوت را با نزدیک‌ترین نمونه از رده‌های مختلف دارد. الگوریتم IFSB-ReliefF از همین روش برای انتخاب نمونه‌ها و ویژگی‌ها استفاده می‌کند. شکل (۲) شبکه IFSB-ReliefF را نشان می‌دهد.

کاهش زمان اجرای الگوریتم پیشنهادی استفاده و در بخش بعدی، الگوریتم پیشنهادی توضیح داده شده است.

۳- الگوریتم پیشنهادی (IFSB-ReliefF)

این بخش نحوه استفاده از الگوریتم ReliefF برای انتخاب هم‌زمان ویژگی‌ها و نمونه‌ها را توضیح می‌دهد. همان‌طور که در قبل گفته شد، الگوریتم ReliefF یک الگوریتم رتبه‌بندی ویژگی است که بر اساس تابع ارزیابی برای هر یک از

IFSB-ReliefF Algorithm (input:X, B)
/* X= مجموعه داده آموزشی */

1. /*X=the set of training examples */
2. /*B=the number of nearest neighbors to compute*/
3. /*L= Number of instances.*/
4. For t:=1 to L do (This loop can run in parallel)
 - a. تعداد B تا از نزدیک‌ترین نمونه‌ها به x_t از رده y_t (مجموعه Hit) را با شاخص ڈاکارد یا فاصله منهن بیاب.
 - b. برای هر رده $c \neq y_t$ Find B nearest instances to x_t from class y_t (Hit set) by Jaccard index or Manhattan distance
 - c. تعداد B تا از نزدیک‌ترین نمونه‌ها به x_t از رده c را با شاخص ڈاکارد یا فاصله منهن بیاب و به مجموعه Miss اضافه کن.
 - d. وزن هر نمونه x_t را محاسبه کن و آن را در آرایه W قرار بده.
5. End for (line 4.b)
6. Calculate the weight of each instance x_t and put it in array W:

$$w_t = \sum_{c \neq y_t} \frac{1}{LB} \sum_{(x_j, y_j) \in \text{Miss}} \delta(x_t, x_j) - \frac{1}{LB} \sum_{(x_j, y_j) \in \text{Hit}} \delta(x_t, x_j)$$
7. برای هر ویژگی k از نمونه x_t وزن را محاسبه کن و آن را در آرایه WFI قرار بده.
8. For each feature k of x_t calculate the weight and put it in array WF:

$$wf_{tk} = \sum_{c \neq y_t} \frac{pc}{(1-p(class(x_t)))LB} \sum_{(x_j, y_j) \in \text{Miss}} \delta(x_{t,k}, x_{j,k}) - \frac{1}{LB} \sum_{(x_i, y_i) \in \text{Hit}} \delta(x_{t,k}, x_{i,k})$$
9. برای هر ویژگی k میانگین وزن تمام مقادیر موجود در WFI را محاسبه کن و یک وزن برای هر ویژگی بدست آور و در WF قرار بده.
10. Calculate the average of all wf_{tk} and get a weight for each feature Wf.
11. Sort the weight array W and Wf.
12. Based on the sorted array W, select instances that have more weight from each class according to the proportion to be selected and put in the output array.
13. بر اساس آرایه WF، ویژگی‌های با وزن بیشتر را از آرایه Out انتخاب کرده و در آرایه خروجی قرار بده.
14. In the output array, based on WF array, keep the features with more weight and remove the rest.
15. پایان الگوریتم پیشنهادی

(شکل-۲): الگوریتم IFSB-ReliefF

(Figure-2): Pseudo code of IFSB-ReliefF

در خط ۴، یک حلقه تکرار به تعداد همه نمونه‌ها وجود دارد، اما در الگوریتم ReliefF، تنها چند نمونه تصادفی

همان‌طور که در شکل نشان داده شده، IFSB-ReliefF به شرح زیر عمل می‌کند:



به صورت معادله (۲) تعریف می‌شود:

انتخاب می‌شود. اگر چه این کار وقت‌گیر خواهد بود، چون دستورهای داخل حلقه به طور مستقل برای یک نمونه اجرا می‌شوند، می‌توانند به طور موازی در سیستم‌های چندپردازنه اجرا شوند. چنانچه در بخش نتایج، ملاحظه خواهد شد، این امر به کاهش زمان اجرا و امکان اجرای الگوریتم روی مجموعه‌داده‌های بزرگ، کمک زیادی خواهد کرد. مانند ReliefF در خطوط ۴.۴ و ۴.۵ تعداد B تا برای مجموعه Hits و B تا برای مجموعه Misses از هر رده دیگر انتخاب شده است. IFSB-ReliefF می‌تواند از هر فرمول فاصله برای تابع ارزیابی (۸) برای اندازه‌گیری فاصله بین نمونه‌ها استفاده کند. همان‌گونه که پیش‌تر گفته شد ReliefF اولیه [21] از تابع فاصله منهتن برای اندازه‌گیری فاصله بین دو نمونه استفاده می‌کند، اما از آنجا که الگوریتم ReliefF اندازه فاصله بین دو نمونه را برای به دست آوردن ارزش و وزن ویژگی‌ها استفاده می‌کند و بنابراین، به این که به طور کلی دو نمونه چقدر با هم تشابه دارند، کاری ندارد؛ اما در این مقاله، از تابع فاصله منهتن و شاخص ژاکارد به عنوان تابع ارزیابی استفاده می‌شود. گفتنی است که برای نخستین بار در مقاله [33]، توسط نویسندهای همین مقاله از شاخص ژاکارد به عنوان تابع ارزیابی در ReliefF استفاده می‌شود. شاخص ژاکارد، که همچنین به عنوان ضریب تشابه ژاکارد شناخته می‌شود، برای مقایسه شباهت دو مجموعه استفاده می‌شود؛ بنابراین می‌توان مقایسه کرد که آیا دو نمونه به طور کلی با هم تشابه‌ی دارند یا خیر. اگر حقیقی بزرگ‌تر از صفر باشند، شاخص ژاکارد از آنها به صورت معادله (۱) تعریف می‌شود:

$$J(\mathbf{X}, \mathbf{Y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)} \quad (1)$$

از آنجا که این فرمول برای اعداد حقیقی بیشتر از صفر تعریف می‌شود. برای استفاده از این فرمول در تابع، ویژگی‌های عددی برخی از مجموعه‌های داده ابتدا باید به فاصله‌ای از صفر تا یک نرمال‌سازی شوند؛ بنابراین نتایج معادله (۱) نیز در محدوده صفر تا یک قرار می‌گیرند. اگر نتیجه یک باشد، دو نمونه به طور دقیق همانند هستند و اگر نتیجه صفر باشد، به این معنی است که آنها به طور کامل متفاوت هستند و بین آنها هیچ شباهتی وجود ندارد.

