# ارائه یک فیلتر جدید برای حذف نویزهای ضربه‌ای و ترکیب فیلتر پیشنهادی با الگوریتم PSO به‌منظور کشف و دفاع در برابر حملات سیل‌آسای SYN

محمد مؤمنی[۱]، دانشجوی دکتری؛ مهدی آقاصرام[۲]، استادیار؛ وحید شاکر[۳]، دانشجوی دکتری؛ شهرام جمالی[۴]، دانشیار؛ مهدی نوشیار[۵]، استادیار

۱- گروه مهندسی کامپیوتر - پردیس فنی و مهندسی - دانشگاه یزد - یزد - ایران - mohamad.momeny@stu.yazd.ac.ir
۲- گروه مهندسی کامپیوتر - پردیس فنی و مهندسی - دانشگاه یزد - یزد - ایران - mehdi.sarram@yazd.ac.ir
۳- گروه مهندسی کامپیوتر - دانشگاه آزاد اسلامی واحد علوم و تحقیقات - تهران - ایران - vahidshaker93@gmail.com
۴- دانشکده فنی‌مهندسی - دانشگاه محقق اردبیلی - اردبیل - ایران - jamali@iust.ac.ir
۵- دانشکده فنی‌مهندسی - دانشگاه محقق اردبیلی - اردبیل - ایران - nooshyar@uma.ac.ir

**چکیده:** در حمله‌های SYN-flooding، مهاجم با ایجاد ترافیک بی‌استفاده، حجم زیادی از منابع سرویس‌دهنده و پهنای باند شبکه را مصرف کرده و یا سرویس‌دهنده را به نوعی درگیر رسیدگی به این تقاضاهای بی‌مورد می‌کند. برای پی‌ریزی این حمله، از ضعف پروتکل TCP در برقراری ارتباط بین دو کامپیوتر استفاده می‌کنند، جایی که الگوریتم دست‌تکانی سه‌مرحله‌ای استفاده شده است. این مقاله سیستم تحت حمله را با استفاده از تئوری صف‌بندی مدل‌سازی کرده و مسئله دفاع در برابر حملات SYN-flooding را به یک مسئله بهینه‌سازی نگاشت می‌کند سپس با ارائه یک فیلتر جدید برای حذف نویزهای ضربه‌ای در تصاویر نویزی و ترکیب فیلتر ارائه‌شده با الگوریتم PSO، رهیافت پیشنهادی خود را معرفی کرده و به حل این مسئله می‌پردازد. نتایج شبیه‌سازی نشان می‌دهد که مکانیزم دفاعی پیشنهادی از نظر میزان درخواست‌های بلوکه‌شده، احتمال موفقیت در برقراری ارتباط، کاهش احتمال موفقیت مهاجم و همچنین استفاده بهینه از بافر اختصاص داده‌شده دارای کارایی مؤثر است.

**واژه‌های کلیدی:** نویز ضربه‌ای، فیلتر، حذف نویز تصویر، الگوریتم PSO، DoS، حملات SYN-flooding، TCP.

# Proposing a New Filter to Remove Impulse Noise and its Composition with PSO Algorithm to Detect and Defense against SYN Flooding Attacks

Mohammad Momeny[1], PhD Student; Mehdi Agha Sarram[2], Assistant Professor; Vahid Shaker[3], PhD Student; Shahram Jamali[4], Associate Professor; Mehdi Nooshyar[5], Assistant Professor

1- Department of Computer Engineering, Yazd University, Yazd, Iran, Email: mohamad.momeny@stu.yazd.ac.ir
2- Department of Computer Engineering, Yazd University, Yazd, Iran, Email: mehdi.sarram@yazd.ac.ir
3- Department of Computer Engineering, Islamic Azad University of Science and Research Branch, Tehran, Iran, Email: vahidshaker93@gmail.com
4- Faculty of Technical and Engineering, University of Mohaghegh Ardabili, Ardabil, Iran, Email: jamali@iust.ac.ir
5- Faculty of Technical and Engineering, University of Mohaghegh Ardabili, Ardabil, Iran, Email: nooshyar@uma.ac.ir

**Abstract:** In a DoS attack can be regarded as an attempt of attackers to prevent legal users from gaining a normal network service. The TCP connection management protocol sets a position for a classic DoS attack, namely, the SYN flood attack. In this attack some sources send a large number of TCP SYN segments, without completing the third handshake step to quickly exhaust connection resources of the under attack system. This paper models the under attack server by using the queuing theory in which attack requests are recognized based on their long service time. Then it proposes a framework in which the defense issue is formulated as an optimization problem and employs combination of the particle swarm optimization (PSO) algorithm and proposed filter for reduction degradation caused by impulsive noise to optimally solve this problem. The goal is to direct the server to an optimum defense point by dynamically setting of two TCP parameters, namely, maximum number of connections and maximum duration of a half-open connection. The simulation results show that the proposed defense strategy improves the performance of the under attack system in terms of rejection probability of connection requests and efficient consumption of buffer space.

**Keywords:** Impulse noise removal, Image enhancements, Filter, PSO, DoS attack, SYN flooding, TCP.

تاریخ ارسال مقاله: ۱۳۹۲/۱۲/۲۵
تاریخ اصلاح مقاله: ۱۳۹۳/۰۷/۱۴ و ۱۳۹۳/۰۸/۲۰
تاریخ پذیرش مقاله: ۱۳۹۴/۰۱/۲۶
نام نویسنده مسئول: محمد مؤمنی
نشانی نویسنده مسئول: ایران - یزد - صفائیه - بلوار دانشگاه - دانشگاه یزد - پردیس فنی و مهندسی - گروه مهندسی کامپیوتر.

