Electronical & Cyber Defence
Vol. 13, No. 1, Spring 2025, Serial No. 49

An Improved Flow Direction Optimization Algorithm for Spam Email Detection

H. Raie[1], F. Soleimanian Gharehchopogh[2] *

*Associate Professor, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Received:2024 /11/30, Revised: 2025/02/19, Accepted: 2025/03/13, Published: 2025/04/21

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1404.13.1.2.7

ABSTRACT

With the advancement of science and technology, the popularity of the Internet, particularly email, has increased significantly. Email spam has become one of the most prevalent forms of cyberattack, primarily used to disseminate malicious content, including commercial advertisements, computer viruses, and misleading information. Cyber attackers often target systems and servers with various types of malware and viruses to compromise or gain unauthorized access to systems or email accounts. This paper presents an improved flow direction algorithm for feature selection and a k-nearest neighbor algorithm for email spam classification. The Flow Direction Algorithm (FDA) typically faces challenges such as getting stuck in local optima and lacking population diversity. To enhance the FDA's capabilities, chaos operators have been introduced to promote population diversity and expedite convergence. The proposed method employs two types of chaos: circular chaos and logistic mapping. The performance of the proposed model was evaluated using the Spambase dataset, which consists of 4601 samples and 57 features. The results demonstrate that the accuracy of the proposed model, particularly with logistic mapping, is higher than that of other methods.

Keywords: Spam email detection, feature selection, flow direction algorithm, chaos mapping

.

*Corresponding Author Email: bonab.farhad@gmail.com

"پدافند الکترونیکی و سایبری"
سال سیزدهم، شماره ۱،بهار ۱۴۰۴، ص۲۸-۱۷
شاپا الکترونیکی: ۸۹۷۹-۲۹۸۰ شاپا چاپی: ۴۳۴۷-۲۳۲۲

تشخیص ایمیل‌های هرزنامه با استفاده از الگوریتم بهینه‌سازی جهت جریان بهبودیافته

حجت راعی[۱]، فرهاد سلیمانیان قره چپق[۲*]

۱- کارشناسی ارشد، ۲- دانشیار، گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه ، ایران

(دریافت: ۱۴۰۳/۰۹/۱۰، بازنگری: ۱۴۰۳/۱۲/۰۱، پذیرش: ۱۴۰۳/۱۲/۲۳، انتشار: ۱۴۰۴/۰۲/۰۱)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1404.13.1.2.7

چکیده

با پیشرفت علم‌وفناوری، محبوبیت روزافزون اینترنت خصوصاً ایمیل به طور گسترده‌ای افزایش‌یافته است. ایمیل هرزنامه یکی از مخرب‌ترین حملات توسط مهاجمان سایبری است که عمدتاً برای انتشار محتوای مخرب ازجمله تبلیغات تجاری، ویروس‌های رایانه‌ای و اطلاعات گمراه‌کننده استفاده می‌شود. مهاجمان معمولاً سیستم‌ها و سرورها را با انواع مختلف بدافزارها و ویروس‌ها هدف قرار می‌دهند و هدف آن‌ها به خطر انداختن یا دسترسی غیرمجاز به حساب‌های سیستم یا ایمیل است. در این مقاله، از الگوریتم جهت جریان بهبودیافته برای انتخاب ویژگی و الگوریتم k نزدیک‌ترین همسایه برای طبقه‌بندی ایمیل هرزنامه استفاده‌شده است. الگوریتم جهت جریان معمولاً معایبی مانند گیرافتادن در بهینه محلی و عدم تنوع جمعیتی دارد. برای افزایش توانایی الگوریتم جهت جریان از عملگرهای آشوب به‌منظور تنوع جمعیتی و همگرایی سریع استفاده‌شده است. درروش پیشنهادی از دو نوع آشوب به نام‌های آشوب دایره‌ای و نگاشت لجستیک استفاده‌شده است. ارزیابی مدل پیشنهادی بر روی مجموعه‌داده *Spambase* با ۴۶۰۱ نمونه و ۵۷ ویژگی انجام‌شده است. نتایج نشان می‌دهد که درصد صحت مدل پیشنهادی با نگاشت لجستیک در مقایسه با روش‌های دیگر بیشتر است.

کلیدواژه‌ها: تشخیص ایمیل هرزنامه، انتخاب ویژگی، الگوریتم جهت جریان، نگاشت آشوب

*رایانامه نویسنده مسئول: bonab.farhad@gmail.com

۱- مقدمه

در دنیای اینترنت، کاربران با یکسری پیام‌های ناخواسته که حتی به علایق و حیطه کاری آنان مرتبط نیست و حاوی مطالب تبلیغاتی یا حتی مخرب هستند، مواجه می‌شوند. ایمیل‌های هرزنامه شامل مجموعه گسترده‌ای از ایمیل‌های تبلیغاتی آلوده و مخرب است که باهدف‌های متفاوت تجاری، تبلیغاتی و حتی خرابکارانه که موجب زیان، از بین رفتن محتوا و داده‌ها و سرقت اطلاعات شخصی به‌صورت انبوه توسط ربات‌ها و شبکه‌های کامپیوتری در حجم عظیم ارسال می‌شود [۱, ۲]. در اغلب موارد، ایمیل‌های هرزنامه حاوی بدافزارهایی هستند که به‌طورمعمول در قالب فایل‌های ضمیمه برای کاربران ارسال می‌شوند و کاربر با بارگیری و اجرای فایل ضمیمه‌شده، کامپیوتر خود را به بدافزار آلوده کرده و موجب سو استفاده افراد سودجو می‌شود [۳, ۴].

استفاده گسترده از ایمیل‌ها برای ارتباطات و سایر معاملات منجر به افزایش تعداد ایمیل‌های هرزنامه در سطح جهانی شده است [۵]. هرزنامه یک تهدید جدی برای دنیای اینترنت و خانواده ایمیل است. اگرچه توضیح مشخصی برای ایمیل‌های هرزنامه (که به آن ایمیل‌های ناخواسته نیز گفته می‌شود) و تفاوت آن با نامه‌های قانونی وجود ندارد، اما ایمیل‌های هرزنامه پیام‌های مفید و درخواست شده نیستند. در بازه زمانی بین سال‌های ۲۰۱۴ تا ۲۰۱۷، پیام‌های هرزنامه ۵۹/۵۶ درصد از ترافیک ایمیل در سراسر جهان را تشکیل می‌دهند که روزانه بیش از ۱۴,۵ میلیارد پیام هرزنامه در سراسر جهان ارسال می‌شود [۶]. ایمیل هرزنامه یک مشکل بزرگ در دنیای فناوری و ارتباطات است که نه‌تنها کاربران اینترنت را تحت تأثیر قرار می‌دهد، بلکه باعث ایجاد یک مشکل دائمی برای شرکت‌ها و سازمان‌ها می‌شود.

مسائل بهینه‌سازی با ابعاد بالا همیشه یک چالش قابل‌توجه برای طبقه‌بندی می‌باشند [۷, ۸]. حذف ویژگی‌های نامربوط و زائد در ایمیل‌های هرزنامه می‌تواند عملکرد طبقه‌بندی را در توزیع کلاس‌ها بهبود دهد. بنابراین، عملیات پیش‌پردازش قبل از طبقه‌بندی ضروری است. انتخاب ویژگی یک ابزار پیش‌پردازش مؤثر است که عملکرد طبقه‌بندی را با کاهش ویژگی‌های اضافی در داده‌هایی با ابعاد بالا بهبود می‌دهد [۹]. روش‌های فیلتر و پوششی از روش‌های اصلی برای انتخاب ویژگی هستند. روش‌های فیلتر، ویژگی‌ها را بدون استفاده از فرآیند آموزشی و تأثیر آن‌ها بر مدل‌ها انتخاب می‌کنند. متداول‌ترین روش‌های فیلتر شامل اطلاعات متقابل [۱۰]، بهره‌ی اطلاعاتی [۱۱]، حداکثر ارتباط و حداقل افزونگی (mRMR) [۱۲] و Relief [۱۳] هستند. این الگوریتم‌ها ویژگی‌ها را عمدتاً بر اساس چهار معیار ارتباط، اطلاعات، فاصله و سازگاری انتخاب می‌کنند. روش‌های پوششی توسط الگوریتم‌های فراابتکاری یک مدل آموزشی برای انتخاب ویژگی‌های مناسب توسط یک مدل طبقه‌بندی تعریف می‌کنند.

الگوریتم‌های فراابتکاری یک فرآیند تکراری تولید می‌کنند که با مفاهیم کاوش و بهره‌برداری از فضای جستجو، یک اکتشاف جدید را هدایت می‌کنند. هدف از بهینه‌سازی، جستجو برای راه حلی است که تابع هدف تعریف شده توسط کاربر را به حداکثر یا حداقل برساند. الگوریتم‌های فراابتکاری ابزار کارآمدی برای دستیابی به راه حل‌های تقریباً بهینه برای مسائل در مقیاس بزرگ هستند. الگوریتم‌های فراابتکاری متعددی در طول سال‌های اخیر برای مسائل مختلف مانند خوشه‌بندی [۱۴]، مسائل مهندسی [۱۵, ۱۶]، انتخاب ویژگی [۱۷]، تشخیص نفوذ [۱۸]، زمانبندی وظایف [۱۹] پیشنهاد شده‌اند. زیرا این الگوریتم‌ها (۱) نسبتاً ساده و پیاده‌سازی آسانی دارند. (۲) به اطلاعات گرادیان نیاز ندارند. (۳) عدم گیرافتادن در بهینه محلی.