برای دو مجموعه غیر عددی (رشته‌ها) $x=(x_1, x_2, \dots, x_n)$ و $y=(y_1, y_2, \dots, y_n)$ ، شاخص ژاکارد

$$J(\mathbf{X}, \mathbf{Y}) = \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|} \quad (2)$$

برخی از ویژگی‌های مجموعه‌داده‌ها مقدار مفقود دارند؛ در این حالت، اگر مقدار x_i یا y_i یا هر دو ناشناخته باشد، آنگاه $\emptyset \cap y_i = \emptyset$ از IFSB-ReliefF معادله (۲) به عنوان یکتابع ارزیابی استفاده می‌کند. اگر ویژگی‌ها عددی باشند، معادله (۱) و اگر غیر عددی یا دارای مقدار مفقود باشند، معادله (۲) استفاده می‌شود.

برای مثال اگر سه نمونه با ویژگی‌های زیر در مجموعه‌داده موجود باشد:

$$\text{نمونه ۱ (I1)} : 0.9, 0, ?, c$$

$$\text{نمونه ۲ (I2)} : 0.15, 0.5, a, b$$

$$\text{نمونه ۳ (I3)} : 0.1, 0.5, a, ?, ?$$

برای محاسبه شاخص ژاکارد برای نمونه ۱ و ۲ و نمونه ۲ و ۳ به این شکل عمل می‌شود که ابتدا ویژگی‌های عددی و غیر عددی جدا و برای هر کدام با فرمول مخصوص محاسبه صورت گرفته و در اخر بر حسب تعداد ویژگی‌های عددی و غیر عددی، میانگین گرفته می‌شود.

$$J(I1, I2)_{num} = \frac{\min(0.9, 0.15) + \min(0, 0.5)}{\max(0.9, 0.15) + \max(0, 0.5)} = \frac{0.15 + 0}{0.9 + 0.5} = 0.107$$

$$J(I1, I2)_{str} = \frac{|\emptyset|}{|(a, b, c, ?)|} = 0$$

$$J(I1, I2) = \frac{0.107 + 0}{2} = 0.053$$

$$J(I2, I3)_{num} = \frac{\min(0.15, 0.1) + \min(0.5, 0.5)}{\max(0.15, 0.1) + \max(0.5, 0.5)} = \frac{0.1 + 0.5}{0.15 + 0.5} = 0.92$$

$$J(I2, I3)_{str} = \frac{|(a)|}{|(a, b, ?)|} = 0.333$$

$$J(I2, I3) = \frac{0.92 + 0.333}{2} = 0.628$$

با توجه به مثال، مشاهده می‌شود که از نظر شاخص ژاکارد دو نمونه ۲ و ۳ تشابه بیشتری به هم نسبت به نمونه‌های ۱ و ۲ دارند.

برای هر نمونه، الگوریتم نمونه‌هایی را برای مجموعه \mathbf{Z} اکارد هستند. درنهایت، در خط ۴.۴ وزن هر نمونه با استفاده از مقدار مجموعه Hit و Miss محاسبه می‌شود. همان‌طور که مشاهده می‌شود، هر قدر یک نمونه مشابه نمونه‌هایی از سایر رده‌ها است، وزن آن افزایش می‌یابد. در مقابل، اگر نمونه شبیه به نمونه‌های رده خودش باشد، وزن کاهش می‌یابد. دلیل این افزایش و کاهش این است که در الگوریتم پیشنهادی، نمونه‌های بحرانی در نقاط مرزی بین دو

اجرا شود؛ سپس نتایج رده بندی داده های کاهش یافته با IFSB-ReliefF با نتایج به دست آمده از سایر الگوریتم های کاهش مقایسه می شود. برای آزمایش نتایج الگوریتم، چندین مجموعه داده استفاده شده که خصوصیات و نتایج آنها در بخش بعدی توضیح داده خواهد شد.

۴- نتایج تجربی

در این بخش، نتایج رده بندی داده های انتخاب شده به وسیله IFSB-ReliefF با رده بندی داده های اولیه و برخی از الگوریتم های دیگر کاهش داده مقایسه می شود. ابتدا برخی از مسائل مانند نیازمندی های سخت افزاری و نرم افزاری، ویژگی های مجموعه داده ها، پارامترهای الگوریتم و معیارهای ارزیابی مورد استفاده برای ارزیابی توصیف، سپس با استفاده از الگوریتم رده بندی برای کاهش داده ها، عملکرد الگوریتم ها در چهار مجموعه داده ارزیابی می شود؛ در نهایت، نتایج به دست آمده ارائه و در مورد آنها بحث خواهد شد. برای آزمایش IFSB-ReliefF، مجموعه داده های استفاده شده شامل انواع داده های مختلف (عددی و غیر عددی) و مقادیر مفقود است. مشخصات مجموعه داده های استفاده شده در جدول (۱) ارائه شده است.

جدول (۱): مشخصات مجموعه های داده

(Table-1): 1 Datasets characteristics

دارای مقادیر مفقود	نوع ویژگی	تعداد کلاس	تعداد ویژگی	تعداد نمونه	مجموعه داده
خیر	عددی	10	784	50,000	MNIST (test part) ^۱
خیر	عددی	3	4	625	Balance-Scale ²
بله	عددی و غیر عددی	2	41	30,000	Census ^۳
خیر	عددی و غیر عددی	2	20	1000	German credit ⁴

مشخصات سخت افزاری و نرم افزاری سامانه عبارتند از:

- پردازنده: Intel® Xeon® CPU E5-2620 v4 @ 2.10GHz (2 processors)

¹<http://www.cs.ubc.ca/labs/beta/Projects/autoweka/datasets/mnist.zip>

² <http://archive.ics.uci.edu/ml/datasets/balance+scale>

³ <https://archive.ics.uci.edu/ml/datasets/Census-Income+-%28KDD%29>

⁴[https://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))

رده برای تشخیص مهم تر هستند. در حقیقت، نمونه هایی که بسیار شبیه یکدیگرند، نمونه های تکراری هستند. آنها فقط اندازه مجموعه داده را افزایش می دهند و عملکرد و دقت طبقه بندی را بهبود نمی دهند.