## ۱- مقدمه

حملات در شبکه‌های کامپیوتری حاصل پیوند عواملی از قبیل سرویس‌های فعال، پروتکل‌های استفاده‌شده و پورت‌های باز است. برخی از سرویس‌ها دارای استعداد لازم برای حملات بوده و لازم است در زمان پیکربندی آن‌ها، مسائل امنیتی مورد توجه قرار گرفته شود. از میان حمله‌های موجود برای از بین بردن امنیت شبکه‌های کامپیوتری، حمله DoS نوعی از حمله است که هدف آن از کار اندازی سیستم هدف با استفاده از هدر دادن منابع آن است. متداول‌ترین نوع حمله DoS، حمله‌های SYN flooding است که سبب می‌شود منابع اختصاص‌یافته برای برقراری ارتباط در سیستم تحت حمله به سرعت مصرف شود درنتیجه، سیستم تحت حمله از ادامه فعالیت باز می‌ماند. برای پی‌ریزی این حمله، از ضعف پروتکل TCP در برقراری ارتباط بین دو کامپیوتر استفاده می‌کنند، جایی که الگوریتم دست‌تکانی سه‌مرحله‌ای استفاده شده است. در این الگوریتم، شروع‌کننده ارتباط (مبدأ) یک بسته SYN به طرف مقابل (مقصد) می‌فرستد و مقصد با دریافت این بسته، آن را در صف پشتیبان قرار داده، یک ارتباط نیمه‌باز ایجاد کرده و یک بسته SYN-ACK به مبدأ ارسال می‌کند. حال اگر مبدأ، به این بسته فرستاده‌شده، با ارسال یک بسته ACK جواب دهد، ارتباط نیمه‌باز ایجادشده خاتمه یافته و منابع اختصاص داده‌شده به آن، آزاد می‌شود و ارتباط بین مبدأ و مقصد برقرار می‌شود. اما اگر آدرس IP مبدأ، جعلی باشد، این بسته فرستاده‌شده، بی‌جواب خواهد ماند. بنابراین ارتباط نیمه‌باز ایجادشده تا زمانی که مقصد آن را خاتمه دهد، باقی می‌ماند. این تقاضاها تا جایی که دستگاه سرویس‌دهنده را از کار بیندازد، ادامه پیدا می‌کنند به‌گونه‌ای که سیستم سرویس‌دهنده دیگر توانایی پاسخ‌گویی به کاربران قانونی خود را نداشته باشد.

Haris و همکارانش در [۱] از کشف ناهنجاری برای تشخیص حملات SYN-flooding استفاده کرده‌اند. Nski و همکارانش طرح نمونه‌برداری دقیق برای دفاع در برابر حملات SYN پیشنهاد کردند که این طرح به بررسی بخش‌های TCP برای پیدا کردن حداقل یکی از بخش‌های چندگانه ACK که به سرور وارد می‌شوند، جهت تأیید اعتبار اتصالات قانونی می‌پردازد [۲]. Al-Dabagh و همکارانش در [۳] طراحی و پیاده‌سازی یک سیستم ایمنی مصنوعی برای تشخیص حملات SYN-flooding بر اساس الگوریتم DCA را ارائه دادند. Bo و همکارانش با استفاده از الگوریتم CUSUM، روشی برای تشخیص حملات SYN-flooding در [۴] ارائه دادند. Bhirud و همکارانش از مکانیزم وزن سبک برای تشخیص و پیشگیری از حملات SYN-flooding با استفاده از MMDBMS را پیشنهاد دادند [۵]. Arshadi و همکارانش در [۶] با بررسی آنتروپی بسته SYN بین زمان ورود، یک معیار اندازه‌گیری تصادفی را ارائه دادند. Safa و همکارانش در [۷] مکانیسم دفاعی پیشنهاد کردند که از طریق روتر لبه با شناسایی آدرس IP جعلی شبکه برای تعیین اعتبار SYN-ACK ورودی استفاده می‌کند. Xiao و همکارانش در [۸] روشی را ارائه کردند که با استفاده از اطلاعات تأخیر توزیع ترافیک شبکه، حملات را به‌طور مستقل در طرف قربانی تشخیص می‌دهد. Kim و همکارانش در [۹] برای مقابله با حملات روش لیست سفید را ارائه دادند که سعی دارد احتمال موفقیت ارتباطات قانونی را افزایش دهد. Y. W در [۱۰] مکانیزم کنترل پارامترهای مورد استفاده (UPC) در مد انتقال ناهمگام (ATM) شبکه‌ها در جهت محافظت از سرور تحت حملات DoS به‌کار برده است. Xiaofeng و همکارانش به جزئیات مورد بحث، با تمرکز بر روی ویژگی‌های سرویس حفاظت، دفع حمله و چگونگی ردیابی منبع حملات می‌پردازند [۱۱]. Khan و Traore در [۱۲]، حمله‌های DoS از نوع flooding را بررسی کرده‌اند و در این بررسی روی منبع cpu تمرکز کرده‌اند و برای کشف حمله‌های DoS از سه پارامتر نرخ ورود درخواست‌ها، نرخ رشد صف و زمان پاسخ‌گویی استفاده کرده‌اند.

Wang و همکارانش در [۱۳]، برای ارزیابی حمله‌های DoS در شبکه‌های کامپیوتری از یک مدل صف‌بندی استفاده کرده‌اند. در این روش، شبکه تحت حملات DoS، به‌وسیله یک مدل زنجیری دوبعدی مارکوف مشخص شده است و در مدل آن‌ها بر روی مصرف منابع محدود که یک مشخصه تعیین‌کننده برای حمله‌های DoS مختلف است، تمرکز کرده‌اند و برای کشف حمله‌های DoS، از شاخص‌های احتمال گم‌شدن درخواست‌ها و درصد تملک بافر ارتباطات نیمه‌باز استفاده کرده‌اند. Crosby در [۱۴]، یک مثالی از حمله پهنای باند ارائه می‌کند. ولی در آن یک مکانیسم کلی برای کشف حمله ارائه نمی‌شود و برای جلوگیری از حمله‌های DoS، استفاده از یک الگوریتمی که قابلیت آسیب‌پذیری پایینی داشته باشد را پیشنهاد می‌کند.

در ادامه این مقاله بخش دوم به تشریح الگوریتم بهینه‌سازی اجتماع ذرات، می‌پردازد. در بخش سوم فیلتر پیشنهادی برای حذف نویز ضربه‌ای در تصاویر نویزی ارائه می‌گردد. در بخش چهارم نحوه نگاشت الگوریتم PSO، فیلتر ارائه‌شده و حمله SYN-flooding به‌طور کامل شرح داده شده و رهیافت پیشنهادی برای کشف و دفاع در برابر حملات سیل‌آسای SYN، ارائه می‌گردد. در بخش پنجم رهیافت ارائه‌شده، در محیط MATLAB شبیه‌سازی شده و نتایج به‌دست‌آمده، مورد ارزیابی قرار می‌گیرد. بخش ششم شامل نتیجه‌گیری این مقاله است.

## ۲- الگوریتم بهینه‌سازی اجتماع ذرات

الگوریتم PSO از رفتار اجتماعی دسته پرندگان یا گروه ماهی‌ها در حین جستجوی غذا، برای هدایت جمعیت به منطقه امیدبخش در فضای جستجو استفاده می‌کند [۱۵، ۱۶]. در PSO هر جواب مسئله، موقعیت یک پرنده در فضای جستجو است که آن را ذره می‌نامند. تمام ذره‌ها دارای یک مقدار شایستگی هستند که توسط تابع برازندگی که هدف بهینه‌سازی است، به‌دست می‌آیند. همچنین هر ذره دارای مؤلفه‌ای به نام سرعت است که مسیر حرکت آن را در فضای جستجو

تعیین می‌کند. جمعیت PSO شامل تمامی ذره‌هاست که Swarm نامیده می‌شود.

