

یک الگوریتم سبک وزن جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر بی‌سیم متحرک به کمک عامل‌های یادگیر

مجتبی جمشیدی^۱، سمیرا عباسی^۲، مهدی اثنی عشری^۳، محمدرضا میبیدی^۴

^۱آموزشکده فنی و حرفه‌ای سما، دانشگاه آزاد اسلامی، واحد کرمانشاه، کرمانشاه، ایران jamshidi.mojtaba@gmail.com

^۲مرکز آموزش علمی کاربردی میراث بیستون کرمانشاه، دانشگاه علمی کاربردی، کرمانشاه، ایران sco.iaun.ac.irsamiraabasi@

^۳پژوهشکده فناوری اطلاعات، پژوهشگاه فضای مجازی، تهران، ایران esnaashari@itrc.ac.ir

^۴دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران mmeybodi@aut.ac.ir

تاکنون الگوریتم‌های زیادی نظیر [11-3] جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر ثابت مطرح شده است. ولی این الگوریتم‌ها در شبکه‌های حسگر متحرک قابل بکارگیری نیستند، چراکه اکثر این الگوریتم‌ها یا متکی بر تعیین مکان گره‌ها و ارسال ادعاهای مکانی به گره‌های شاهد یا مکان‌های خاص در شبکه هستند، یا مختص توپولوژی‌های خاص (نظیر، گرید) می‌باشند. همچنین در [20-12] نیز الگوریتم‌هایی جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر متحرک ارائه شده است که به‌طور کلی دارای معایبی نظیر سربار ارتباطی و حافظه بالا، عدم مقیاس‌پذیری، مکانیزم پیچیده و سنگین، نیاز به تعیین مکان گره‌ها و استفاده از کلیدهای عمومی و امضاهای دیجیتال می‌باشند.

در این مقاله، یک الگوریتم جدید، هوشمند و سبک وزن مبتنی بر عامل‌های یادگیر و گره‌های نگهبان^۴ جهت شناسایی گره‌های کپی در شبکه‌های حسگر متحرک پیشنهاد می‌گردد، به‌طوری که معایب الگوریتم‌های موجود را برطرف کند. الگوریتم پیشنهادی نیاز به تعیین مکان گره‌ها، انتشار پیغام‌های ادعای مکانی، کلیدهای عمومی و فرایندهای پیچیده تشخیص گره‌های کپی ندارد.

ادامه این مقاله بدین ترتیب سازماندهی می‌شود. در بخش ۲، کارهای گذشته و در بخش ۳، عامل‌های یادگیر آمده است. فرضیات سیستم در بخش ۴ آمده است. در بخش ۵ الگوریتم پیشنهادی شرح داده می‌شود. ارزیابی کارایی و نتایج شبیه‌سازی در بخش ۶ ارائه شده است. بخش آخر نیز به نتیجه‌گیری می‌پردازد.

۲- کارهای گذشته

در [11-3] الگوریتم‌هایی جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر ثابت مطرح شده است. ولی این الگوریتم‌ها در شبکه‌های متحرک قابل بکارگیری نیستند. در ادامه به بررسی الگوریتم‌های خاص شبکه‌های متحرک می‌پردازیم.

در [12] یک الگوریتم توزیعی مبتنی بر تولید و معاوضه اعداد تصادفی جهت شناسایی گره‌های کپی مطرح شده است. ایده اصلی الگوریتم‌های ارائه شده در [13]، [14] و [15] برگرفته از این حقیقت است که یک گره متحرک ضابط نشده (قانونی) نباید هرگز در سرعتی بیش از حداکثر سرعت سیستم پیکربندی شده حرکت کند. وجود گره‌های کپی^۵، سبب می‌شود گره‌های دیگر گمان کنند این گره^۵ با سرعتی بیش از حداکثر سرعت از پیش تعریف شده حرکت می‌کند. ایده اصلی دو الگوریتم EDD و SEDD ارائه شده [16] برگرفته از این ملاحظه است که برای یک شبکه بدون گره تکراری، در یک دوره زمانی مشخص با طول T ، تعداد دفعات رویارویی گره^۵ با یک گره خاص^۶ به احتمال زیاد باید محدود

چکیده: تاکنون حمله‌های متنوعی بر روی شبکه‌های حسگر بی‌سیم تعریف شده است که یکی از خطرناک‌ترین آنها، حمله تکرار گره (یا گره کپی) است. در این حمله، دشمن یک (یا چند) گره نرمال درون شبکه را ضبط کرده، مواد قفل‌گذاری درون آن را استخراج نموده و کپی‌هایی از آن تولید و در شبکه تزریق می‌کند. در این مقاله، یک الگوریتم جدید، هوشمند و سبک‌وزن به کمک عامل‌های یادگیر و گره‌های نگهبان جهت شناسایی گره‌های کپی در شبکه‌های حسگر متحرک پیشنهاد می‌گردد. کارایی الگوریتم پیشنهادی از نظر سربار ارتباطات و حافظه با دیگر الگوریتم‌های موجود مقایسه گردیده که نتایج این مقایسه حاکی از برتری الگوریتم پیشنهادی است. همچنین، الگوریتم پیشنهادی توسط شبیه‌ساز JSIM پیاده‌سازی گردیده و با انجام یک‌سری آزمایش‌ها کارایی آن در قالب معیارهای احتمال تشخیص گره‌های کپی و احتمال تشخیص غلط ارزیابی شده است.