در خط ۴.e، با استفاده از مقادیر مجموعه Hit و Miss وزن هر ویژگی نیز محاسبه می شود. گفتنی است که p_{C} و $p(\text{class}(x_t))$ احتمال پیشین هر رده، یعنی نسبت نمونه های هر رده به کل نمونه ها، است. این احتمال باید قبل از شروع الگوریتم محاسبه شود. پس از پایان حلقه اصلی برنامه در خط ۵، دو آرایه از وزن وجود دارد، یکی برای نمونه (W) و یکی برای ویژگی های هر نمونه (WFI).

در خط ۶ برای به دست آوردن وزن نهایی هر ویژگی، میانگین مقادیر آرایه WFI برای هر ویژگی محاسبه می شود و در آرایه WF قرار می گیرد. در خط ۷، مقادیر آرایه های WF و W برای شناسایی نمونه ها و ویژگی های بهتر، مرتب می شوند.

در خط ۸، بر اساس p_{C} نمونه هایی با وزن بالاتر انتخاب می شوند. یکی از اصلاحاتی که در IFSB-ReliefF انجام شده، درنظر گرفتن نسبت نمونه های انتخاب شده از هر رده نسبت به همه نمونه های آن رده است. دلیل این امر این است که برخی از الگوریتم ها، مانند الگوریتم All k-NN [34]، احتمال پیشین هر رده را در نظر نگرفته و انتخاب کورکرانه ای انجام می دهند. این ممکن است باعث عدم تعادل بین نسبت رده ها شود. همچنین در برخی موارد، که در آن یک رده فقط چند نمونه دارد، ممکن است، تمام نمونه های آن حذف شده و هیچ رده ای از این رده باقی نماند؛ بنابراین، این بهبود در IFSB-ReliefF به حفظ وضعیت اولیه مجموعه داده ها کمک می کند. البته، روش محاسبه را می توان برای نگهداری تعداد بیشتری از نمونه های رده اقلیت و حل مشکل مجموعه داده های نامتوافق استفاده کرد. این کار می تواند با محاسبه احتمال پیشین هر رده و با انتخاب نمونه هایی با وزن بیشتری از هر رده انجام شود. تعداد نمونه های انتخاب شده از هر رده بستگی به میزان کاهش دارد. در نهایت، در خط ۹، فیلتر کردن ستون ها (ویژگی ها) در مجموعه نمونه انتخاب شده انجام می شود و ویژگی هایی که وزن بیشتری دارند، حفظ شده و بقیه حذف می شوند.

خروجی الگوریتم IFSB-ReliefF یک مجموعه داده است که هم از نظر تعداد سطرها و هم از نظر تعداد ستون ها کاهش یافته است. به منظور بررسی موققیت الگوریتم کاهش داده، باید الگوریتم رده بندی بر روی داده های کاهش یافته



رده‌بندی می‌شود. این الگوریتم یک درخت تصمیم را تولید می‌کند که می‌تواند مجموعه داده‌هایی با ویژگی‌های گستته و پیوسته و مقادیر مفقود را مدیریت کند [35]. C4.5 از آنروپی اطلاعات برای ایجاد درخت استفاده می‌کند، C4.5 در هر گره، یک ویژگی را انتخاب می‌کند که نمونه‌های هر رده را به طور مؤثرتر جدا می‌کند. در تمام آزمایش‌ها (به جزء آزمایش مقایسه با مقالات دیگر در جدول (۵) و جدول (۸)، ۶۶٪ از داده‌ها برای یادگیری و بقیه برای آزمایش استفاده شدند). این یکی از روش‌های استاندارد برای آموزش و آزمایش سامانه‌ها در WEKA است. البته روش‌های دیگری مانند K-fold cross validation نیز وجود دارند که برای این کار استفاده می‌شوند، اما به دلیل حجم زیاد داده‌ها، به ویژه در مجموعه‌داده‌های اولیه و بدون کاهش، استفاده از آن‌ها موجب افزایش زمان رده‌بندی یا عدم امکان اجرا و بروز خطا می‌شود. فقط در مورد مقایسه دو مجموعه داده german credit و Balance-scale (در جدول (۵) و جدول (۸)) از روش 10-fold cross validation استفاده شده است.

برای ارزیابی نتایج، معیارهای کارایی زیر به کار رفته‌اند:

- نرخ کاهش: این معیار نرخ کاهش داده را مشخص کرده و مشخص می‌کند چه مقدار از مجموعه داده حذف شده است. و با فرمول زیر به دست می‌آید [36, 1]:

$$\text{نرخ کاهش} = \text{اندازه مجموعه کاهش‌یافته} / \text{اندازه مجموعه اولیه}. \quad (3)$$

در این مقاله این معیار با واحد درصد نشان داده می‌شود. گفتنی است که در بسیاری از الگوریتم‌های کاهش داده مانند All k-NN، نمی‌توان نرخ کاهش داده را از ابتدا مشخص کرد و خود الگوریتم با توجه به پارامترهای تصمیم می‌گیرد که چه داده‌ای را حذف کند. هم‌چنین، چون الگوریتم‌های All k-NN و ReliefF فقط کاهش ویژگی یا کاهش نمونه را انجام می‌دهند، حتی اگر نرخ کاهش آن‌ها با IFSB-ReliefF برابر باشد، نمی‌تواند نشان‌دهنده کارایی یکسان آن‌ها باشد؛ بنابراین، نرخ کاهش الگوریتم می‌تواند به عنوان یک پارامتر مهم در انتخاب آن باشد. در عین حفظ کیفیت داده، هر قدر نرخ کاهش بیشتر باشد، نشان‌دهنده برتری الگوریتم خواهد بود.

- دقت: این معیار نشان‌دهنده تعداد نمونه‌هایی که به طور صحیح رده‌بندی شده‌اند نسبت به کل نمونه‌ها است.