الگوریتم PSO شامل دو مدل معادله سرعت و مکان است. مختصات هر ذره نمایانگر یک جواب ممکن مرتبط با دو بردار است. بردارهای موقعیت ($X_i$) و سرعت ($V_i$) دو بردار وابسته و مرتبط با هر ذره i در فضای جستجوی N بعدی می‌باشند که به‌ترتیب به‌صورت زیر بیان می‌گردند:

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{iN}] \quad (۱)$$

$$\mathbf{V_i} = [v_{i1}, v_{i2}, \ldots, v_{iN}] \quad (۲)$$

یک اجتماع از پرندگان از تعدادی ذره (پاسخ‌های ممکن) تشکیل شده است که در یک فضای پاسخ ممکن برای جستجوی جواب‌های بهینه پیش می‌روند (پرواز می‌کنند). موقعیت هر ذره بر اساس بهترین جستجوی خود ذره، بهترین تجربه کلی پرواز گروهی و بردار سرعت پیشین خود ذره، بر اساس روابط زیر به‌هنگام می‌شود:

$$x_i^{k+1} = x_i^k + Cv_i^{k+1} \quad (۳)$$

$$v_i^{k+1} = wv_i^k + c_1r_1\left(Pbest_i^k - x_i^k\right) + c_2r_2\left(Gbest^k - x_i^k\right) \quad (۴)$$

که در آن، $c_1$ و $c_2$ دو ثابت عددی مثبت، $r_1$ و $r_2$ دو عدد تصادفی با توزیع یکنواخت در محدوده [۰،۱] و w وزن لختی است که به‌صورت زیر انتخاب می‌گردد:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (۵)$$

که $iter_{max}$، تعداد ماکزیمم تکرار و iter تعداد تکرار جاری است. $Pbest_i^k$ بهترین موقعیت ذره i است که بر اساس تجربه ذره به‌دست آمده و به‌صورت زیر قابل بیان است:

$$Pbest_i^k = [x_{i1}^{pbest}, x_{i2}^{pbest}, \ldots, x_{iN}^{pbest}] \quad (۶)$$

$Gbest^k$ بهترین موقعیت ذره بر اساس تجربه کلی گروهی است و عبارت است از:

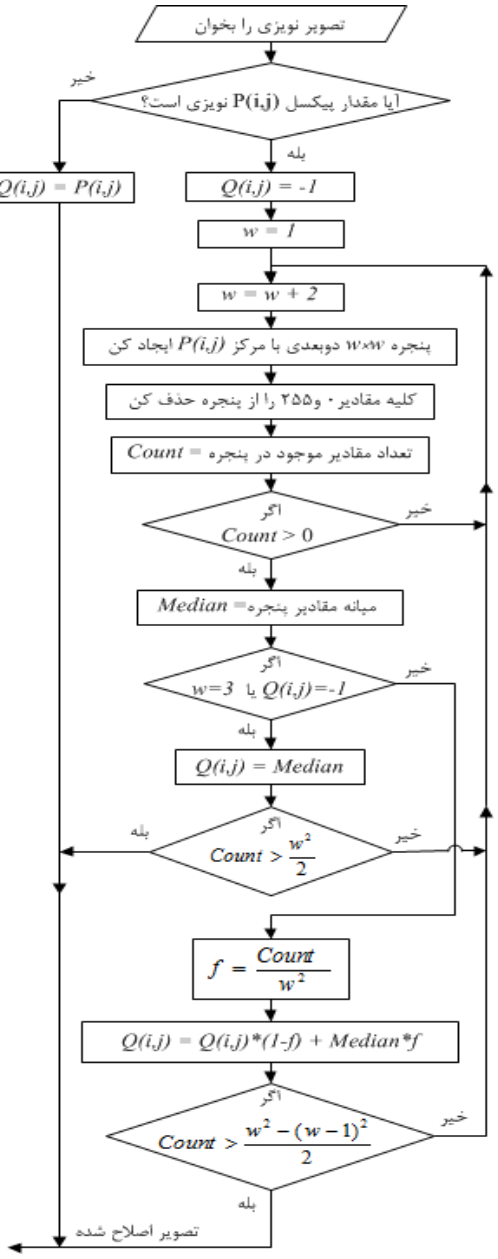
$$Gbest = [x_1^{gbest}, x_2^{gbest}, \ldots, x_N^{gbest}] \quad (۷)$$

k هم شاخص تکراراست.

## ۳- فیلتر پیشنهادی برای حذف نویز ضربه‌ای

حذف اختلالات ناخواسته نظیر انواع نویزها و بهبود کیفیت تصاویر نویزی، در کاربردهای مختلف مرتبط با پردازش تصویر، جزء چالش‌های بسیار ضروری محسوب می‌شود. نویز ضربه‌ای که در آن اختلاف شدت پیکسل نویزی با پیکسل‌های اطراف بسیار زیاد است، یکی از عوامل تضعیف کیفیت در تصاویر دیجیتال است. یکی از مسائل در زمینه حذف نویز ضربه‌ای، نگهداری جزئیات تصویر نویزی مانند لبه‌ها و بافت تصویر به‌همراه کاهش نویز است [۱۷].

در فیلتر پیشنهادی در این مقاله، برای تشخیص و حذف نویز ضربه‌ای، کل تصویر نویزی از ابتدا پیمایش شده و پیکسل به پیکسل مورد پردازش قرار می‌گیرد. اگر پیکسل جاری مقدار بیش‌ترین شدت (مقدار ۲۵۵) یا کم‌ترین شدت (مقدار ۰) را دارا باشد، آنگاه مقدار پیکسل، نویزی در نظر گرفته می‌شود. برای اصلاح مقدار پیکسل نویزی، پنجره دوبعدی با مرکزیت پیکسل جاری ایجاد می‌شود. اندازه پنجره بر اساس تعداد پیکسل‌ها با مقدار واقعی در همسایگی پیکسل جاری تعیین می‌شود. اعمال فیلتر میانه و میانگین وزن‌دار برای مقادیر واقعی (بدون نویز) موجود در پنجره به‌عنوان مقدار اصلاح‌شده پیکسل نویزی، در نظر گرفته می‌شود. در شکل (۱) با فرض این‌که $P$ تصویر نویزی و $Q$ تصویر اصلاح‌شده باشد، ساختار فیلتر پیشنهادی در این مقاله نمایش داده شده است.