کلمات کلیدی: امنیت، عامل‌های یادگیر، گره‌های کپی، گره‌های نگهبان.

۱- مقدمه

یک شبکه حسگر بی‌سیم از مجموعه‌ای از گره‌های حسگر منابع محدود تشکیل شده است که با همکاری یکدیگر امکان نظارت بر محیط را فراهم می‌آورند. این نوع شبکه‌ها کاربردهای متنوعی در بخش‌های نظامی، صنعت، بهداشت و علوم دیگر دارند. گره‌های حسگر محدودیت‌های بسیاری از نظر ظرفیت حافظه، توان محاسباتی، برد رادیویی و میزان انرژی دارند. برقراری امنیت در این نوع شبکه‌ها امری بسیار مهم و چالش‌زا می‌باشد که توجه بسیاری از محققان را به خود جلب کرده است [1].

یکی از حمله‌های خطرناک در شبکه‌های حسگر بی‌سیم حمله تکرار گره^۱ یا گره کپی^۲ است. با توجه به گسترش بدون مراقبت گره‌ها در محیط عملیاتی، دشمن می‌تواند یک (یا چند) گره قانونی درون شبکه را ضبط و اطلاعات مهم از جمله مواد قفل‌گذاری^۳ داخل آن را استخراج کند و با استفاده از این مواد قفل‌گذاری، گره‌های تکراری (یا گره‌های کپی) ایجاد کند. گره‌های کپی دقیقاً حاوی مشخصات و اطلاعات (از جمله شناسه، مواد قفل‌گذاری و ...) گره قانونی ضبط شده می‌باشند، از این‌رو، قابلیت برپایی کلید با دیگر گره‌های قانونی شبکه را دارند. دشمن سپس می‌تواند این گره‌های کپی را در شبکه پخش کند و عملیات شبکه را مختل و یا شنود کند [2].

در این صورت، عامل یادگیر، یک عمل را به طور تصادفی و بر اساس بردار احتمال برای تکرار $k+1$ خود انتخاب می کند.

۴- فرضیات سیستم و مدل حمله

شبکه حسگر حاوی دو مجموعه گره های حسگر معمولی (SN) و گره های نگهبان (WN) است که به طور تصادفی در یک محیط دوبعدی پراکنده می شوند. تعداد گره های حسگر معمولی، $\eta = |SN|$ و تعداد گره های نگهبان، $\varpi = |WN|$ است. تعداد گره های نگهبان خیلی کمتر از تعداد حسگرهای معمولی است ($\eta \ll \varpi$). تعداد کل گره های شبکه را با n نشان می دهیم که در الگوریتم پیشنهادی از $n = \varpi + \eta$ بدست می آید. گره های حسگر معمولی، SN ، مأموریت شبکه (نظیر جمع آوری اطلاعات، ارسال داده ها به سمت ایستگاه پایه و ...) را انجام می دهند و گره های نگهبان، WN ، وظیفه شناسایی گره های کپی را بر عهده دارند. هر گره یک شناسه یکتا دارد و از موقعیت مکانی خود آگاه نیست. برد رادیویی تمام گره ها یکسان است. تمام گره ها متحرک می باشند و در طول حیات شبکه مطابق مدل های تحرک، نظیر Random waypoint در محیط عملیاتی مورد نظر حرکت می کنند. گره های حسگر معمولی (SN) در برابر مداخله مقاوم نیستند و دشمن در صورت ضبط یک گره می تواند به اطلاعات محرمانه آن دسترسی داشته باشد و آن را برنامه ریزی مجدد کند. ولی فرض می شود که گره های نگهبان در برابر مداخله مقاوم بوده و در صورت ضبط توسط دشمن، قابل کدگذاری و برنامه ریزی مجدد نمی باشند [25][24]. هم چنین، با توجه به متحرک بودن گره های حسگر در محیط عملیاتی، گره ها می بایست به طور پریودیک (به عنوان مثال بعد از هر t واحد زمانی یا پس از این که به یک مکان جدید در شبکه می رسند) یک پیغام "Hello"، درخواست مسیر، ارسال داده، زنده بودن^۱ و ... از خود منتشر کنند [26]. این عمل در واقع یکی از نیازمندی های شبکه های حسگر متحرک است تا هر گره بتواند در هر لحظه از زمان، همسایه های جاری خود را شناسایی کرده، در صورت نیاز با آن ها کلیدهای امنیتی برپا کند، با هم مخابره کنند و جدول مسیریابی خود را ایجاد کند. البته در این جا، گره های نگهبان از ارسال پریودیک این گونه پیغام ها خودداری می کنند تا حضورشان از دید دیگر گره ها مخفی بماند. چراکه این گره های نگهبان وظیفه شناسایی گره های بدخواه دشمن (گره های کپی) را دارند.