- RAM: ۶۴ گیگابایت
 - سیستم عامل: Win 7 64 bit
 - نرم‌افزارهای مورد استفاده برای کاهش داده‌ها با سایر الگوریتم‌های کاهش و هم‌چنین رده‌بندی داده‌ها: WEKA 3.8 و Rapid Miner
 - نرم‌افزار مورد استفاده برای اجرای IFSB ReliefF Matlab r2013
- برای ارزیابی نتایج IFSB-ReliefF، آن‌ها با نتایج برخی از الگوریتم‌های انتخاب نمونه و ویژگی مقایسه شدند. به همین ترتیب، نتایج رده‌بندی داده‌های کاهش‌یافته با نتایج مجموعه داده‌های اصلی مقایسه شد.
- All k-NN الگوریتم مورد استفاده برای انتخاب نمونه‌ها است. در تمام نمونه‌هایی که به درستی به وسیله نزدیک‌ترین k تعداد همسایگانشان رده‌بندی نشده‌اند (برای $i = 1 .. k$) حذف می‌شوند. الگوریتم اولیه نیز برای انتخاب ویژگی‌ها استفاده می‌شود؛ علاوه بر این، چون IFSB-ReliefF همزمان کاهش نمونه‌ها و ویژگی‌ها را انجام می‌دهد، ترکیبی از All k-NN و ReliefF نیز بر روی داده‌ها انجام می‌شود. یعنی، در ابتدا، تعدادی از نمونه‌ها با استفاده از All k-NN انتخاب شده و سپس از میان نمونه‌های انتخاب شده، تعدادی از ویژگی‌ها بر اساس میزان کاهش مورد نظر انتخاب و بقیه حذف می‌شوند.

جدول (۲) پارامترهای هر الگوریتم را نشان می‌دهد. گفتنی است که نرخ کاهش ۵۰٪ و مقدار L به صورت تجربی انتخاب شده و سایر پارامترها پیش‌فرضهای مورد استفاده در WEKA و Rapid Miner است. البته این مقادیر پیش‌فرض بر اساس مقالات متعدد و مقدار استفاده شده در بیشتر آنها است. برای مثال در [24] گفته شده است که برای بیشتر اهداف انتخاب مقدار $B=10$ مطمئن خواهد بود.

جدول (۲): پارامترهای الگوریتم‌ها

(Table-2): Algorithm parameters

الگوریتم‌ها	پارامترها
ReliefF	<p>k start:3 k stop:10</p> <p>تعداد نمونه‌های انتخاب شده (L): ۵۰٪ نمونه‌ها</p> <p>تعداد همسایه‌ها (B): 10</p> <p>تعداد ویژگی‌های انتخاب شده بعد از رتبه بندی ویژگی‌ها: ۵۰٪ ویژگی‌ها</p>

برای ارزیابی کیفیت نتایج حاصل از IFSB-ReliefF مجموعه داده‌های کاهش‌یافته به وسیله الگوریتم C4.5



به صورت موازی ۸۶۷۱ ثانیه است، در صورتی که اگر الگوریتم به صورت غیرموازی اجرا شود، زمان اجرا ۵۴۹۲۳ ثانیه خواهد بود. همچنین در مورد تابع ارزیابی مورد استفاده در IFSB-ReliefF نتایج شاخص ژاکارد و فاصله منهتن در نرخ کاهش ۰/۷۵٪، از نظر میزان دقت ردهبندی و RMSE به طور تقریبی مشابه هم هستند و فقط در نرخ کاهش ۰/۹۱٪، فاصله منهتن عملکرد بهتری داشته است. از نظر میزان زمان مصرف شده نیز به طور تقریبی مشابه‌اند، و فقط روش ژاکارد، کمی زمان بیشتری صرف می‌کند.

در جدول (۴) نتایج مربوط به مجموعه‌داده German credit ارائه شده است. در این جدول نیز بهترین دقت مربوط به روش All k-NN است، اما همان‌گونه که مشاهده می‌شود، اگرچه زمان کمی برای اجرا صرف شده، نرخ کاهش تنها ۳۰.۲٪ است، در صورتی که روش پیشنهادی این مقاله با فاصله منهتن و نرخ کاهش ۸۹.۵٪ به نتایجی با شاخص kappa بالاتر و دقیق‌تر است. در این مجموعه داده، شاخص ژاکارد اگرچه در شاخص‌های ردهبندی عملکرد ضعیفتری از فاصله منهتن داشته، اما زمان مورد نیاز برای کاهش آن، به طور تقریبی ۲/۵ برابر کمتر بوده و همچنین، نتایج به دست آمده به وسیله روش IFSB-ReliefF مقایسه شده است. برای مقایسه بین روش‌ها از ردهبند KNN با استفاده شده است. البته مقاله [۱]، مقدار K را به طور صریح بیان نکرده و ممکن است از مقادیر بیشتری برای K استفاده کرده باشد که تأثیر زیادی بر دقت به دست آمده دارد. برای مثال در روش IFSB-ReliefF اگر مقدار K=2 باشد، دقت به دست آمده برای نرخ تابع ارزیابی فاصله منهتن (مورد آخر جدول) ۷۱.۶۶٪ خواهد بود. متأسفانه هیچ کدام از این دو مقاله سخت‌افزار مورد استفاده خود را بیان نکرده‌اند تا زمان اجرای الگوریتم با توجه به آن سنجیده شود؛ اما با توجه به این که هر دو الگوریتم [۱] و [۳۰] از الگوریتم ژنتیک و ردهبندها برای بررسی و کاهش نمونه‌ها و پیشگی‌ها استفاده می‌کنند، و الگوریتم‌های ژنتیک ذاتاً زمان زیادی را صرف ارزیابی یک جواب و محاسبه میزان بهینگی آن می‌کنند، زمان مورد استفاده آن‌ها به احتمال بیشتر از روش پیشنهادی خواهد بود. همچنین همان‌گونه که در جدول‌های (۴) و (۵) مشخص است، مقدار شاخص Kappa برای IFSB-ReliefF بیشتر از سایر روش‌ها است. این شاخص نشان‌دهنده تصادفی بودن یا نبودن نتایج است و هر چقدر به یک نزدیک باشد، یعنی نتایج آزمایش از ثبات بیشتری برخوردار است.

- زمان اجرا: زمان لازم برای اجرای الگوریتم‌های کاهش را نشان می‌دهد. زمان با واحد ثانیه نشان داده می‌شود.