**شکل ۱: ساختار فیلتر پیشنهادی برای حذف نویز ضربه‌ای از تصاویر نویزی**

نتیجه حاصل از اجرای فیلتر پیشنهادی برای تصویر استاندارد لنا، با چگالی نویز ضربه‌ای ۲۰٪ در شکل (۲) نمایش داده شده است. برای ارزیابی کارایی فیلترهای حذف نویز ضربه‌ای از تصاویر نویزی، از معیار PSNR، طبق روابط زیر استفاده می‌کنیم [۱۷]:

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \quad (۸)$$

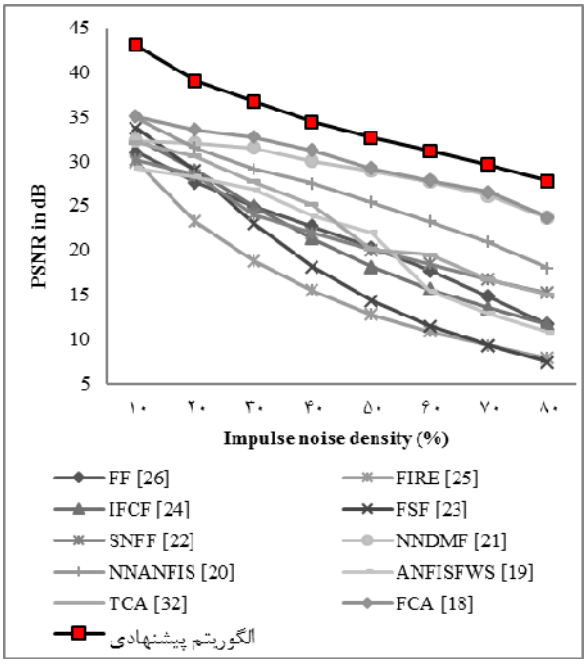
$$MSE = \frac{\Sigma_{i=1}^{N}\Sigma_{j=1}^{M}([O(i,j)-Q(i,j))^2}{M\times N} \quad (۹)$$

فیلتر پیشنهادی با روش‌های مختلف تشخیص و حذف نویز ضربه‌ای از قبیل: FCA[۱] [۱۸]، ANFISFWS[۲] [۱۹]، NNANFIS[۳] [۲۰]، NNDMF[۴] [۲۱]، SNFF[۵] [۲۲]، FSF[۶] [۲۳]، IFCF[۷] [۲۴]، FF[۹] [۲۵]، FIRE[۸] [۲۶]، TCA[۱۰] [۲۷] مورد مقایسه قرار گرفته است. نتایج مقایسه فیلترها در شکل (۳) نشان داده شده است.

نمودار عملکرد فیلتر پیشنهادی در قیاس با فیلترهای ذکرشده با آزمایش بر روی تصاویر استاندارد Lena با چگالی نویز ضربه‌ای از ۱۰٪ تا ۸۰٪ در شکل (۳) نشان‌دهنده برتری فیلتر پیشنهادی نسبت به روش‌های ذکرشده است.

الف ب ج

**شکل ۲: الف. تصویر اصلی، ب. تصویر با چگالی نویز ضربه‌ای ۲۰٪، ج. تصویر بهبودیافته با فیلتر پیشنهادی**




**شکل ۳: نمودار مقایسه PSNR تصویر با چگالی نویز از ۱۰٪ تا ۸۰٪ برای تصویر استاندارد لنا**

## ٤- رهیافت پیشنهادی برای کشف و دفاع در برابر حملات سیل‌آسای SYN

پروتکل TCP در فرایند برقراری ارتباط از دو پارامتر اصلی m و h استفاده می‌کند تا مدت‌زمان نگه‌داری ارتباطات نیمه‌باز و همچنین تعداد این ارتباطات را کنترل نماید. بنابراین m و h دو پارامتری هستند که رفتار پروتکل TCP را تحت تأثیر خود قرار می‌دهند. به‌همین خاطر، m و h را به‌عنوان ذرات اجتماع در نظر می‌گیریم. با توجه به این‌که در پروتکل TCP، m و h مقادیر ثابتی هستند، با استفاده از روش بهینه‌سازی اجتماع ذرات، این مقادیر به‌صورت پویا تغییر کرده و با حرکت به‌سمت نقطه بهینه، باعث حذف زودهنگام ارتباطات نیمه‌بازی که برای درخواست‌های حمله اختصاص داده شده است، می‌شوند و همچنین تعداد کل ارتباطاتی که سیستم می‌تواند ایجاد نماید را افزایش می‌دهند و به‌طورکلی باعث بهبود عملکرد TCP در برابر حملات SYN-flooding می‌شوند.

در این مقاله، ابتدا با استفاده از الگوریتم PSO مقادیر m و h را در هر مرحله محاسبه و به‌روز کرده و بعد از یک‌سری اجرا، فیلتر ارائه‌شده به کار گرفته می‌شود تا از مجموعه داده‌ای که برای m و h از طریق الگوریتم PSO حاصل شده است، مقداری را انتخاب کرده و بقیه مقادیر را فیلتر نماید.

### ۴-۱- به‌کارگیری الگوریتم PSO

روش PSO، شامل دو مدل معادله سرعت و مکان است. مختصات هر ذره نمایانگر یک جواب ممکن مرتبط با دو بردار است. بردارهای موقعیت (xi) و سرعت (vi) دو بردار وابسته و مرتبط با هر ذره i در فضای جستجوی N بعدی می‌باشند. که به‌ترتیب به‌صورت $x_i=[x_{i1}, x_{i2},\ldots, x_{iN}]$ و $v_i=[v_{i1}, v_{i2}, \ldots, v_{iN}]$ بیان می‌گردند.

$$v_h^{k+1} = wv_h^k + c_1r_1\left(Pbest_h^k - h^k\right) + c_2r_2\left(Gbest^k - h^k\right) \quad (۱۰)$$

$$v_m^{k+1} = wv_m^k + c_1r_1\left(Pbest_m^k - m^k\right) + c_2r_2\left(Gbest^k - m^k\right) \quad (۱۱)$$

که در آن، $c_1$ و $c_2$ دو ثابت عددی مثبت، r1 و r2 دو عدد تصادفی می‌باشند. $Pbest_i^k$ بهترین موقعیت ذره i است که بر اساس تجربه ذره به‌دست‌آمده و بر طبق معادله زیر بیان می‌شود:

$$Pbest_i^k = [x_{i1}^{pbest}, x_{i2}^{pbest}, \ldots, x_{iN}^{pbest}] \quad (۱۲)$$