در این مقاله، یک روش جدید برای تشخیص ایمیل‌های هرزنامه با استفاده از الگوریتم جهت جریان [۲۰] بهبودیافته پیشنهاد می‌شود. الگوریتم جهت جریان بهبودیافته برای انتخاب ویژگی‌های برجسته برای افزایش دقت تشخیص استفاده می‌شود. الگوریتم جهت جریان یک الگوریتم فراابتکاری جدید است که عملکرد بالایی در اکتشاف و بهره‌برداری دارد. الگوریتم جهت جریان برای تقلید جریان آب به سمت نقطه خروجی با کمترین ارتفاع در حوضه زهکشی طراحی شده است. اساساً، جریان به (نماینده راه‌حل‌های بالقوه در یک مسئله‌بهینه‌سازی) سمت یک همسایه با بهترین تابع هدف هدایت می‌شود. الگوریتم جهت جریان با ایجاد یک جمعیت اولیه در فضای مسئله تعریف شده آغاز می‌شود. هر جریان در این جمعیت دارای موقعیت و ارتفاع است و به سمت مکانی با ارتفاع کمتر حرکت می‌کند و هدف آن رسیدن به بهینه‌ترین نتیجه است. سرعت حرکت هر جریان مستقیماً با شیب ارتباط دارد و جریان در جهتی با کمترین ارتفاع حرکت می‌کند. این الگوریتم در یافتن بهترین و بهینه‌ترین راه حل‌ها برای مسائل با ابعاد بالا موفق بوده است. بهبود این الگوریتم با استفاده از ترکیب نگاشت آشوب انجام شده است. به طور کلی، مشارکت‌های اصلی این مقاله بشرح زیر است:

- الگوریتم بهینه‌سازی جهت جریان بهبودیافته برای انتخاب ویژگی استفاده می‌شود.
- بهبود الگوریتم بهینه‌سازی جهت جریان با استفاده از روش آشوب دایره‌ای و لجستیک انجام می‌شود.
- پس از انتخاب ویژگی‌های بهینه، از روش طبقه‌بندی K نزدیک‌ترین همسایه برای طبقه‌بندی مجموعه ایمیل‌های هرزنامه استفاده می‌شود.

- روش پیشنهادی روی مجموعه‌داده Spambase ارزیابی شده است و با روش‌های دیگر مقایسه شده است.

ساختار کلی این مقاله به شرح زیر سازماندهی شده است: در بخش دوم مطالعات پیشین را توضیح می‌دهیم. در بخش سوم، مراحل مدل پیشنهادی را توضیح می‌دهیم. در بخش چهارم، به ارزیابی و نتایج مدل پیشنهادی و مقایسه آن با مدل‌های دیگر می‌پردازیم و نهایتاً در بخش پنجم به نتیجه‌گیری و کارهای آینده خواهیم پرداخت.

۲- مطالعات پیشین

تشخیص ایمیل هرزنامه بر مبنای ترکیب الگوریتم جستجوی هماهنگی با الگوریتم بهینه‌سازی مغناطیسی انجام شده است [۲۱]. روش پیشنهادی به منظور انتخاب ویژگی‌های موثر استفاده شده است و سپس طبقه‌بندی با استفاده از الگوریتم k نزدیکترین همسایه انجام شده است. با استفاده از الگوریتم بهینه‌سازی مغناطیسی، بهترین ویژگی‌ها برای الگوریتم جستجوی هارمونی پیدا شده و ماتریس هارمونی برمبنای آنها تشکیل شده است و سپس الگوریتم جستجوی هارمونی برمبنای به‌روزرسانی و نرخ تغییرات گام در هر مرحله بردارهای هارمونی را تغییر داده است تا در میان آنها بهترین بردار به عنوان بردار ویژگی‌ها انتخاب شود. نتایج ارزیابی‌های الگوریتم ارائه شده نشان داد که دقت تشخیص برابر با ۹۷/۱۹ درصد بوده است.

یک مدل جدید مبتنی بر یادگیری عمیق با استفاده از واحد بازگشتی (GRU) برای شناسایی کارآمد ایمیل‌های هرزنامه ارائه شده است [۲۲]. الگوریتم GRU با تجزیه و تحلیل و استخراج ویژگی‌های ایمیل برای شناسایی و طبقه‌بندی آنها به کلاس‌های هرزنامه و غیرهرزنامه استفاده شده است. مجموعه داده استفاده شده حاوی ۵۵۷۲ نمونه جمع‌آوری است. مجموعه داده به دو بخش آموزشی(۸۰ درصد) و آزمایشی(۲۰ درصد) تقسیم شده است.نتایج تجربی نشان می‌دهد که دقت روش پیشنهادی در حدود ۹۸/۹۲ درصد بدست آمده است.

روش جدیدی برای تشخیص هرزنامه با استفاده از الگوریتم بهینه‌سازی گله اسب پیشنهاد شده است [۲۳]. ابتدا الگوریتم بهینه‌سازی گله اسب از حالت پیوسته به گسسته تبدیل شده است. نتایج ارزیابی نشان می‌دهد که روش پیشنهادی در مقایسه با روش‌های دیگر مانند kنزدیک‌ترین همسایه با بهینه‌سازی گرگ خاکستری، k نزدیک‌ترین همسایه، پرسپترون چندلایه، ماشین بردار پشتیبان و بیزین ساده بهتر عمل می‌کند. نتایج نشان می‌دهد که دقت طبقه‌بندی الگوریتم بهینه‌سازی گله اسب دودویی مبتنی بر مخالفت چندهدفه روی مجموعه داده‌های UCI موفق‌تر از روش‌های فراابتکاری استاندارد بوده است. با توجه به یافته‌ها، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های مشابه، در تشخیص ایمیل‌های هرزنامه به طور قابل ملاحظه‌ای دقیق‌تر است و پیچیدگی محاسباتی کمتری را نشان می‌دهد.

رویکرد انتخاب ویژگی جهت افزایش دقت طبقه‌بندی بر اساس دو بهینه‌ساز یعنی الگوریتم خفاش و الگوریتم گرگ خاکستری پیشنهاد شده است [۲۴]. در روش‌های فیلترینگ، نرخ تشخیص مثبت کاذب به دلیل افزایش تعداد زیادی ویژگی‌ها در حال افزایش است. در نتیجه استفاده از روش‌های بهینه‌سازی برای به‌روزرسانی و انتخاب ویژگی‌ها ضروری هستند. انتخاب ویژگی یکی از موثرترین راه‌ها برای افزایش عملکرد سیستم‌های تشخیص و طبقه‌بندی داده‌ها است. در الگوریتم خفاش به‌روزرسانی موقعیت مبتنی بر حرکت جهت حل مشکل همگرایی تقویت شده است. یک به‌روزرسانی جدید پرواز لوی مبتنی بر الگوریتم گرگ خاکستری برای تولید راه‌حل بهینه محلی معرفی شده است. ارزیابی‌ها روی دو مجموعه داده هرزنامه (Spambase و SpamAssassin) با ویژگی‌های مهم انجام شده است. نتایج نشان می‌دهد که دقت تشخیص دو الگوریتم بهینه‌سازی بالای ۹۰ درصد بوده است.

مدل‌های ماشین بردار پشتیبان، درخت J48، شبکه بیزین و Lazy IBK که از تکنیک‌های داده‌کاوی هستند بر روی ۴۳۰۷ ایمیل (۶۳۵ ایمیل هرزنامه) تست و اجرا شده‌اند. نتایج نشان داد که درصد صحت در J48 برابر ۹۳/۱۳ است که در مقایسه با مدل‌های دیگر بیشتر است [۲۵]. مدل ترکیبی NB-K-Means برای تشخیص ایمیل هرزنامه برروی مجموعه داده‌های Ling-spam ، Assassin و Spambase تست و اجرا شده است. از مدل بیزین ساده به منظور فاصله تشابه و از K-Means برای مشابه بودن نمونه‌های داخل هر خوشه استفاده شده است. نتایج نشان داد که دقت مدل ترکیبی در مقایسه با بیزین ساده بیشتر است [۲۶].

مدل‌های درخت تصمیم‌گیری، ماشین بردار پشتیبان و شبکه عصبی مصنوعی پس انتشار و ترکیب آنها بر روی دو مجموعه‌داده با ۱۴ ویژگی تست و اجرا شده‌اند. مجموعه داده اولی شامل ۵۰۴ ایمیل (۳۳۶ غیرهرزنامه و ۱۶۸ هرزنامه) و مجموعه داده دومی شامل ۶۵۷ ایمیل (۳۸۷ غیرهرزنامه و ۲۷۰ هرزنامه) است. در مدل درخت تصمیم‌گیری از آنتروپی، مدل ماشین بردار پشتیبان از تابع کرنل و شبکه عصبی مصنوعی پس انتشار از خطای میانگین استفاده شده است. نتایج برروی مجموعه داده اولی نشان داد که درصد صحت در مدل ترکیبی برابر ۹۱/۰۷ و در مدل‌های درخت تصمیم‌گیری، ماشین بردار پشتیبان و شبکه عصبی مصنوعی پس انتشار به ترتیب برابر ۸۹/۸۸، ۸۸/۶۹، ۸۹/۸۸ بوده است و برروی مجموعه داده دومی درصد صحت در مدل تریبی برابر با ۹۱/۷۸ و در مدل‌های درخت تصمیم‌گیری، ماشین بردار پشتیبان و شبکه عصبی مصنوعی پس انتشار به ترتیب برابر

۹۰/۸۷، ۹۰/۸۷ و ۸۹/۰۴ بوده است [۲۷].