- شاخص Kappa: این شاخص برای اندازه‌گیری میزان تصادفی بودن نتایج است. مقدار آن بین ۱-۰ تا ۱ متغیر است که مقدار صفر نشان‌دهنده تصادفی بودن نتایج است و هرچه به مقدار یک نزدیک باشد، احتمال تصادفی بودن نتایج کمتر است [۱].

$$\text{Kappa} = \frac{n \sum_{i=1}^c x_{ii} - \sum_{i=1}^c x_{i\cdot} \cdot x_{\cdot i}}{n^2 - \sum_{i=1}^c x_{i\cdot} \cdot x_{\cdot i}} \quad (4)$$

در فرمول (۴)، x_{ii} تعداد سلول‌ها در قطر اصلی ماتریس اغتشاش، n تعداد نمونه‌ها، c تعداد رده‌ها و $x_{i\cdot}$ و $x_{\cdot i}$ ستون‌ها و سطرهای تعداد کل هستند.

- انحراف معیار (RMSE): RMSE تفاوت بین مقادیر پیش‌بینی شده با ردهبند برای داده‌های کاهش‌یافته و نتایج واقعی را نشان می‌دهد [۳۷]. در واقع، RMSE خطاهای پیش‌بینی ردهبند را نشان می‌دهد. یک تخمین گر $\hat{\theta}$ نسبت به پارامتر تخمین زده شده به صورت زیر تعریف می‌شود:

$$\text{RMSE}(\hat{\theta}) = \sqrt{\text{MSE}(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)}. \quad (5)$$

جدول (۳) نتایج مجموعه‌داده MNIST را نشان می‌دهد. همان‌گونه که در جدول دیده می‌شود، الگوریتم ReliefF تنها ویژگی‌ها را کاهش می‌دهد و دقت ردهبندی تفاوت زیادی با مجموعه داده اولیه ندارد. همچنین، All k-NN فقط نمونه‌ها را کاهش می‌دهد و نرخ کاهش آن هم بسیار پایین است و در واقع می‌توان گفت که همه نمونه‌ها حفظ شده‌اند. البته دقت ردهبندی این الگوریتم بالاتر از مجموعه‌داده اولیه است؛ اما هدف کاهش داده برآورده All k-NN نمی‌شود. همچنین اگر چه ترکیب ReliefF و RMSE را دقت را به میزان زیادی افزایش داده و خطای کاهش می‌دهد، اما نرخ کاهش همچنان رضایت‌بخش نیست. علاوه‌بر این، زمان اجرای هر سه الگوریتم بادشده، به ویژه ترکیب ReliefF و All k-NN بسیار بیشتر از الگوریتم IFSB-ReliefF است. دلیل کاهش زمان ReliefF، استفاده از ساختار کمتر بودن آن حتی از ReliefF اولیه، استفاده از برنامه‌نویسی موازی است که به الگوریتم اجازه می‌دهد حلقه اصلی برنامه را به طور مستقل برای هر یک از نمونه‌ها اجرا کند. برای مثال، زمان اجرا برای IFSB-ReliefF با شاخص ژاکارد با نرخ کاهش ۰/۷۵٪ روی مجموعه داده MNIST

جدول-۳: نتایج اعمال الگوریتم‌های کاهش داده بر مجموعه‌داده MNIST
(Table-3): The results of applying data reduction algorithms on MNIST data set.

Kappa	RMSE	زمان اجرا (ثانیه)	دقت	تعداد ویژگی باقی‌مانده	تعداد نمونه باقی‌مانده	نرخ کاهش	الگوریتم کاهش داده
0.8503	0.1575	-	86.53	784	50000	0 %	Original data set
0.6023	0.1567	34975	86.73	392	50000	50 %	ReliefF
0.8643	0.1499	61320	87.80	784	48843	2.31 %	All k-NN
0.4802	0.1463	94623	88.45	392	48843	51.15 %	All k-NN+ ReliefF
0.8341	0.1657	8671	85.07	392	25000	75 %	IFSB-ReliefF with Jaccard Index
0.7962	0.1898	8649	80.49	235	15000	91 %	IFSB-ReliefF with Jaccard index
0.8374	0.1634	8237	85.38	392	25000	75%	IFSB-ReliefF with Manhattan distance
0.8252	0.1708	8313	84.27	235	15000	91 %	IFSB-ReliefF with Manhattan distance

جدول-۴: نتایج اعمال الگوریتم‌های کاهش داده بر مجموعه‌داده German Credit
(Table-4): The results of applying data reduction algorithms on German Credit data set

Kappa	RMSE	زمان اجرا (ثانیه)	دقت	تعداد ویژگی‌های باقی‌مانده	تعداد نمونه‌های باقی‌مانده	نرخ کاهش	الگوریتم کاهش داده
0.2687	0.4836	-	72.65	20	1000	0%	Original data set
0.284	0.4221	1	74.70	10	1000	50%	ReliefF
0.3737	0.298	2	90.29	20	698	30.2%	All k-NN
0	0.3329	2.8	87.34	10	698	65.1%	All k-NN+ ReliefF
0.2793	0.4574	44.7	71.76	10	500	75 %	IFSB-ReliefF with Jaccard Index
0.4344	0.4031	44.3	76.47	7	300	89.5%	IFSB-ReliefF with Jaccard index
0.5921	0.3639	112.5	84.12	10	500	75%	IFSB-ReliefF with Manhattan distance
0.7414	0.3067	112.4	89.21	7	300	89.5%	IFSB-ReliefF with Manhattan distance

جدول-۵: مقایسه نتایج بدست آمده به وسیله IFSB-ReliefF و روش‌های دیگر برای مجموعه‌داده German credit
(Table-5): Comparison between results obtained by IFSB-ReliefF and other methods for German credit

Kappa	(زمان اجرا(ثانیه)	دقت	تعداد ویژگی‌ها	تعداد نمونه‌ها	الگوریتم کاهش داده
0.2335	555.26	71.8	-	-	IFSB-CoCo [1]
-	-	69.2	8	55	GA:FSIS [30]
-	-	67.2	6	210	GA:ISFS [30]
0.178	44.7	66.2	10	500	IFSB-ReliefF with Jaccard Index
0.3369	44.3	71	7	300	IFSB-ReliefF with Jaccard index
0.194	112.5	66.6	10	500	IFSB-ReliefF with Manhattan distance
0.2295	112.4	68.67	7	300	IFSB-ReliefF with Manhattan distance

الگوریتم پیشنهادی کمتر بوده و فقط ویژگی‌ها را کاهش می‌دهد. همچنین، All k-NN به نتایجی با دقت بالا (حتی دقیق‌تر از IFSB-ReliefF) نتایج خوبی داشته است.