$Gbest^k$ بهترین موقعیت ذره بر اساس تجربه کلی گروهی است و به‌صورت زیر تعریف می‌شود:

$$Gbest = [x_1^{gbest}, x_2^{gbest}, \ldots, x_N^{gbest}] \quad (۱۳)$$

همان‌طور که گفته شد در الگوریتم پیشنهادی ذره‌ها m و h هستند. لذا سرعت باید مقادیر m و h را به مقادیر Gbest نزدیک‌تر کند. Pbest در الگوریتم پیشنهادی زمانی حاصل می‌شود که مقادیر m و h باعث شوند تا معیارهای مورد نظر ما به حد بهینه خود برسند. در این

حالت m و h به‌هنگام شده‌ای که سبب شود تا معیارهای مورد نظر ما بهینه باشد، به‌عنوان m و h برای Pbest در نظر گرفته می‌شود. همچنین بهترین سراسری زمانی حاصل می‌شود که مقادیر m و h با مقادیر Gbest برابر باشند. با جایگذاری Pbest و Gbest در معادله ۱۰، سرعت ذرات محاسبه می‌شود. و سپس با جایگذاری سرعت ذرات در معادله ۱۱، موقعیت ذرات را می‌توان محاسبه کرد.

برای محاسبه موقعیت بعدی هر ذره، از روابط زیر استفاده می‌شود.

$$h^{k+1}=h^{k}+v_h^{k+1} \quad (۱۴)$$

$$m^{k+1}=m^{k}+v_m^{k+1} \quad (۱۵)$$

$h_i+1$ و $m_i+1$ به‌دست‌آمده از روابط (۱۲) و (۱۳)، ذره‌های جدید هستند و به‌جای m و h موجود در مرحله قبل، جایگزین می‌شوند. بدین طریق مقادیر m و h که در پروتکل TCP، به‌صورت استاتیک تعریف شده بودند، در الگوریتم PSO، تبدیل به متغیرهای پویا شده و بسته به شرایط شبکه، تغییر می‌یابند.

### ۴-۲- به‌کارگیری فیلتر پیشنهادی

در الگوریتم PSO، رسیدن به مقادیر m و h بهینه، بر اساس تجربه و گذر زمان میسر می‌شود. با تشدید ورود درخواست‌های حمله، الگوریتم PSO در تشخیص m و h بهینه با خطا مواجه شده و بهبود مقادیر m و h به‌کندی انجام می‌شود. با توجه به اینکه زمان نقش بسیار مهمی در کشف و دفاع در برابر حملات سیل‌آسای SYN ایفا می‌کند، برای تسریع در رسیدن به مقادیر بهینه m و h، نتایج الگوریتم PSO را با استفاده از الگوریتم ارائه‌شده فیلتر می‌کنیم. به‌عبارت دیگر پس از تعیین مقادیر m و h بهینه توسط الگوریتم PSO، مقادیر m و h را فیلتر کرده تا نتایج بهبود یابد. فرض این‌که:

• پنجره: مقادیر همسایه‌های یک گره
• اندازه پنجره: به‌عنوان پارامتر ورودی تعیین می‌شود.
• بالاترین شدت: بیش‌ترین مقدار h موجود در پنجره
• پایین‌ترین شدت: کم‌ترین مقدار h موجود در پنجره
• مقدار معمول: مقداری غیر از بیش‌ترین و کم‌ترین مقدار در پنجره
• مقدار نویزی: مقدار بالاترین شدت یا پایین‌ترین شدت در پنجره

برای اعمال فیلتر پیشنهادی، در ابتدا پنجره‌ای با مقادیر همسایه‌های یک گره در اندازه مشخص‌شده با پارامتر ورودی شکل می‌گیرد. اگر گره دارای مقدار بالاترین شدت یا پایین‌ترین شدت بود از روش زیر برای فیلتر استفاده می‌کنیم.

• گام اول: پنجره به‌عنوان پارامتر ورودی تعیین می‌شود
• گام دوم: Q = -1
• گام سوم: مقادیر بیش‌ترین شدت و کم‌ترین شدت در پنجره ایجادشده را نادیده بگیر.
• گام چهارم: تعداد مقادیر باقی‌مانده در پنجره انتخاب‌شده = count
• گام پنجم: اگر count = 0 آنگاه:
اندازه پنجره را یک واحد بزرگ‌تر کن و به گام سوم برو.
در غیر این‌صورت:
میانه مقادیر باقی‌مانده در پنجره =Med
• گام ششم: اگر Q=-1 آنگاه:
Q=Med
اگر تعداد مقادیر معمول در پنجره بیش‌تر از تعداد مقادیر نویزی بود آنگاه:
Q به‌عنوان مقدار اصلاح‌شده در نظر گرفته می‌شود.
• گام هفتم: اگر Q>-1 آن‌گاه:

$$f = \frac{\text{تعداد مقادیر معمول موجود در پنجره}}{\text{تعداد کل مقادیر موجود در پنجره}}$$

$$Q = Q\times(1-f) + Med\times f$$

• گام هشتم: اگر مقدار اصلاح‌شده برای پیکسل نویزی (P) یافت نشد؛ آن‌گاه، اندازه پنجره را یک واحد بزرگ‌تر کن و به گام سوم برو.
• گام نهم: مراحل فوق برای همه پیکسل‌ها با مقادیر نویزی تکرار می‌شود.

به‌طور مشابه، برای فیلتر کردن m نیز از روش فوق استفاده می‌کنیم. درخواست‌هایی که با بسته SYN وارد سرور می‌شوند، پروتکل TCP، آن‌ها را در بافر پشتیبان قرار داده و منابع لازم برای برقراری یک ارتباط کامل را از بافر پشتیبان به آن‌ها تخصیص می‌دهد. این حالت، حالت ارتباط نیمه‌باز (connection Half open) نامیده می‌شود. از طرفی تعداد ارتباط نیمه‌بازی که سرور می‌تواند ایجاد نماید، محدود بوده و یک مقدار ماکزیممی دارد و همچنین مدت‌زمان نگه‌داری این ارتباطات نیمه‌باز، یک زمان ثابتی است. ما می‌خواهیم با استفاده از رهیافت پیشنهادی، این دو پارامتر را به‌صورت پویا و با توجه به وضعیت شبکه تغییر داده می‌شود. برای این کار پارامترهای زیر را تعریف می‌کنیم.