ماشین بردار پشتیبان بر مبنای تابع شعاعی پایه برای تشخیص ایمیل هرزنامه پیشنهاد شده است. مدل شبکه تابع شعاعی پایه از منحنی‌های نرمال در اطراف نقاط داده استفاده می‌کند و آنها را طوری باهم جمع می‌کند که مرز تصمیم را بتوان به نوعی تعریف کرد. با استفاده از تابع شعاعی پایه، فضای ویژگی از طریق یک تبدیل غیرخطی به دست می‌آید. برای ارزیابی مدل ماشین بردار پشتیبان بر مبنای تابع شعاعی پایه از مجموعه داده‌های spambase با ۴۶۰۱ نمونه و SMS Spam با ۵۵۷۴ نمونه استفاده شده است. نتایج نشان داد که مدل ماشین بردار پشتیبان بر مبنای تابع شعاعی پایه نمونه هرزنامه را با دقت ۹۳/۹۲ تشخیص داده است [۲۸]. مدل‌های مختلف دودویی برگرفته از الگوریتم بهینه‌سازی ملخ برای انتخاب ویژگی‌های بهینه پیشنهاد شده‌اند [۲۹]. دو مکانیسم مبتنی بر توابع انتقال سیگموئید و V شکل با نام‌های BGOA-S و BGOA-V پیشنهاد شده‌اند. مکانیسم دوم از یک تکنیک جدید استفاده می‌کند که بهترین راه حل به دست آمده را حفظ می‌کند. علاوه بر این، یک عملگر جهش برای بهبود مرحله اکتشاف در الگوریتم BGOA (BGOA-M) استفاده می‌شود. روش‌های پیشنهادی با استفاده از ۲۵ مجموعه داده استاندارد UCI ارزیابی شده و با ۸ رویکرد فراابتکاری مقایسه شده است. نتایج مقایسه‌ای عملکرد برتر روش‌های BGOA و BGOA-M را در مقایسه با سایر تکنیک‌های مشابه نشان داده‌اند.

الگوریتم سیستم ایمنی مصنوعی که از الگوریتم‌های مبتنی بر جمعیت است برای تشخیص ایمیل هرزنامه بر روی مجموعه‌داده TREC07 که شامل ۷۵۴۱۹ ایمیل است تست و اجرا شده است. الگوریتم سیستم ایمنی مصنوعی برای انتخاب ویژگی بر مبنای تکرار و تست استفاده شده است. نتایج نشان داده که مدل توانسته به‌دقت بالای ۸۰ درصد دست یابد [۳۰]. مدل RAN-LSH که ترکیبی از شبکه تابع شعاعی پایه و ماشین بردار پشتیبان است برای تشخیص ایمیل هرزنامه پیشنهاد شده است. ارزیابی بر روی دو مجموعه داده که در سال ۲۰۱۳ گردآوری شده‌اند و هر کدام شامل ۸۹۴۸ (۱۳۱۱ ایمیل هرزنامه و ۷۷۳۷ ایمیل غیرهرزنامه) و ۲۰۹۰۵ (۸۸۶۳ ایمیل و ۱۲۰۴ ایمیل غیر هرزنامه) نمونه ایمیل هرزنامه هستند انجام شده است. نتایج نشان داد که مدل RAN-LSH در پیش‌بینی دقت مناسبی دارد. به دلیل اینکه مجموعه داده‌ها شامل ویژگی‌های زیادی هستند از انتخاب ویژگی استفاده شده است. در مدل RAN-LSH برای انتخاب ویژگی‌ها از ماشین بردار پشتیبان و برای آموزش و تست از تابع شعاعی پایه استفاده شده است. همچنین برای انتخاب ویژگی‌های برازنده از لایه وسطی مدل RAN-LSH استفاده می‌شود [۳۱].

برای تشخیص ایمیل‌های هرزنامه، یک مدل ترکیبی بر مبنای الگوریتم بهینه‌سازی نهنگ و الگوریتم گرده‌افشانی گل برای مسئله انتخاب ویژگی بر اساس مفهوم یادگیری مبتنی بر مخالفت با نام HWOAFPA ارائه شده است. در روش پیشنهادی، با استفاده از فرآیندهای الگوریتم بهینه‌سازی نهنگ و الگوریتم گرده‌افشانی گل سعی شده است مشکل انتخاب ویژگی حل شود و از طرف دیگر برای اطمینان از میزان همگرایی و دقت الگوریتم پیشنهادی از روش یادگیری مبتنی بر مخالفت استفاده شده است. در واقع، در روش پیشنهادی، الگوریتم بهینه‌سازی نهنگ با استفاده از فرآیند محاصره و محاصره طعمه، تهاجم حباب و روش‌های جستجوی طعمه، راه‌حل‌هایی را در فضای جستجوی خود ایجاد می‌کند وسعی می‌کند مشکل انتخاب ویژگی را بهبود بخشد [۳۲].

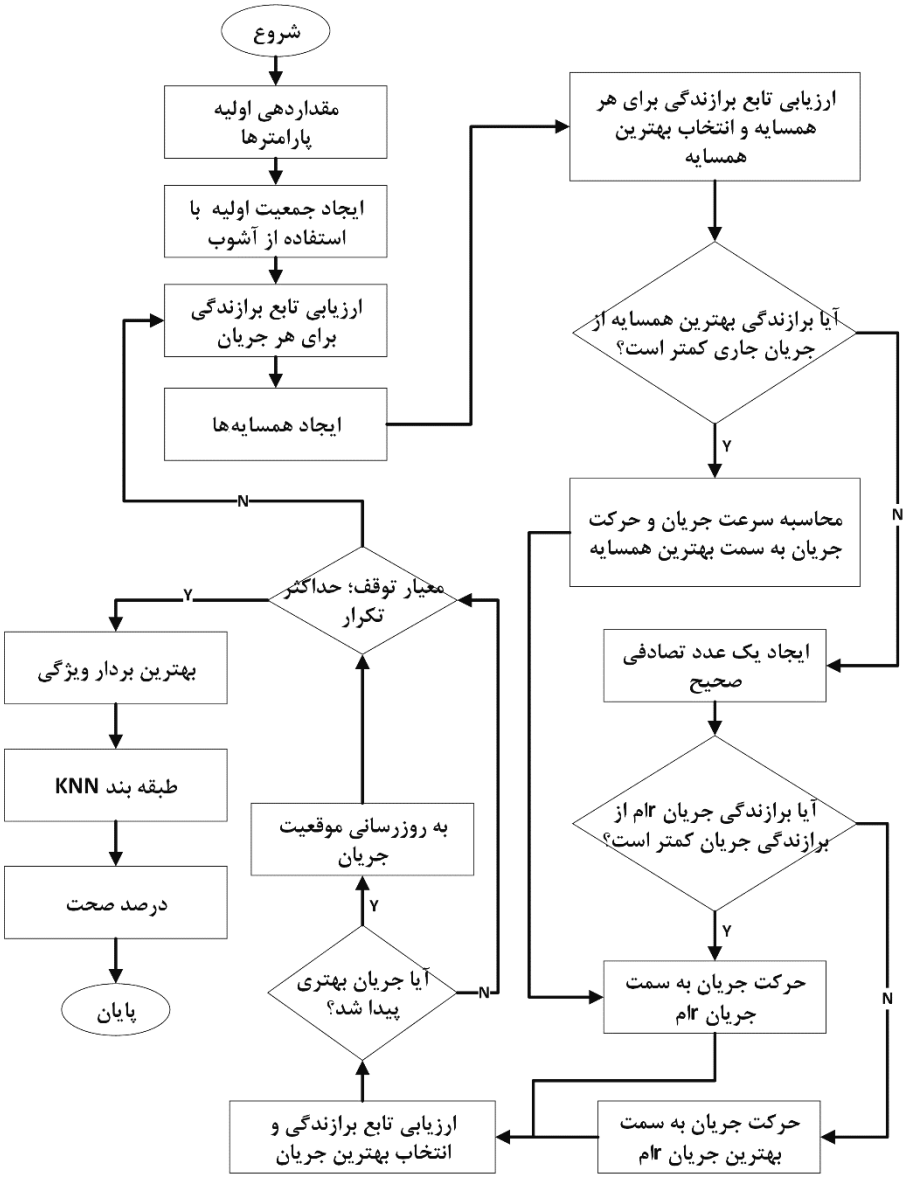
روش درخت رگرسیون تجمیعی بیزین که بر مبنای رگرسیون کار می‌کند بر روی Ling-spam با ۲۸۹۳ نمونه ، PU1 با ۱۰۹۹ نمونه و Spambase با ۴۶۰۱ نمونه تست و اجرا شده است. همچنین از مدل‌های طبقه‌بندی به منظور تخمین دقت استفاده شده است. دقت مدل درخت رگرسیون تجمیعی بیزین برای سه مجموعه داده به ترتیب برابر ۱۰۰٪ ، ۱۰۰ و ۹۷/۶۰ درصد بوده است[۳۳]. طبقه‌بندی بیزین برای تشخیص ایمیل‌های هرزنامه روی سه مجموعه داده که شامل ۱۰۰۰ ، ۱۵۰۰ و ۲۰۰۰ ایمیل هستند اجرا و تست شده است. در طبقه‌بندی بیزین از روابط احتمالی استفاده می‌شود. نتایج نشان داده دقت تشخیص برای سه مدل بیشتر از ۹۳ درصد بوده است. برای ارزیابی از معیارهای صحت، دقت و بازخوانی استفاده شده است. درصد صحت به ترتیب برای سه مجموعه داده برابر ۹۳/۹۸، ۹۴/۸۵ و ۹۶/۴۶ بوده است. الگوریتم سیستم ایمنی مصنوعی به منظور تشخیص ایمیل هرزنامه بر روی ۴۶۰۱ نمونه تست و اجرا شده است[۳۴].