جدول (۶) نتایج را برای مجموعه‌داده Census نشان می‌دهد. اگرچه ReliefF برای کاهش داده زمان کمی صرف کرده و دقت نتایج آن خوب است، اما نرخ کاهش آن از



بیشتری عمل کرده است. البته این الگوریتم زمان بیشتری برای اجرا صرف می‌کند که دلیل آن وجود داده‌های غیر عددی و مقادیر مفقود و زمان بربودن محاسبه شباخت بین این مقادیر است. در مورد این مجموعه داده نیز، شاخص ژاکارد زمان کمتری برای کاهش داده مصرف کرده است، اما در شاخص‌های رده‌بندی عملکرد ضعیفتری از فاصله منهتن داشته است.

مجموعه داده اولیه) می‌رسد، اما نرخ کاهش ان بسیار پایین بوده و در عمل به هدف کاهش داده‌ها دست نمی‌یابد. چنان‌که در جدول دیده می‌شود، الگوریتم IFSB-ReliefF با فاصله منهتن به نتایجی با دقت و بالاتر از همه، در عین اینکه نرخ کاهش داده نیز از همه بیشتر است، دست پاftte است. همچنین نرخ خطای RMSE از همه کمتر و شاخص Kappa نیز بیشتر از همه است. یعنی الگوریتم با ثبات

(جدول-۶): نتایج اعمال الگوریتم‌های کاهش داده بر مجموعه داده Census

(Table-6): The results of applying data reduction algorithms on Census dataset.

Kappa	RMSE	زمان اجرا(ثانیه)	دقت	تعداد ویژگی‌های باقی‌مانده	تعداد نمونه‌های باقی‌مانده	نرخ کاهش	الگوریتم کاهش داده
0.4604	0.1963	-	95.47	41	30000	0 %	Original data set
0.3707	0.2037	490	94.98	20	30000	52 %	ReliefF
0.7202	0.0835	2396	99.23	41	28377	5.41 %	All k-NN
0.4302	0.0954	2832	99.06	20	28377	53 %	All k-NN+ ReliefF
0.239	0.2245	7098	94.13	21	15000	75 %	IFSB-ReliefF with Jaccard Index
0.2375	0.2131	7017	94.50	13	9000	90.48 %	IFSB-ReliefF with Jaccard index
0.9375	0.0946	11529	99.05	21	15000	75 %	IFSB-ReliefF with Manhattan distance
0.9084	0.0788	11201	99.31	13	9000	90.48 %	IFSB-ReliefF with Manhattan distance

(جدول-۷): نتایج اعمال الگوریتم‌های کاهش داده بر مجموعه داده Balance-scale

(Table-7): The results of applying data reduction algorithms on Balance-scale data set

Kappa	RMSE	زمان اجرا(ثانیه)	دقت	تعداد ویژگی‌های باقی‌مانده	تعداد نمونه‌های باقی‌مانده	نرخ کاهش	الگوریتم کاهش داده
0.5495	0.3651	-	74.057	4	625	0 %	Original data set
0.3511	0.3978	0.9	64.62	2	625	50%	ReliefF
0.7496	0.3373	1	87.5	4	565	9.6%	All k-NN
0.4701	0.4317	1.7	73.44	2	565	54.8%	All k-NN+ ReliefF
0.2163	0.4269	5.15	56.60	2	313	74.96%	IFSB-ReliefF with Jaccard Index
0.2149	0.4128	5.45	54.69	1	187	92.52%	IFSB-ReliefF with Jaccard index
0.5972	0.345	1.6	78.302	2	313	74.96%	IFSB-ReliefF with Manhattan distance
0.6007	0.3485	2.2	78.125	1	187	92.52%	IFSB-ReliefF with Manhattan distance



(جدول-۸): مقایسه نتایج به دست آمده توسط IFSB-ReliefF و روش‌های دیگر برای مجموعه‌داده Balance-scale

(Table-8): Comparison between results obtained by IFSB-ReliefF and other methods for Balance-scale

Kappa	زمان اجرا(ثانیه)	دقت	تعداد ویژگی‌ها	تعداد نمونه‌ها	الگوریتم کاهش داده
0.7578	69.58	84.95	-	-	IFS-CoCo [1]
-	-	72.70	3	68	GA:FSIS [30]
-	-	62.38	2	28	GA:ISFS [30]
0.1361	5.15	53.35	2	313	IFSB-ReliefF with Jaccard Index
0.2871	5.45	61.49	1	187	IFSB-ReliefF with Jaccard index
0.6391	1.6	80.511	2	313	IFSB-ReliefF with Manhattan distance
0.5842	2.2	77.54	1	187	IFSB-ReliefF with Manhattan distance

۱. اگرچه الگوریتم All k-NN بالاترین میانگین دقت را به دست آورده است، اما کمترین نرخ کاهش را دارد و سبب می‌شود که برای کاهش داده در مجموعه‌داده‌های بزرگ مفید نباشد؛ علاوه بر این، همان‌گونه که در نتایج مجموعه‌های Balance-scale گفته شد، این الگوریتم برای مجموعه‌های نامتوازن مناسب نیست و ممکن است که نمونه‌های یک رده به طور کلی حذف شود.

۲. اگرچه می‌توان نرخ کاهش را در الگوریتم ReliefF تعیین کرد، اما این الگوریتم چند ضعف در زمینه کاهش داده دارد، نخست این که فقط کاهش ویژگی را انجام می‌دهد و در مجموعه‌داده‌ای که تعداد نمونه زیاد و تعداد ویژگی کمی دارند، چندان مناسب نیست. دوم این که شاخص Kappa این الگوریتم پایین و دلیل آن انتخاب تصادفی برخی نمونه‌ها برای اجرای الگوریتم است که نتایج را بی ثبات می‌کند.

۳. ترکیب الگوریتم‌های All k-NN و ReliefF، علاوه بر تمام معایب گفته شده، سبب افزایش بسیار زیاد زمان اجرا می‌شود و مقدار شاخص Kappa نیز کاهش می‌یابد.