**Ploss**[11]: یک بسته، وقتی loss می‌شود که سرور به‌دلیل پر شدن بافر نتواند به درخواست رسیده مبنی بر ایجاد ارتباط، پاسخ دهد. بنابراین، Ploss را نسبت تعداد بسته‌های بلوکه‌شده به تعداد کل بسته‌های واردشده به سرور، تعریف می‌کنیم.

**درصد تملک بافر توسط درخواست‌های عادی ($P_R$):** وقتی یک ارتباط نیمه‌باز ایجاد شد، در بافر قرار می‌گیرد. مجموع مدت‌زمان ارتباطات نیمه‌باز ایجادشده توسط درخواست‌های عادی نسبت به مجموع مدت‌زمان همه ارتباطات نیمه‌باز را درصد تملک بافر توسط درخواست‌های عادی تعریف می‌کنیم.

**درصد تملک بافر توسط درخواست‌های حمله ($P_A$):** به‌طور مشابه، مجموع مدت‌زمان ارتباطات نیمه‌باز ایجادشده توسط درخواست‌های حمله نسبت به مجموع مدت‌زمان همه ارتباطات نیمه‌باز را درصد تملک بافر توسط درخواست‌های حمله تعریف می‌کنیم.

برای بالا بردن توانائی یک سرور در ارائه خدمات، بایستی مقدار Ploss به‌اندازه کافی کوچک باشد. همچنین برای این‌که سرور به درخواست‌های عادی سرویس بیش‌تری ارائه کند، مقدار تملک بافر توسط درخواست‌های عادی، باید به‌اندازه کافی بزرگ باشد و به‌دلیل مشابه بایستی مدت تملک بافر توسط درخواست‌های حمله، به اندازه کافی کوچک باشد. ما از این اطلاعات استفاده کرده و تابع هدف برای الگوریتم PSO را به‌صورت زیر تعریف می‌کنیم:

$$\text{Objective Function: Maximize } P_R / P_A \qquad (۱۶)$$

بنابراین، هر چقدر مقدار این تابع ماکزیمم شود، توانائی سرور در ارائه خدمات مطلوب است. بنابراین رهیافت پیشنهادی، سعی در ماکزیمم کردن این تابع دارد. شکل (۴) نمایانگر برازندگی و تنظیم مقادیر m و h بر اساس الگوریتم PSO و فیلتر ارائه شده است.


**شکل ۴: برازندگی و تنظیم مقادیر m و h بر اساس الگوریتم PSO و فیلتر پیشنهادی**

## ۵- شبیه‌سازی و ارزیابی نتایج

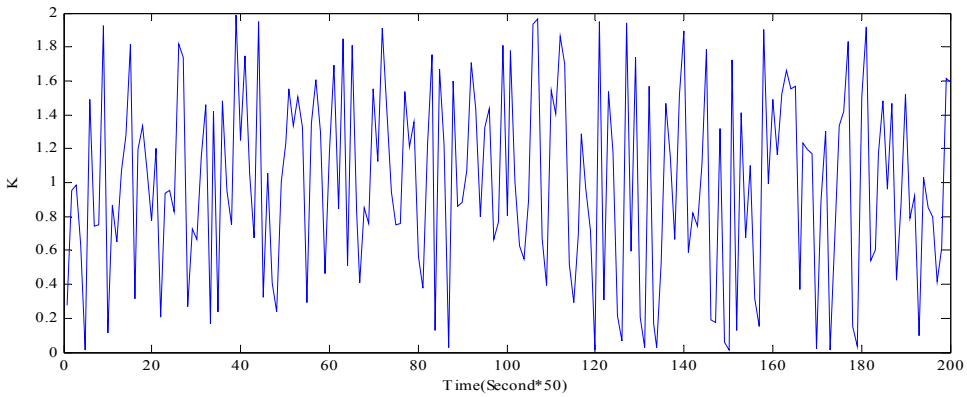
برای شبیه‌سازی حمله‌های SYN flooding، از توزیع‌های آماری برای ورود درخواست‌های عادی و حمله استفاده شده است. نرخ ورود درخواست‌های عادی به سرور توزیع پواسون با آهنگ میانگین λ، در نظر گرفته شده است. همچنین نرخ ورود درخواست‌های حمله نیز توزیع پواسون با آهنگ میانگین kλ، است که k شدت ورود درخواست‌های حمله نسبت به درخواست‌های عادی است. مدت‌زمان نگه‌داری ارتباطات نیمه‌باز برای درخواست‌های عادی، توزیع نمایی با میانگین μ است. همچنین ارتباطات نیمه‌باز برای درخواست‌های حمله، به مدت‌زمان h نگه داشته می‌شوند. برای بررسی روش پیشنهادی از حالتی استفاده می‌کنیم که در آن، مقادیر m و h به‌صورت پیش‌فرض و ثابت استفاده می‌شوند و هیچ مکانیزم دفاعی وجود ندارد. مبنای مقادیر پیش‌فرض را برای m و h مقادیر Linux2.4 در نظر می‌گیریم که به‌ترتیب برابر ۷۵ و ۱۲۸ است. در هر دو روش، λ=10/s و μ=100/s در نظر گرفته شده‌اند و همچنین مقدار k با توزیع یکنواخت بین بازه ۰ تا ۲ تغییر داده شده‌اند تا سرور مورد نظر، تحت شدت حمله‌های مختلف مورد آزمایش قرار گیرد.

این مسئله در محیط Matlab پیاده‌سازی شده و با توجه به سربار محاسباتی پایین الگوریتم بهینه‌سازی اجتماع ذرات، زمان صرف‌شده برای اجرای الگوریتم ناچیز است. ضمن اینکه اجرای الگوریتم نیاز به سخت‌افزار خاصی نداشته و الگوریتم روی سیستم‌های عادی قابل اجرا است.