یک روش جدید مبتنی بر الگوریتم بهینه‌سازی شیر مورچه با عنوان ALO-Boosting برای حل مشکل ایمیل‌های هرزنامه ارائه شده است [۳۵]. الگوریتم بهینه‌سازی شیر مورچه یک مدل محاسباتی است که از ویژگی‌های فنی شکار شیر مورچه‌ها به مورچه‌ها در چرخه زندگی تقلید می‌کند. زیر مجموعه ویژگی‌های بهینه برای ارائه طبقه‌بندی بهتر براساس الگوریتم بهینه‌سازی شیر مورچه به دست آمده است. روش جدید با ماشین بردار پشتیبان، الگوریتم k-نزدیکترین همسایه و مدل Bagging برروی مجموعه داده‌های ایمیل هرزنامه مقایسه شده است. نتایج تجربی نشان داد که روش جدید با تعداد ویژگی‌های انتخابی شامل دقت بالا بوده است. یک نسخه ترکیبی مبتنی بر الگوریتم جستجوی

فلامینگوی دودویی با الگوریتم ژنتیک (HBFS-GA) برای حل مسئله نتخاب ویژگی پیشنهاد شده است [۳۶]. در مدل HBFS-GA کارایی الگوریتم جستجوی فلامینگوی برای مرحله ادغام در لگوریتم ژنتیک استفاده شده است. مدل HBFS-GA با استفاده از توابع پیوسته CEC'2019 و CEC'2020 ارزیابی شده است. مدل HBFS-GA نتایج بهتری نسبت به مدل‌های دیگر به دست آورده است.

۳- مدل پیشنهادی

مدل پیشنهادی، ترکیبی از الگوریتم جهت جریان با نظریه آشوب است. هدف از این بهبود افزایش دقت طبقه‌بندی و کاهش تعداد ویژگی‌های انتخاب شده است. الگوریتم بهینه‌سازی جهت جریان در هنگام استفاده برای حل مسائل با ابعاد بالا، مانند انتخاب ویژگی، دو مشکل قابل‌توجه دارد. این اشکالات شامل تنوع راه‌حل‌های اولیه و مشکلات بهینه محلی هستند. بنابراین، نگاشت آشوبناک برای بهبود الگوریتم جهت جریان پیشنهاد می‌شود. در این مقاله، از دو نگاشت آشوبناک پرکاربرد برای بهینه‌سازی الگوریتم جهت جریان استفاده شده است. نگاشت آشوب دایره‌ای و لجستیک در مرحله اولیه‌سازی برای بهبود تنوع راه‌حل‌های الگوریتم جهت جریان استفاده می‌شود. در شکل (۱) فلوچارت روش پیشنهادی نشان‌داده‌شده است.



شکل (۱): فلوچارت روش پیشنهادی برای انتخاب ویژگی

در شکل (۲) شبه کد مدل پیشنهادی برای انتخاب ویژگی نشان داده شده است.

Read dataset (Spambase)

```
1 .Define FDA with parameters (maxiter, lb, ub, dim, fobj, alpha, beta)
    alpha=Number of flows
    beta= Number of neighborhoods
2 .Primary population production
3 .While (Iter<Max_Iteration)
4 .Boosting primary population production by chaos
5 .Calculate the initial fitness of each flow.
6 .Sort and select the best results, setting Best_fitness and BestX.
7 .Initialize velocity parameters Vmax and Vmin.
8 .For each iteration:
8.1 .Update weight W: Update weight W based on iteration count and random factors
8.2 .For each flow (i from 1 to alpha):
8.2.1 .Produce and evaluate neighborhoods.
8.2.2 .For each neighborhood position (j from 1 to beta)
8.2.3 .Generate random position within bounds
8.2.4 .Calculate new position based on delta and W
8.2.5 .Apply boundary conditions
8.2.6 .Calculate the fitness of the neighborhood position
8.2.7 .Sort and update flow position if the better neighborhood is found.
8.2.8 .Move flow towards better positions or global best.
8.2.9 .Apply boundary conditions and update fitness.
8.2.10 .Update the best flow if a better solution is found.
9 .Display current best fitness.
10 .The best feature vector
11 .End while
12 .Classification: KNN
13 .Evaluation criteria: Precision, Recall, F-Measure, Accuracy
14. End
```

شکل (۲): شبه کد مدل پیشنهادی برای انتخاب ویژگی

$$X_{ij} = LB_j + x_{ij}^k \times (UB_j - LB_j) \qquad (۱)$$

۳-۱- جمعیت اولیه

اولین گام در یک الگوریتم مبتنی بر جمعیت، تعریف جمعیت اولیه است که در آن یک گروه از N عامل جستجو به طور تصادفی تولید می‌شوند. هر راه‌حل کاندید حاوی کران‌های پایین و بالایی در محدوده [۱،۱-] است. هر عامل جستجو بیانگر یک راه حل بالقوه با D بعد است که در مسئله انتخاب ویژگی بیانگر تعداد ویژگی‌های اصلی در مجموعه داده است. فضای جستجو به صورت D×N تعریف می‌شود. موقعیت عامل‌ها در فضا به صورت $X_n = [X_{n1}, X_{n2}, X_{n3}, \cdots, X_{nD}]^{\mathrm{T}}$ تعریف می‌شود. مسئله انتخاب ویژگی برای اهداف طبقه‌بندی شامل یک زیر مجموعه ویژگی مرتبط است که هدفشان به حداکثر رساندن دقت طبقه‌بندی است. در مدل پیشنهادی از نگاشت دایره‌ای و نگاشت لجستیک برای مقداردهی اولیه جمعیت برای دستیابی به شتاب در مراحل اولیه الگوریتم جهت جریان استفاده شده است. مقدار x_{ij}^k طبق تعریف نگاشت دایره‌ای و نگاشت لجستیک تولید می‌شود.

که $x_i \in X$ است به‌طوریکه $i = 1,2,\dots,N; j = 1,2,\dots,D$ است. LB_j و UB_j بترتیب کران پایین($lb = [lb_1, lb_2, lb_3, \cdots, lb_j, \cdots, lb_D], j = 1,2,3,\cdots,D$) و کران بالا($ub = [ub_1, ub_2, ub_3, \cdots, ub_j, \cdots, ub_D]$) را نشان می‌دهند. در الگوریتم جهت جریان مقداردهی اولیه جمعیت به صورت تصادفی است. بنابراین، عملکرد آن ناپایدار است. اگر موقعیت جمعیت اولیه تولید شده نزدیک به راه حل بهینه سراسری باشد، آنگاه سرعت همگرایی الگوریتم و توانایی آن برای به دست آوردن جواب بهینه سراسری خوب خواهد بود. اما موقعیت‌های راه حل اولیه که به طور تصادفی ایجاد می‌شوند به ندرت ایده‌آل هستند. بنابراین، روش پیشنهادی با استفاده از آشوب می‌تواند موقعیت قابل اطمینان‌تری فراهم کند، به طوری که اطمینان حاصل شود که سرعت همگرایی الگوریتم نوسان زیادی ندارد و عملکرد را بهبود می‌دهد.

۳-۲- عملگرهای آشوب

در الگوریتم جهت جریان، مقدار آشوب جایگزین مقادیر تصادفی از موقعیت عامل‌ها در مرحله اولیه می‌شود. آشوب به طور قابل‌توجهی سرعت همگرایی و عملکرد برازش الگوریتم جهت جریان را افزایش می‌دهد. در ادامه آشوب دایره‌ای و نگاشت لجستیک توضیح داده می‌شود.