۴. استفاده از شاخص ژاکارد سبب کاهش زمان اجرای الگوریتم پیشنهادی شده و کمترین زمان اجرا را دارد. البته کمترین میزان دقت در این شاخص، به دست آمده است. اما با داشتن مزایایی مثل مدت زمان اجرا، قابلیت اجرای موازی، حفظ حالت اولیه داده‌ها و نرخ کاهش بالا، برای مجموعه‌داده‌های بزرگ و بهویژه نامتوازن می‌تواند یک انتخاب مناسب باشد.

۵. درنهایت، باثبات‌ترین و دقیق‌ترین نتایج توسط الگوریتم IFSB-ReliefF با فاصله منهتن کسب شده است. البته زمان اجرای این روش، بهویژه در مجموعه‌داده‌های غیر عددی، بیشتر از شاخص ژاکارد است؛ بنابراین می‌توان در مجموعه‌داده‌های غیر عددی از روش شاخص ژاکارد و در داده‌های عددی از فاصله منهتن استفاده کرد.

در جدول (۷) نتایج مجموعه‌داده Balance-scale ارائه شده است. همان‌طور که مشاهده می‌شود، بالاترین درصد دقت مربوط به روش All k-NN است؛ اما این روش نرخ کاهش کمی دارد. همچنین این مجموعه‌داده یک مجموعه نامتوازن است و همان‌گونه که در بخش ۳ گفته شد، All k-NN مانند بسیاری از روش‌های انتخاب نمونه، این مسئله را در نظر نمی‌گیرد. برای مثال با اجرای All k-NN روی مجموعه‌داده Balance-scale، یکی از رده‌ها که تعداد نمونه‌های کمتری داشت، به طور کلی از مجموعه‌داده حذف شد. پس از All k-NN، روش پیشنهادی این مقاله با تابع فاصله منهتن بیشترین دقت را کسب کرده است که با توجه به نرخ کاهش زیاد و زمان اجرای کم و همچنین، حفظ نسبت نمونه‌های هر رده، می‌تواند روش مناسبی برای کاهش این مجموعه داده محسوب شود.

در جدول (۸) نتایج به دست آمده توسط IFSB-ReliefF با نتایج مقالات [1] و [30] مقایسه شده است. همان‌گونه که ملاحظه می‌شود، دقت به دست آمده به وسیله IFSB-ReliefF از دقت روش مقاله [30] بهتر است. همچنین اگرچه دقت روش مقاله [1] بهتر از IFSB-ReliefF است، اما این روش زمان زیادی صرف کرده است. البته همان‌گونه که پیش تر بیان شد، سخت‌افزارهای مورد استفاده و پارامترها برای اجرای الگوریتم‌ها در این مقاله‌ها بیان نشده است.

۵- بحث و یافته‌های پژوهش

در این بخش به اختصار نتایج روش‌های مختلف با هم مقایسه شده و نتایجی از یافته‌ها مستخرج می‌شود.

در جدول (۹) میانگین نتایج مربوط به روش‌های مختلف مشاهده می‌شود.

با توجه به مقادیری که در این جدول و سایر نتایج مشاهده می‌شود، می‌توان نکات و یافته‌های این مقاله را به صورت زیر خلاصه کرد:

فصلنامه
پژوهش‌های
علوم روزمره

(جدول-۹): میانگین نتایج مربوط به الگوریتم های مختلف

(Table-9): The average of results for various algorithms

Kappa	RMSE	زمان اجرا(ثانیه)	دقت	نرخ کاهش	الگوریتم کاهش داده
0.5322	0.3006	-	82.18	0%	Original data set
0.4020	0.2951	8867	80.26	51%	ReliefF
0.6770	0.2172	15930	91.21	12%	All k-NN
0.3451	0.2516	24365	87.07	56%	All k-NN+ ReliefF
0.3922	0.3186	3946	76.89	75%	IFSB-ReliefF with Jaccard Index
0.4208	0.3047	3921	76.54	91%	IFSB-ReliefF with Jaccard index
0.7411	0.2417	4952	86.71	75%	IFSB-ReliefF with Manhattan distance
0.7689	0.2262	4889	87.73	91%	IFSB-ReliefF with Manhattan distance

مجموعه داده های مختلف آزمایش و همچنین نتایج حاصل از سایر الگوریتم های را با آن ها مقایسه کنیم.

7- References

- [1] J. Derrac, S. Garcia and F. Herrera, "IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule," *Pattern Recognition*, vol. 43, no. 6, pp. 2082-2105, 2010.
- [2] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 4, pp. 491-502, 2005.
- [3] V. Bolón-Canedo, N. Sánchez-Marcano and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowledge-Based Systems*, vol. 86, pp. 33-45, 2015.
- [4] L. C. Molina, L. Belanche and À. Nebot, "Feature Selection Algorithms: A Survey and Experimental Evaluation," in *IEEE International Conference on Data Mining*, 2002.

۷- مراجع

۶- جمع بندی

رشد بی رویه داده ها و وجود داده های تکراری و دارای نویفه، نه تنها نیاز به دستگاه های ذخیره سازی عظیم دارد، بلکه دقت الگوریتم یادگیری ماشین را کاهش می دهد؛ بنابراین به روش هایی برای کاهش داده ها و استخراج اطلاعات مفید نیاز است. در این مقاله، الگوریتم انتخاب ویژگی ReliefF به رو شی متفاوت استفاده می شود. در الگوریتم IFSB-ReliefF مشابهت نمونه ها اندازه گیری شده و نزدیک ترین نمونه ها به هم تعیین می شوند. بر اساس مشابهت محاسبه شده، نمونه های خیلی شبیه به هم حذف می شوند. همچنین، بر اساس همین مجموعه همسایگی، وزن هر ویژگی محاسبه و ویژگی ها با وزن کمتر حذف می شوند. نتایج نشان می دهند که اجرای این الگوریتم داده ها را در زمانی مناسب و با نرخ IFSB-ReliefF کاهش بالایی کاهش می دهد؛ علاوه بر این، IFSB-ReliefF می تواند دقت رده بندی را در محدوده مجموعه داده اولیه حفظ کرده و در مواردی حتی بهبود بخشد. همچنین، این الگوریتم با درنظر گرفتن احتمال پیشین هر رده، نسبت بین نمونه ها در مجموعه داده اولیه از بین نمی رود. بدلیل این که دستورها در حلقه اصلی الگوریتم در تکرارهای مختلف از هم مستقل اند، بنابراین می توان آن را با یک سامانه چند پردازنده به صورت موازی اجرا کرد. در پژوهش های بعدی قصد داریم این الگوریتم را با توابع ارزیابی متفاوت و روی