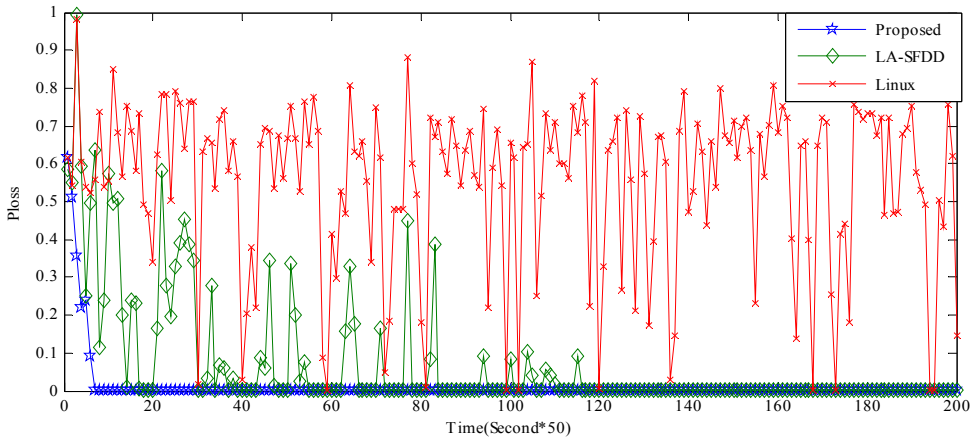
تغییرات مقدار k در طول شبیه‌سازی، در شکل (۵) نشان داده شده است. مقایسه احتمال بلوکه شدن درخواست‌ها با استفاده از رهیافت پیشنهادی در این مقاله، روش LA-SFDD [۲۸] و Linux در شکل (۶) نمایش داده شده است.

همان‌طوری در شکل (۶) مشاهده می‌شود، رهیافت پیشنهادی نسبت به روش LA-SFDD و Linux، دارای مقادیر مطلوب‌تری از نظر سرویس‌دهی به کاربران عادی را دارا می‌باشد. مقدار Ploss رهیافت پیشنهادی در قیاس با روش‌های ذکرشده به‌صورت قابل توجهی، کم‌تر است که این بدین معناست که روش پیشنهادی بعد از رسیدن به حالت پایدار تقریباً به همه درخواست‌های کاربران پاسخ می‌دهد و هیچ درخواستی بلوکه نمی‌شود درحالی‌که روش‌های LA-SFDD و Linux، توانایی پاسخ به درخواست‌های همه کاربران را نداشته و از سرویس‌دهی به تعداد قابل توجهی از درخواست‌های کاربران ناتوان هستند.

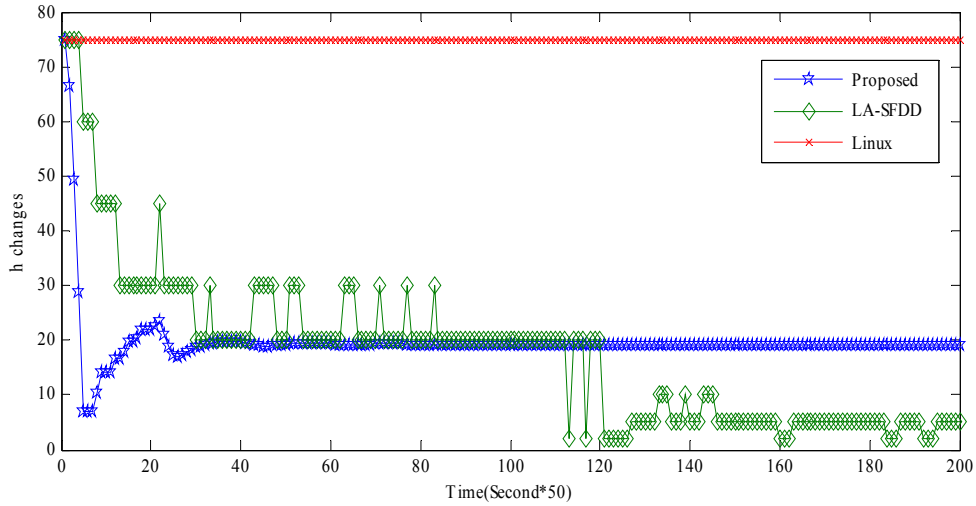
این بهبودها در روش پیشنهادی، حاصل تغییرات پویای مقادیر m و h است که تغییرات این پارامترها نیز به‌ترتیب در شکل‌های (۷) و (۸) ارائه شده است. همان‌طور که از این شکل‌ها مشخص است روش پیشنهادی با توجه به شرایط موجود در سیستم مقادیر m و h را به‌صورت پویا تغییر می‌دهد تا پاسخ‌گوی نیاز همه کاربران باشد. درحالی‌که روش LA-SFDD در انجام این عمل به‌صورت مطلوب عمل نکرده و روش Linux نیز بدون توجه به وضعیت سیستم مقادیر m و h را در طول عمر سیستم به‌صورت ثابت در نظر می‌گیرد.



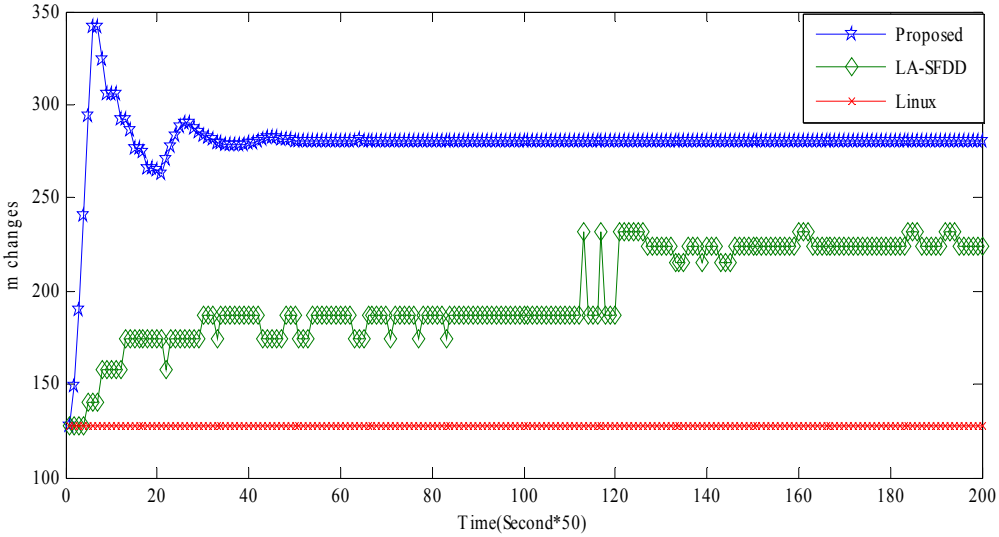

**شکل ۵: مقدار k با شدت حمله مختلف**




**شکل ۶: احتمال بلوکه شدن درخواست‌ها**




**شکل ۷: میزان تغییرات h**




**شکل ۸: میزان تغییرات m**

## ۶- نتیجه‌گیری

این مقاله، روشی را برای مقابله با حملات SYN flooding ارائه می‌کند. در این مقاله، به‌صورت ابتکاری این نوع حملات به یک مسئله بهینه‌سازی تبدیل شده و الگوریتم بهینه‌سازی اجتماع ذرات برای حل این مسئله ارائه می‌شود سپس یک الگوریتم فیلتر نوین برای حذف نویزهای ضربه‌ای معرفی شده و از ترکیب این الگوریتم‌ها برای به‌روز کردن دو پارامتر مهم و تأثیرگذار در حملات SYN flooding استفاده شده است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی، با توجه به عملکرد آگاهانه و هوشمند خود و تصمیم‌گیری‌های به‌موقع جهت تنظیم پارامترهای تأثیرگذار در عملکرد سیستم از نظر افزایش احتمال برقراری ارتباط عادی و همچنین کاهش احتمال موفقیت حمله، دارای بهبود قابل توجهی است.