آشوب دایره‌ای: این آشوب ابتدا توسط Andrey Colmogorov و همکارانش تعریف شد [۳۷]. این عملگر تک بعدی است که x_{k+1} آن از روی x_k به صورت معادله (۲) تعریف می‌شود. مقادیر P و b به ترتیب روی ۰,۵ و ۰,۲ تنظیم می‌شوند تا اعداد آشوبناک بین (۰،۱) ایجاد شود.

$$x_{k+1} = x_k + b - \left(\frac{P}{2\pi}\right)\sin(2\pi x_k) mod(1) \qquad (۲)$$

نگاشت لجستیک: این نگاشت به‌صورت معادله (۳) تعریف می‌شود. که P به منظور تولید اعداد در محدوده (۰،۱) روی ۴ تنظیم شده است.

$$x_{k+1} = P.x_k(1 - x_k) \qquad (۳)$$

انگیزه اصلی رویکردهای ترکیبی، تعادل بین رفتار اکتشاف و بهره‌برداری و ایجاد یک الگوریتم بهینه‌سازی سراسری است. اگرچه الگوریتم جهت جریان عملکرد قابل‌توجهی در مسائل طراحی پیچیده داشته است، در یک بهینه محلی گرفتار شود. این محدودیت توسط ماهیت پویای نگاشت‌های آشوب بهبود می‌یابد، زیرا آنها با پرش به خارج از منطقه محلی به فرآیند جستجو کمک می‌کنند.

۳-۳- الگوریتم جهت جریان

عملکرد الگوریتم جهت جریان به موقعیت جاری، بهترین موقعیت و موقعیت همسایه‌ها بستگی دارد. اطراف هر جریان (راه‌حل)، همسایه‌های β وجود دارد که موقعیت آنها طبق معادله (۴) تعریف می‌شود.

$$Neighbor\_X(j) = Flow\_X(i) + randn \times \Delta \qquad (۴)$$

که $Flow_{X(i)}$ موقعیت جریان iام، $Neighbor\_X$ موقعیت همسایه jام و $randn$ یک مقدار تصادفی با توزیع نرمال، میانگین صفر و انحراف استاندارد یک است. مقادیر کوچک تولید شده توسط Δ منجر به جستجو در محدوده کوچک و مقادیر بزرگ تولید شده منجر به جستجو در محدوده بزرگ می‌شوند. در واقع، جستجو در محدوده وسیع منجر به تنوع بیشتر راه حل‌ها می‌شود و احتمال یافتن راه حل‌های تقریباً بهینه (جستجوی سراسری) را افزایش می‌دهد. هنگامی که راه‌حل‌های الگوریتم به راه حل بهینه سراسری نزدیک می‌شوند، جستجو در یک محدوده کوچک منجر به یافتن راه حل بهینه سراسری با دقت بالاتر (جستجوی محلی) می‌شود. از طرفی دیگر، انجام جستجوی محلی باعث می‌شود که الگوریتم در بهینه محلی به دام بیفتد. از این رو، به منظور ایجاد تعادل بین جستجوی سراسری و محلی، از معادله (۵) با کاهش خطی Δ از یک مقدار بزرگ به یک مقدار کوچک استفاده می‌شود. جهت Δ برای تنوع بیشتر به سمت موقعیت تصادفی است.

$$\Delta = (rand \times Xrand - rand \times Flow\_X(i)) \times \|Best\_X - Flow\_X(i)\| \times W \qquad (۵)$$

که rand یک عدد تصادفی با توزیع یکنواخت است، Xrand یک موقعیت تصادفی در فضای جستجو است، W وزن غیرخطی با عدد تصادفی بین ۰ و inf است. طرف اول نشان می‌دهد که Flow_X(i) با موقعیت تصادفی (Xrand) حرکت می‌کند. عبارت دوم با افزایش تکرار، Flow_X(i) به $Best\_X$ نزدیک می‌شود و فاصله اقلیدسی بین $Best\_X$ و $Flow\_X(i)$ به صفر کاهش می‌یابد. بنابراین، جستجوی محلی قطع می‌شود. W طبق معادله (۶) محاسبه می‌شود [۲۰].

$$W = \left(1 - \frac{iter}{Max_{iter}}\right)^{(2\times randn)} \times \left(\frac{rand}{\overline{rand} \times Max\_Iter}\right) \qquad (۶)$$

که $\overline{rand}$ بردار تصادفی با توزیع یکنواخت است. W در طول دوره تکرار دارای تنوع زیادی است. این موضوع فرار از بهینه محلی در الگوریتم جهت جریان را تضمین می‌کند. جریان با سرعت V به سمت همسایه‌های بهینه حرکت می‌کند. از سوی دیگر، سرعت جریان به همسایگان رابطه مستقیمی با شیب دارد. معادله (۷) برای تعیین بردار سرعت جریان استفاده می‌شود.

$$V = randn \times S_0 \qquad (۷)$$

که S_0 بردار شیب بین همسایه و موقعیت فعلی جریان را نشان می‌دهد. عدد تصادفی randn راه‌حل‌های مختلفی تولید می‌کند و جستجوی سراسری را افزایش می‌دهد. بردار شیب جریان i ام نسبت به همسایه j ام طبق معادله (۸) تعیین می‌شود.

$$S_0(i,j,d) = \frac{Flow\_fitness(i) - Neighbor\_fitness(j)}{\|Flow\_x(i,d) - Neighbor\_x(j,d)\|} \qquad (۸)$$

که Flow_fitness (i) و Neighbor_fitness (j) به ترتیب مقدار برازندگی جریان i و همسایه j را نشان می‌دهند. پارامتر d

ابعاد مسئله را نشان می‌دهد. معادله (۹) برای تعیین موقعیت جدید جریان استفاده می‌شود.

$$Flow\_newX(i) = flow\_X(i) + v \times \frac{Flow\_X(i) - Neighbor\_X(j)}{\|Flow\_x(i) - Neighbor\_x(j)\|} \quad (۹)$$

که Flow_newX(i) موقعیت جدید جریان i را نشان می‌دهد. قانون به‌روزرسانی موقعیت با موقعیت‌های همسایه‌ها و موقعیت جاری تعیین می‌شود.

۳-۴- انتخاب ویژگی

یک مجموعه‌داده شامل نمونه‌ها (سطرها)، ویژگی‌ها (ستون‌ها) و کلاس‌ها است. در یک مجموعه‌داده، برخی از ویژگی‌ها بدون استفاده، نامربوط یا اضافی هستند که بر عملکرد طبقه‌بند تأثیر منفی می‌گذارند. بنابراین، برای افزایش عملکرد طبقه‌بند و کاهش اندازه یک مجموعه‌داده، مسئله انتخاب ویژگی لازم است. برای شناسایی ویژگی‌های برجسته از مقادیر ۱ و حذف ویژگی‌ها از مقادیر ۰ استفاده می‌شود. موقعیت اولیه هر عامل در فضای جستجو بایستی به حالت دودویی تبدیل شود. در نسخه دودویی، راه‌حل پیوسته X_i^{t+1} توسط تابع انتقال S شکل در محدوده [۰،۱] به فضای دودویی تبدیل می‌شود. بنابراین، راه حل‌های روش پیشنهادی در یک فضای جدید بین ۰ و ۱ نگاشت می‌شوند. سپس از آستانه تصادفی برای تولید یک راه حل باینری جدید استفاده می‌شود که در قالب معادله (۱۰) تعریف می‌شود.

$$S(X_i^{t+1}) = \frac{1}{1+e^{\left(-\frac{X_i^{t+1}}{2}\right)}} \quad (۱۰)$$

$$X_i^{t+1} = \begin{cases} 0 & if\ r_1 < S(X_i^{t+1}) \\ 1 & if\ r_1 \geq S(X_i^{t+1}) \end{cases} \quad (۱۱)$$

که r_1 عددی در محدوده [۰،۱] است که مقدار نهایی هر یک از ابعاد X_i^{t+1} را تعیین می‌کند و در نتیجه یک راه‌حل دودویی جدید ایجاد می‌کند. علاوه بر این، X_i^{t+1} موقعیت راه حل iام در طول تکرار tام از روش پیشنهادی را نشان می‌دهد.

۳-۵- تابع برازندگی

هدف تابع برازندگی به‌حداقل‌رساندن تعداد ویژگی‌های انتخاب شده و حداکثرکردن دقت طبقه‌بندی است. عمدتاً هدف انتخاب ویژگی، دستیابی به‌دقت طبقه‌بندی بالا با کمترین زیر مجموعه ویژگی‌ها است. تابع برازندگی طبق معادله (۱۲) تعریف شده است. تابع برازندگی پیشنهادی برای محاسبه دقت طبقه‌بندی هر راه‌حل و همچنین تعداد ویژگی‌های انتخاب شده استفاده می‌شود. در هر تکرار، تابع برازندگی به دنبال کاوش و بهره‌برداری از بهترین موقعیت فعلی محاسبه می‌شود. فرض بر این است که ارزش برازندگی موقعیت جدید (راه حل) بهتر از موقعیت جاری است. در نتیجه بهترین راه حل یا راه حل بهبودیافته جایگزین می‌شود و جستجوی محلی انجام می‌شود. این روند تا زمانی که معیارهای توقف انجام شود تکرار می‌شود. تابع برازندگی پیشنهادی طبق معادله (۱۲) تعریف شده است.

$$FF = \alpha y_R(D) + \beta \frac{R}{N} \quad (۱۲)$$

که $\alpha y_R(D)$ توسط نرخ خطای طبقه‌بندی k نزدیکترین همسایه بدست می‌اید. علاوه بر این، R تعدادی از زیر ویژگی‌های انتخاب شده است، و N تعداد کل ویژگی‌های مجموعه داده است، و دو پارامتر α و β مربوط به اهمیت کیفیت طبقه‌بندی و طول زیر مجموعه هستند و $\alpha \in [0,1]$ و $\beta = (1-\alpha)$ است.