[۵] ج. پورامینی، ب. مینایی بیدگلی و م. اسماعیلی، "یک روش جدید انتخاب ویژگی یک طرفه در دسته بندی داده های متغیر نامتوازن،" پردازش



- methods based on information entropy," in *IEEE International Joint Conference on Neural Networks*, 2004.
- [20] G. Chandrashekhar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16-28, 2014.
- [21] I. Kononenko, E. Šimec and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, no. 1, pp. 39-55, 1997.
- [22] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.
- [23] L. A. R. Kenji Kira, "The feature selection problem: Traditional methods and a new algorithm," *AAAI*, vol. 2, pp. 129-134, 1992.
- [24] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine learning*, vol. 53, no. (1-2), pp. 23-69, 2003.
- [25] H. Liu and H. Motoda, Computational methods of feature selection, CRC Press, 2007.
- [26] K. Yu, X. Xu, M. Ester and H.-P. Kriegel, "Feature weighting and instance selection for collaborative filtering: An information-theoretic approach," *Knowledge and Information Systems*, vol. 5, no. 2, pp. 201-224, 2003.
- [27] T. Chen, X. Zhang, S. Jin and O. Kim, "Efficient classification using parallel and scalable compressed model and its application on intrusion detection," *Expert Systems with Applications*, vol. 41, pp. 5972-5983, 2014.
- [28] W. T. Hadoop, The Definitive Guide, O'Reilly Media, Inc., 2012.
- [29] P. Perner, "Prototype-based classification," *Applied Intelligence*, vol. 28, no. 3, pp. 238-246, 2008.
- [30] C.-F. Tsai, W. Eberle and C.-Y. Chu, "Genetic algorithms in feature and instance selection," *Knowledge-Based Systems*, vol. 39, p. 240-247, 2013.
- [31] F. Dimitris, D. Meretakis and L. Spiros, "Integrating Feature and Instance Selection for text classification," In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002.
- [32] H. Ahn and K.-j. Kim, "Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach," *Applied Soft Computing*, vol. 9, no. 2, pp. 599-607, 2009.
- [33] Z. Abbasi and M. Rahmani, "An Instance Selection Algorithm Based on ReliefF," *International Journal on Artificial Intelligence Tools*, vol. 28, no. 1, p. 1950001, 2019.
- [34] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 499-502, 1991.
- [5] علائم و داده‌ها، ۱۶(۱)، ۲۱-۴۰، ۱۳۹۸
- J. Pouramini, B.Minaei-Bidgoli, M.Esmaili, "A Novel One Sided Feature Selection Method for Imbalanced Text Classification", *JSDP*, 2019, vol. 16 (1), pp.21-40.
- [6] P. S. Bradley, U. M. Fayyad and C. Reina, "Scaling clustering algorithms to large databases," in Proceedings of the Fourth International Conference on Knowledge Discovery & Data Mining, New York, 1998.
- [7] H. Liu, H. Motoda and L. Yu, "A selective sampling approach to active feature selection," *Artificial Intelligence*, vol. 159, pp. 49-74, 2004.
- [8] W. Cochran, Sampling Techniques, New York: Wiley, 1977.
- [9] H. Liu and H. Motoda, Instance Selection and Construction for Data Mining, Boston,MA: Kluwer Academic, 2001.
- [10] S. Garc'ia, J. Derrac, J. R. Cano and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 34, no. 3, pp. 417-435, MARCH 2012.
- [11] S. Garc'ia, J. Derrac, J. R. Cano and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Survey of Methods," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 34, no. 3, pp. 417-435, 2012.
- [12] F. H. M. L. Jose Ramon Cano, "On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining," *Applied Soft Computing*, vol. 6, pp. 323-332, 2006.
- [13] S. d. Río, V. Lopez, J. M. Benítez and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest," *Information Sciences*, vol. 285, pp. 112-137, 2014.
- [14] d. Wilson and t. r. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 257-286, 2000.
- [15] H. Liu and H. Motoda, "On issues of instance selection," *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 115-130, 2002.
- [16] D. R. Wilson and M. Tony R, "Instance pruning techniques," *ICML*, vol. 97, pp. 403-411, 1997.
- [17] p. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, p. 547-579, 1901.
- [18] G. H. J. Ron Kohavi, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273-324, 1997.
- [19] W. Duch, T. Wieczorek, J. Biesiada and M. Blachnik, "Comparison of feature ranking



Systems, Man, and Cybernetics, vol. 6, pp. 448–452, 1976.

- [35] J. R. Quinlan, "Improved use of continuous attributes in c4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.
- [36] I. Triguero, D. Peralta, J. Bacardit, S. García and F. Herrera, "MRPR: A MapReduce solution for prototype reduction in big data classification," *Neurocomputing*, vol. 150, pp. 331–345, 2015.
- [37] R. Hyndman and K. Anne B., "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.



زینب عباسی. دکترای خود را در رشته مهندسی نرم‌افزار در دانشگاه اراک در سال ۱۳۹۸ دریافت کرده است. حوزه پژوهشی ایشان، داده‌کاوی و الگوریتم‌های کاهش داده است.

نشانی رایانامه ایشان عبارت است از:

Zabasi@gmail.com



محسن رحمانی. مدرک دکترای خود را در سال ۱۳۸۷ از دانشگاه علم و صنعت ایران دریافت کردند. ایشان هم‌اکنون عضو هیأت علمی دانشگاه اراک هستند. تخصص ایشان پردازش سیگنال‌های صوتی و هوش مصنوعی است.

نشانی رایانامه ایشان عبارت است از:

m-rahmani@araku.ac.ir



حسین غفاریان. دکترای خود را از دانشگاه علم و صنعت ایران در سال ۱۳۹۲ دریافت کردند و در حال حاضر عضو هیأت علمی دانشگاه اراک هستند. حوزه تخصصی فعالیت ایشان داده‌کاوی و الگوریتم‌های تکاملی است.

نشانی رایانامه ایشان عبارت است از:

H-ghaffarian@araku.ac.ir