## مراجع


[1] S.H.C. Haris, R.B. Ahmad, and M.A.H.A. Ghani, "Detecting TCP SYN flood attack based on anomaly detection," *2nd International Conference on Network Applications, Protocols and Services*, pp. 240-244, 2010.

[2] M. Korczynski, L. Janowski, and A. Duda, "An accurate sampling scheme for detecting SYN flooding attacks and port scans," *ICC*, pp. 1-5, 2011.

[3] N.B.I. Al-Dabagh, and I.A. Ali, "Design and implementation of artificial immune system for detecting flooding attacks," *High Performance Computing & Simulation-HPCS*, pp. 381-390, 2011.

[4] Bo Yang, and X. Wang, "Selection of parameter for SYN flood source-end detection," *Electronic & Mechanical Engineering and Information Technology*, pp. 106-109, 2011.

[5] S.G. Bhirud, and V. Katkar, "SYN flood attack prevention using main-memory database management system," *Internet (AH-ICI), 2nd Asian Himalayas*, pp. 1-6, 2011.

[6] L. Arshadi, and A. Jahangir, "Entropy based SYN flooding detection," *Local Computer Networks*, pp. 139-142, 2011.

[7] H. Safaa, A collaborative defense mechanism against SYN flooding attacks in IP networks," *Journal of Network and Computer Applications*, vol. 31, pp. 509–534, 2008.

[8] B. Xiao, W. Chen, and Y. He, "An autonomous defense against SYN flooding attacks: detect and throttle attacks at the victim side independently," *Journal of Parallel and Distributed Computing Archive*, vol. 68 no. 4, pp. 456-470, 2008.

[9] T. Kim, Y. Choi, J. Kim, and S. Je Hong, "Annulling SYN flooding attacks with whitelist," *Advanced Information Networking and Applications - Workshops*, 2008.

[10] Y.W. Chen, "Study on the prevention of SYN flooding by using traffic policing," *Network Operations and Management Symposium*, 2000.

[11] Q. Xiaofeng, H. Jihong, and C. Ming, "A mechanism to defend SYN flooding attack based on network measurement system," *Information Technology: Research and Education*, 2004.

[12] S. Khan, and I. Traore, "Queue-based analysis of DoS attacks," *Proc. Int. Conf. on IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, pp. 266–273, 2005.

[13] Y. Wang, C. Lin, Q. Li, and Y. Fang, "A queueing analysis for the denial of service (DoS) attacks in computer network," *Computer Networks*, vol. 51, pp. 3564–3573, 2007.

[14] S.A. Crosby, and Dan S. Wallach, "Denial of service via algorithmic complexity attacks," *USENIX Security Symposium*, 2003.

[15] S.H. Zahiria, and S.A. Seyedin, Swarm intelligence based classifiers," *Franklin Institute*, 2007.

[16] J. Kennedy, and R. Eberhart, "Particle swarm optimization," *IEEE Neural Networks*, 1995.

[17] R.C. Gonzalez, and R.E. Wood, *Digital Image Processing*. Prentice Hall, 2002.

[18] S. Sadeghi, *et al.*, "An efficient method for impulse noise reduction from images using fuzzy cellular automata," *Int. J. Electron. Commun.*, vol. 66, pp. 772–779, 2012.

[19] S. Tavassoli, A. Rezvanian, and M.M. Ebadzadeh, "A new method for impulse noise reduction from digital images based on adaptive neuro-fuzzy system and fuzzy wavelet shrinkage," *Proc. Int. Conf. on Computer Engineering and Technology (ICCET)*, vol. 4, pp. 297-301, 2010.

[20] A. Rezvanian, K. Faez, and F. Mahmoudi, "A two-pass method to impulse noise reduction from digital images based on neural networks," *Proc. Int. Conf. on Computer Engineering*, pp. 400–5, 2008.

[21] I. Turkmen, "Efficient impulse noise detection method with ANFIS for accurate image restoration," *Int. J. Electron Commun. (AEU)*, vol. 65, pp. 132–9, 2010.

[22] P.L. Rosin, "Image processing using 3-state cellular automata," *Comp Vision Image Unders*, Vol. 114, pp. 790–802, 2010.

[23] I. Kalaykov, and G. Tolt, "Real-time image noise cancellation based on fuzzy similarity," *Fuzzy Filters for Image Processing*, Springer, pp. 54–71, 2003.

[24] F. Farbiz, and M.B. Menhaj, "A fuzzy logic control based approach for image filtering," *Fuzzy Techniques in Image Processing*, Springer, pp. 194–221, 2000.

[25] F. Russo, "FIRE operators for image processing," *Fuzzy Sets Syst*, vol. 103, pp. 265–275, 1999.

[26] F. Russo, and G. Ramponi, "A fuzzy filter for images corrupted by impulse noise," *IEEE Signal Process Lett*, vol. 3, no. 6, pp. 168–70, 1996.

[27] A. Rezvanian, S. Rezvanian, and H. Khotanlou, "A new method to impulse noise reduction from medical images using cellular automata," *Proc. Int. Conf. on Electrical Engineering, ICEE*, vol. 8. pp. 53–8, 2009.

[28] M. Bekravi, S. Jamali, and G. Shaker, "Defense against SYN-Flood denial of service attacks based on learning automata," *International Journal of Computer Science*, vol. 9, no. 3, pp. 514-520, 2012.


### زیرنویس‌ها

[1] Fuzzy Cellular Automata

[2] Adaptive Neuro-Fuzzy System and Fuzzy Wavelet Shrinkage

[3] Neural Network Detector Adaptive Neuro-Fuzzy Inference System
[4] Neural Network Detector Median Filter
[5] Simple Neuro-Fuzzy Filter
[6] Fuzzy Similarity Filter
[7] Iterative Fuzzy Control based Filter
[8] Fuzzy Inference Ruled by Else-action
[9] Fuzzy Filter
[10] Traditional Cellular Automata with noise detection
[11] Loss Probability