۳-۶- الگوریتم k نزدیکترین همسایه

نقش طبقه‌بند k نزدیک‌ترین همسایه این است که هر نمونه داده را به یک کلاس اختصاص دهد به‌طوری‌که به نزدیک‌ترین همسایگان k تعلق داشته باشد. برای مجموعه‌داده آزمایشی توسط الگوریتم k نزدیک‌ترین همسایه باید K نزدیک‌ترین همسایگان برای هر نمونه از مجموعه‌داده آموزشی طبق محاسبه اقلیدسی به‌صورت معادله (۱۳) تعیین شود که d تعداد ویژگی‌های یک مجموعه‌داده است. x و y به ترتیب نمونه داده آموزشی و آزمایشی هستند.

$$D(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \quad (۱۳)$$

۳-۷- معیارهای ارزیابی

نتایج مدل پیشنهادی، در مرحله ارزیابی توسط معیارهای معیار Precision ، Recall، F-measure و Accuracy مورد تحلیل قرار گرفته است. این معیارها را می‌توان هم برای مجموعه‌داده‌های آموزشی در مرحله یادگیری و هم برای مجموعه رکوردهای آزمایشی در مرحله ارزیابی محاسبه نمود. در معیارهای ذکر شده، accuracy مهم‌ترین معیار در دقت تشخیص است.

$$Precision = \frac{TP}{TP + FP} \quad (۱۴)$$

$$Recall = \frac{TP}{TP + FN} \quad (۱۵)$$

$$F - Measure = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \quad (۱۶)$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (۱۷)$$

پارامتر TP (مثبت درست) نشان‌دهنده تعداد نمونه‌هایی که جز دسته مثبت بوده و درست پیش‌بینی شده‌اند. پارامتر FP (کاذب مثبت) نشان‌دهنده تعداد نمونه‌هایی که نادرست به‌عنوان

دسته مثبت پیش‌بینی شده‌اند. پارامتر FN (کاذب منفی) نشان‌دهنده تعداد نمونه‌هایی که نادرست به‌عنوان دسته منفی پیش‌بینی شده‌اند. پارامتر TN (منفی درست) نشان‌دهنده تعداد نمونه‌هایی که جز دسته منفی بوده و درست پیش‌بینی شده‌اند.

۴- ارزیابی و نتایج

ارزیابی و نتایج بر روی مجموعه‌داده Spambase (https://archive.ics.uci.edu/ml/datasets/spambase) برگرفته از سایت UCI با ۴۶۰۱ نمونه و ۵۷ ویژگی در محیط سی‌شارپ دانت ۲۰۲۰ انجام شده است. به دلیل آنکه الگوریتم‌های فراابتکاری از جستجوی تصادفی پیروی می‌کنند، برنامه به طور میانگین ۱۵ بار اجرا شده است. برای ارزیابی مدل پیشنهادی، مقدار پارامترهایی مانند جمعیت اولیه و تعداد تکرار به ترتیب برابر ۳۰ و ۳۰۰ است و همچنین مرحله آموزش و تست به ترتیب برابر با ۸۰ و ۲۰ درصد هستند.

۴-۱- ارزیابی برمبنای تعداد تکرار

در جدول (۱) نتایج مدل پیشنهادی برمبنای تعداد تکرارهای مختلف نشان داده شده است. درصد صحت با ۱۵۰ تکرار برای الگوریتم کرم شب‌تاب (۹۲/۸۶)، الگوریتم جستجوی کلاغ(۹۳/۲۳)، روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۳/۷۹) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۴/۴۲) بدست آمده است. درصد صحت با ۳۰۰ تکرار برای الگوریتم کرم شب‌تاب (۹۳/۹۴)، الگوریتم جستجوی کلاغ (۹۴/۶۶)، روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۵/۴۱) و روش پیشنهادی مبتنی بر نگاشت لجستیک(۹۶/۲۸) بدست آمده است.

جدول (۱): نتایج مدل‌ها برمبنای تعداد تکرار

تعداد تکرار	مدل	Precision	Recall	F-Measure	Accuracy
۱۵۰	الگوریتم کرم شب‌تاب	۹۲/۵۲	۹۲/۶۴	۹۲/۵۸	۹۲/۸۶
	الگوریتم جستجوی کلاغ	۹۲/۷۸	۹۲/۸۲	۹۲/۸۰	۹۳/۲۳
	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۳/۲۱	۹۳/۳۵	۹۳/۲۸	۹۳/۷۹
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۴/۱۷	۹۴/۲۹	۹۴/۲۳	۹۴/۴۲
۳۰۰	الگوریتم کرم شب‌تاب	۹۲/۹۶	۹۳/۲۵	۹۳/۱۰	۹۳/۹۴
	الگوریتم جستجوی کلاغ	۹۳/۳۷	۹۳/۶۲	۹۳/۴۹	۹۴/۶۶
	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۴/۸۹	۹۵/۰۸	۹۴/۹۸	۹۵/۴۱
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۰۷	۹۵/۵۷	۹۵/۳۲	۹۶/۲۸

۴-۲- ازریابی برمبنای تعداد اجرا

در جدول (۲) نتایج مدل پیشنهادی برمبنای تعداد اجراهای مختلف نشان داده شده است. در اجرای اول، درصد صحت روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۵/۶۶) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۵/۸۶) بدست آمده است. در اجرای سوم، درصد صحت روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۵/۷۳) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۵/۹۶) بدست آمده است. در اجرای پنجم، درصد صحت روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۵/۸۴) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۶/۰۵) بدست آمده است. این نتایج نشان می‌دهد که روش پیشنهادی با هر اجرا دارای نتایج متفاوتی است. چون روش پیشنهادی مبتنی بر الگوریتم‌های فراابتکاری است و به صورت تصادفی عمل می‌کند.

جدول (۲): نتایج روش پیشنهادی برمبنای تعداد اجراهای مختلف

تعداد اجرا	مدل	Precision	Recall	F-Measure	Accuracy
۱	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۴/۵۲	۹۵/۶۳	۹۵/۰۷	۹۵/۶۶
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۲۱	۹۵/۷۴	۹۵/۴۷	۹۵/۸۶
۲	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۴/۹۶	۹۵/۱۶	۹۵/۰۶	۹۵/۴۵
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۲۷	۹۵/۴۲	۹۵/۳۴	۹۵/۷۲
۳	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۴/۸۲	۹۵/۹۵	۹۵/۳۸	۹۵/۷۳
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۱۶	۹۵/۳۳	۹۵/۲۴	۹۵/۹۴
۴	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۴/۹۲	۹۵/۱۹	۹۵/۰۵	۹۵/۳۲
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۱۶	۹۵/۴۸	۹۵/۳۲	۹۵/۵۶
۵	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۵/۱۲	۹۵/۲۵	۹۵/۱۸	۹۵/۸۴
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۲۰	۹۵/۶۹	۹۵/۴۴	۹۶/۰۵

۴-۳- ارزیابی برمبنای تعداد ویژگی

در جدول (۳) نتایج روش پیشنهادی برمبنای تعداد ویژگی‌ها نشان داده شده است. طبق نتایج بدست آمده مشخص است که درصد صحت با ۸ ویژگی برای روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۸/۲۸) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۸/۴۵) است. درصد صحت با ۲۲ ویژگی برای روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۷/۷۱) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۸/۱۴) است. درصد صحت با ۴۲ ویژگی برای روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۶/۷۹) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۷/۲۵) است. درصد صحت با ۵۲ ویژگی برای روش پیشنهادی مبتنی بر نگاشت دایره‌ای (۹۶/۱۴) و روش پیشنهادی مبتنی بر نگاشت لجستیک (۹۶/۴۶) است. نتایج نشان می‌دهد که درصد صحت روش پیشنهادی مبتنی بر نگاشت لجستیک بیشتر است.

جدول (۳): نتایج روش پیشنهادی برمبنای تعداد ویژگی‌ها

تعداد ویژگی	مدل	Precision	Recall	F-Measure	Accuracy
۸	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۸/۰۵	۹۸/۱۶	۹۸/۱۰	۹۸/۲۸
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۸/۳۲	۹۸/۶۵	۹۸/۴۸	۹۸/۴۵
۱۴	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۷/۸۶	۹۷/۴۲	۹۷/۶۴	۹۷/۹۵
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۸/۱۵	۹۸/۲۶	۹۸/۲۰	۹۸/۳۲
۲۲	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۷/۵۳	۹۷/۳۹	۹۷/۴۶	۹۷/۷۱
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۷/۹۴	۹۸/۰۲	۹۷/۹۸	۹۸/۱۴
۲۸	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۷/۲۰	۹۷/۳۸	۹۷/۲۹	۹۷/۴۸
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۷/۸۳	۹۷/۶۴	۹۷/۷۳	۹۷/۹۶
۳۲	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۷/۷۵	۹۷/۸۸	۹۷/۸۱	۹۷/۱۸
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۷/۶۲	۹۷/۹۳	۹۷/۷۷	۹۷/۷۷
۳۶	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۶/۶۵	۹۶/۴۳	۹۶/۵۴	۹۶/۹۸
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۷/۲۵	۹۷/۱۲	۹۷/۱۸	۹۷/۴۲
۴۲	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۶/۵۷	۹۶/۶۸	۹۶/۶۲	۹۶/۷۹
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۷/۱۴	۹۷/۲۳	۹۷/۱۸	۹۷/۲۵
۴۸	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۶/۳۴	۹۶/۴۶	۹۶/۴۰	۹۶/۵۲
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۶/۶۳	۹۶/۷۴	۹۶/۶۸	۹۶/۸۱
۵۲	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۵/۸۶	۹۵/۹۷	۹۵/۹۱	۹۶/۱۴
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۶/۲۴	۹۶/۴۰	۹۶/۳۲	۹۶/۴۶
۵۷	روش پیشنهادی مبتنی بر نگاشت دایره‌ای	۹۵/۶۲	۹۵/۷۶	۹۵/۶۹	۹۵/۸۴
	روش پیشنهادی مبتنی بر نگاشت لجستیک	۹۵/۹۴	۹۶/۰۶	۹۶/۰۰	۹۶/۲۸

۴-۴- مقایسه و ارزیابی

در جدول (۴) مقایسه روش پیشنهادی با روش‌های دیگر نشان داده شده است. درصد صحت الگوریتم بهینه‌سازی گله اسب با k نزدیکترین همسایه (۹۴/۵۱) است. در حالیکه درصد صحت روش پیشنهادی (۹۶/۲۸) است. درصد صحت شبکه عصبی مصنوعی با الگوریتم سینوس کسینوس (۹۷/۹۲)، جنگل چرخشی با الگوریتم بهینه‌سازی نهنگ (۹۹/۸۹) و الگوریتم بهینه‌سازی جستجوی عنکبوت یادگیری مبتنی بر مخالفت با k نزدیکترین همسایه (۹۶/۹۷) بدست آمده است.

جدول (۴): مقایسه روش پیشنهادی با روش‌های دیگر

مرجع	مدل	توضیحات	درصد صحت
[۳۸]	HHOBSA	الگوریتم بهینه‌سازی شاهین هریس با تبرید شبیه‌سازی شده	۹۴/۴۰
[۳۹]	ANN	شبکه عصبی مصنوعی	۹۴
	J48	درخت تصمیم‌گیری	۹۷/۱۷
	Random Forset	جنگل تصادفی	۹۹/۹۳
[۴۰]	ANN-SCA	شبکه عصبی مصنوعی با الگوریتم سینوس کسینوس	۹۷/۹۲
[۴۱]	Rota tion Forest	جنگل چرخشی	۹۴/۲۰
	WOA-Rota tion Forest	جنگل چرخشی با الگوریتم بهینه‌سازی نهنگ	۹۹/۸۹

[۴۲]	BBH	الگوریتم سیاه‌چاله دودویی	۹۱/۸۵
[۴۳]	OBSSO	الگوریتم بهینه‌سازی جستجوی عنکبوت یادگیری مبتنی بر مخالفت با جنگل تصادفی	۹۶/۴۸
		الگوریتم بهینه‌سازی جستجوی عنکبوت یادگیری مبتنی بر مخالفت با k نزدیک‌ترین همسایه	۹۶/۹۷
	ABC	الگوریتم کلونی زنبور مصنوعی با جنگل تصادفی	۹۱/۱۰
		الگوریتم کلونی زنبور مصنوعی با k نزدیک‌ترین همسایه	۸۵/۸۸
	FA	الگوریتم کرم شب‌تاب با جنگل تصادفی	۹۳/۱۵
		الگوریتم کرم شب‌تاب با k نزدیک‌ترین همسایه	۹۰/۳۳
	روش پیشنهادی	روش پیشنهادی	۹۶/۲۸

درصد صحت الگوریتم کلونی زنبور مصنوعی با جنگل تصادفی(۹۱/۱۰)، درصد صحت الگوریتم کلونی زنبور مصنوعی با k نزدیک‌ترین همسایه(۸۵/۸۸)، درصد صحت الگوریتم کرم شب‌تاب با جنگل تصادفی(۹۳/۱۵)، درصد صحت الگوریتم کرم شب‌تاب با k نزدیک‌ترین همسایه(۹۰/۳۳) به‌دست‌آمده است. نتایج نشان داد که روش پیشنهادی در مقایسه با اغلب مدل‌های دیگر درصد صحت بیشتری دارد. هر مدل مزایا و معایبی دارد، به‌عنوان‌مثال برخی از الگوریتم‌ها به نتایج بهینه دست یافته‌اند؛ اما پیچیدگی زمانی آنها افزایش‌یافته است.

۵- نتیجه‌گیری و کارهای آینده

افزایش حجم داده‌ها منجر به تأثیر منفی بر روی طبقه‌بندی‌های یادگیری ماشین می‌شود. در یادگیری ماشین، فرآیند انتخاب ویژگی برای انتخاب مرتبط‌ترین ویژگی‌ها و کاهش موارد اضافی و نامربوط حیاتی است. الگوریتم‌های فراابتکاری، توانایی خوبی برای حل مسائل انتخاب ویژگی دارند. الگوریتم جهت جریان برای مسائل انتخاب ویژگی با ابعاد بالا از تنوع جمعیت و محدودیت‌های بهینه محلی رنج می‌برد. در این مقاله، الگوریتم جهت جریان بهبودیافته مبتنی بر نگاشت آشوب برای غلبه بر محدودیت پیشنهاد شد. عملکرد روش پیشنهادی بر روی *Spambase* از مخزن یادگیری ماشین UCI ارزیابی شد. نتایج نشان داد که درصد صحت روش پیشنهادی با نگاشت لجستیک بیشتر است. درصد صحت روش پیشنهادی با نگاشت دایره‌ای ۹۵/۴۱ و درصد صحت روش پیشنهادی با نگاشت لجستیک ۹۶/۲۸ بدست آمده است. درصد صحت روش پیشنهادی با نگاشت لجستیک برای ۸ ویژگی و ۴۲ ویژگی بترتیب ۹۸/۴۵ و ۹۷/۲۵ بدست آمده است. روش این مقاله فقط برای انتخاب ویژگی برروی مسئله ایمیل هرزنامه اعمال شده است. لذا، در آینده می‌توان سایر مشکلات بهینه‌سازی ترکیبی باینری، از جمله طبقه‌بندی اسناد متنی و پیش‌بینی‌های مالی، تشخیص نفوذ و غیره را برطرف کرد. انتظار می‌رود کارهای آینده شامل بهبود الگوریتم بهینه‌سازی گوریل مصنوعی دودویی برای حل مسائل چندهدفه باشد.

۶- مراجع

[1]. E. Blanzieri, A. Bryl, "A survey of learning-based techniques of email spam filtering," Artificial Intelligence Review, vol. 29, no. 1, pp. 63-92, 2008. https://doi.org/10.1007/s10462-009-9109-6

[2]. N. N. Nicholas, V. Nirmalrani, "An enhanced mechanism for detection of spam emails by deep learning technique with bio-inspired algorithm," e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 8, no. 1, pp. 1-42, 2024. https://doi.org/10.1016/j.prime.2024.100504

[3]. S. O. Olatunji, "Improved email spam detection model based on support vector machines," Neural Computing and Applications, vol. 31, no. 3, pp. 691-699, 2019. https://doi.org/10.1007/s00521-017-3100-y

[4]. M. Rezvani, F. Bagheri, M. Fateh and E. Tahanian, "Malicious Domain Detection using DNS Records," Electronic and Cyber Defense, vol. 9, no. 3, pp. 83-97, 2021. (In Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1400.9.3.7.8

[5]. S. Qesmati, "Spam Management in Social Networks by Content Rating," Electronic and Cyber Defense, vol. 2, no. 2, pp. 53-62, 2014. (In Persian).https://dorl.net/dor/20.1001.1.23224347.1393.2.2.14.4

[6]. R. A. Zitar, A. Hamdan, "Genetic optimized artificial immune system in spam detection: a review and a model," Artificial Intelligence Review, vol. 40, no. 3, pp. 305-377, 2013. https://doi.org/10.1007/s10462-011-9285-z

[7]. N. Mahmoodi, H. Shirazi, M. Fakhredanesh and K. Dadashtabar Ahmadi, "Improving the performance of the convolutional neural network using incremental weight loss function to deal with class imbalanced data," Electronic and Cyber Defense, vol. 11, no. 4, pp. 19-34, 2024. (In Persian).https://dorl.net/dor/20.1001.1.23224347.1402.11.4.2.9

approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm," Advances in Engineering Software, vol. 176, no. 1, pp. 103402, 2023. https://doi.org/10.1016/j.advengsoft.2022.103402

[19]. S. Velliangiri, P. Karthikeyan, V. M. Arul Xavier and D. Baswaraj, "Hybrid electro search with genetic algorithm for task scheduling in cloud computing," Ain Shams Engineering Journal, vol. 12, no. 1, pp. 631-639, 2021. https://doi.org/10.1016/j.asej.2020.07.003

[20]. H. Karami, M. V. Anaraki, S. Farzin and S. Mirjalili, "Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems," Computers & Industrial Engineering, vol. 156, no. 2, pp. 1-38, 2021. https://doi.org/10.1016/j.cie.2021.107224

[21]. f. Soleimanian Gharehchopogh, M. Sakhidek Hovshin, "A New Model for Email Spam Detection using Hybrid of Magnetic Optimization Algorithm with Harmony Search Algorithm," Biannual Journal Monadi for Cyberspace Security (AFTA), vol. 9, no. 1, pp. 50-39, 2020. http://monadi.isc.org.ir/article-1-163-en.html

[22]. P. Wanda, "GRUSpam: robust e-mail spam detection using gated recurrent unit (GRU) algorithm," International Journal of Information Technology, vol. 15, no. 8, pp. 4315-4322, 2023. https://doi.org/10.1007/s41870-023-01516-z

[23]. A. Hosseinalipour, R. Ghanbarzadeh, "A novel approach for spam detection using horse herd optimization algorithm," Neural Computing and Applications, vol. 34, no. 15, pp. 13091-13105, 2022. https://doi.org/10.1007/s00521-022-07148-x

[24]. P. Dhal, C. Azad, "Hybrid momentum accelerated bat algorithm with GWO based optimization approach for spam classification," Multimedia Tools and Applications, vol. 83, no. 9, pp. 26929-26969, 2024. https://doi.org/10.1007/s11042-023-16448-w

[25]. A. Sharaff, H. Gupta (2019) Extra-tree classifier with metaheuristics approach for email classification, in *Advances in computer communication and computational sciences*, Springer. p. 189-197. https://doi.org/10.1007/978-981-13-6861-5_17

[26]. N. O. F. Elssied, O. Ibrahim, "K-means clustering scheme for enhanced spam detection," Research Journal of Applied Sciences, Engineering and Technology, vol. 7, no. 10, pp. 1940-1952, 2014. http://dx.doi.org/10.19026/rjaset.7.486

[27]. K.-C. Ying, S.-W. Lin, Z.-J. Lee and Y.-T. Lin, "An ensemble approach applied to classify spam e-mails," Expert Systems with Applications, vol. 37, no. 3, pp. 2197-2201, 2010. https://doi.org/10.1007/s10207-023-00756-1

[28]. H. He, A. Tiwari, J. Mehnen, T. Watson, C. Maple, Y. Jin and B. Gabrys. Incremental information gain analysis of input attribute impact on RBF-kernel SVM

[8]. M. khorram, M. Rahmanimanesh, "DDoS Attack Detection System Using Ensemble Method Classification and Active Learning Approach," Electronic and Cyber Defense, vol. 11, no. 3, pp. 101-118, 2023. (In Persian). https://dorl.net/dor/20.1001.1.23224347.1402.11.3.10.5

[9]. E. Bastami, H. Soltanizadeh, M. Rahmanimanesh and P. Keshavarzi, "A Malware Classification Method Using visualization and Word Embedding Features," Electronic and Cyber Defense, vol. 11, no. 1, pp. 1-13, 2023. (In Persian) https://dorl.net/dor/20.1001.1.23224347.1402.11.1.1.2

[10]. J. R. Vergara, P. A. Estévez, "A review of feature selection methods based on mutual information," Neural Computing and Applications, vol. 24, no. 1, pp. 175-186, 2014. https://doi.org/10.1007/s00521-013-1368-0

[11]. Z. M. Hira, D. F. Gillies, "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," Adv Bioinformatics, vol. 2015, no. pp. 198363, 2015. https://doi.org/10.1155/2015/198363

[12]. P. Hanchuan, L. Fuhui and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226-1238, 2005. https://doi.org/10.1109/TPAMI.2005.159

[13]. K. Kira, L. A. Rendell (1992) A Practical Approach to Feature Selection, in *Machine Learning Proceedings 1992*, Morgan Kaufmann: San Francisco (CA). p. 249-256. https://doi.org/10.1016/B978-1-55860-247-2.50037-1

[14]. F. S. Gharehchopogh, A. A. Khargoush *A Chaotic-Based Interactive Autodidactic School Algorithm for Data Clustering Problems and Its Application on COVID-19 Disease Detection*. Symmetry, 2023. volume, 1-20 DOI: 10.3390/sym15040894.

[15]. Y. Shen, C. Zhang, F. Soleimanian Gharehchopogh and S. Mirjalili, "An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems," Expert Systems with Applications, vol. 215, no. 1, pp. 119269, 2023. https://doi.org/10.1016/j.eswa.2022.119269

[16]. J. Yang, S. Gao, X. Zhao, G. Li and Z. Gao, "Enhanced Sparrow Search Algorithm Based on Improved Game Predatory Mechanism and its Application," Digital Signal Processing, vol. 2023, no. 1, pp. 104310, 2023. https://doi.org/10.1016/j.dsp.2023.104310

[17]. R. Ranjan, J. K. Chhabra, "Automatic clustering and feature selection using multi-objective crow search algorithm," Applied Soft Computing, vol. 142, no. 1, pp. 110305, 2023. https://doi.org/10.1016/j.asoc.2023.110305

[18]. M. Abd Elaziz, M. A. A. Al-qaness, A. Dahou, R. A. Ibrahim and A. A. A. El-Latif, "Intrusion detection

[36]. R. K. Eluri, N. Devarakonda, "Feature Selection with a Binary Flamingo Search Algorithm and a Genetic Algorithm," Multimedia Tools and Applications, vol. 82, no. 17, pp. 26679-26730, 2023. https://doi.org/10.1007/s11042-023-15467-x

[37]. H. Lu, X. Wang, Z. Fei and M. Qiu, "The Effects of Using Chaotic Map on Improving the Performance of Multiobjective Evolutionary Algorithms," Mathematical Problems in Engineering, vol. 2014, no. pp. 924652, 2014. https://doi.org/10.1155/2014/924652

[38]. M. Abdel-Basset, W. Ding and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," Artificial Intelligence Review, vol. 54, no. 1, pp. 593-637, 2021. https://doi.org/10.1007/s10462-020-09860-3

[39]. A. Ghosh, A. Senthilrajan, "Comparison of machine learning techniques for spam detection," Multimedia Tools and Applications, vol. 2023, no. 1, pp. 1-18, 2023. 10.1007/s11042-023-14689-3

[40]. R. Talaei Pashiri, Y. Rostami and M. Mahrami, "Spam detection through feature selection using artificial neural network and sine–cosine algorithm," Mathematical Sciences, vol. 14, no. 3, pp. 193-199, 2020. https://doi.org/10.1007/s40096-020-00327-8

[41]. M. Shuaib, S. i. M. Abdulhamid, O. S. Adebayo, O. Osho, I. Idris, J. K. Alhassan and N. Rana, "Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification," SN Applied Sciences, vol. 1, no. 5, pp. 390, 2019. https://doi.org/10.1007/s42452-019-0394-7

[42]. P. Ovhal, S. Kulkarni and J. K. Valadi, "Improved Filter Ranking Incorporated Binary Black Hole Algorithm for Feature Selection," SN Computer Science, vol. 3, no. 1, pp. 1-23, 2021. https://doi.org/10.1007/s42979-021-00933-w

[43]. R. A. Ibrahim, M. A. Elaziz, D. Oliva, E. Cuevas and S. Lu, "An opposition-based social spider optimization for feature selection," Soft Computing, vol. 23, no. 24, pp. 13547-13567, 2019. https://doi.org/10.1007/s00500-019-03891-x

spam detection. in 2016 IEEE Congress on Evolutionary Computation (CEC). 2016. 1022-1029. https://doi.org/10.1109/CEC.2016.7743901

[29]. M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," Expert Systems with Applications, vol. 117, no. pp. 267-286, 2019. https://doi.org/10.1016/j.eswa.2018.09.015

[30]. W. Ma, D. Tran and D. Sharma. A novel spam email detection system based on negative selection. in 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology. 2009. 987-992. https://doi.org/10.1109/ICCIT.2009.58

[31]. S. Ozawa, J. Nakazato, T. Ban and J. Shimamura. An autonomous online malicious spam email detection system using extended RBF network. in 2015 International Joint Conference on Neural Networks (IJCNN). 2015. 1-7. https://doi.org/10.1109/IJCNN.2015.7280826

[32]. H. Mohammadzadeh, F. S. Gharehchopogh, "A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection," Computational Intelligence, vol. 37, no. 1, pp. 176-209, 2021. https://doi.org/10.1111/coin.12397

[33]. S. Abu-Nimeh, D. Nappa, X. Wang and S. Nair. Bayesian additive regression trees-based spam detection for enhanced email privacy. in 2008 Third International Conference on Availability, Reliability and Security. 2008. 1044-1051. https://doi.org/10.1109/ARES.2008.136

[34]. S. B. Rathod, T. M. Pattewar. Content based spam detection in email using Bayesian classifier. in 2015 International Conference on Communications and Signal Processing (ICCSP). 2015. 1257-1261. https://doi.org/10.1109/ICCSP.2015.7322709

[35]. A. A. Naem, N. I. Ghali and A. A. Saleh, "Antlion optimization and boosting classifier for spam email detection," Future Computing and Informatics Journal, vol. 3, no. 2, pp. 436-442, 2018. https://doi.org/10.1016/j.fcij.2018.11.006