فرض می شود شبکه حسگر در یک محیط خصمانه گسترش می یابد، بنابراین، شبکه ناامن بوده و دشمن می تواند گره هایی را ضبط کند و کپی هایی از این گره های ضبط شده را ایجاد و سپس در شبکه تزریق کند. هم چنین، فرض می شود هر گره کپی نیز در هر دوره زمانی t ، یک پیغام "Hello"، درخواست مسیر، ارسال داده یا زنده بودن منتشر می کند. گره های کپی، می توانند هم چون گره های نرمال متحرک باشند و یا دشمن آنها را به طور ثابت در مکان های خاص مستقر نماید. هیچ یک از این دو حالت تأثیری بر الگوریتم پیشنهادی ندارد.

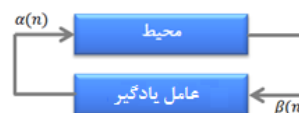
۵- الگوریتم پیشنهادی

ایده اصلی الگوریتم پیشنهادی، بکارگیری عامل های یادگیر و استفاده از پیغام های "Hello" (و درخواست مسیر، ارسال داده یا ...) منتشر شده توسط گره ها جهت شناسایی گره های کپی در شبکه های حسگر متحرک است. همان طور که گفته شد، در الگوریتم پیشنهادی علاوه بر گره های حسگر معمولی، تعدادی گره نگهبان در شبکه وجود دارد که ترافیک شبکه را نظارت کرده و گره های کپی را

باشد. در [17] یک الگوریتم جهت شناسایی گره های کپی در شبکه های سکتوربندی شده ارائه شده است. ایده اصلی الگوریتم [18]، استفاده از پروتکل پیش توزیع کلید جفتی مبتنی بر چند جمله ای و فیلترهای شمارشی Bloom [21] جهت شناسایی گره های کپی است. ایده اصلی الگوریتم [19]، SHD، مبادله لیست همسایه ها میان گره های متحرک و انتخاب گره های شاهد برای عمل تشخیص گره ها کپی است. در [20] نیز دو الگوریتم در حوزه زمان (TDD) و فضا (SDD) به کمک تابع درهم سازی تک-راهه رمزنگاری جهت مقابله با حمله گره های کپی مطرح شده است.

۳- عامل یادگیر

در این بخش، یک مدل از عامل های یادگیر ارائه می دهیم. مدل عامل یادگیر پیشنهادی، برگرفته از مدل اتوماتاهای یادگیر [22][23] است با این تفاوت کوچک که نحوه پاداش به عمل های درست، اندکی متفاوت است. عامل یادگیر یک ماشین با حالات محدود است که می تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده، توسط یک محیط تصادفی ارزیابی شده و پاسخی به عامل یادگیر داده می شود. عامل یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می کند. در طی این فرآیند، عامل یادگیر، یاد می گیرد که چگونه بهترین عمل را از بین اعمال مجاز خود انتخاب کند. شکل (۱) ارتباط بین عامل یادگیر و محیط را نشان می دهد.



شکل (۱) مدل عامل های یادگیر

محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی های محیط، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی های محیط و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال های جرمیه می باشند. ورودی محیط یکی از r عمل انتخاب شده عامل یادگیر است. خروجی (پاسخ) محیط به هر عمل i توسط β_i مشخص می شود. اگر β_i یک پاسخ دودویی باشد، محیط مدل P نامیده می شود. در چنین محیطی $\beta_i(n) = 1$ به عنوان پاسخ نامطلوب یا شکست و $\beta_i(n) = 0$ به عنوان پاسخ مطلوب یا موفقیت در نظر گرفته می شوند. به این ترتیب عامل یادگیر را می توان با چهارتایی $LA \equiv \{\alpha, \beta, p, T\}$ نشان داد که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل های اتوماتا (r تعداد عمل های عامل یادگیر)، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه ورودی های عامل یادگیر، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال عمل های عامل یادگیر و $T \equiv p(k+1) = T[\alpha(k), \beta(k), p(k)]$ الگوریتم یادگیری می باشد. اگر عامل یادگیر در تکرار k ام، یک عمل خود مانند α_i را انتخاب کند، تغییر احتمال عمل ها بصورت زیر خواهد بود (a پارامتر پاداش و b پارامتر جریمه):

الف- پاسخ مطلوب از محیط

$$\begin{aligned} p_i(k+1) &= p_i(k) + a[1 - p_i(k)] \\ p_j(k+1) &= (1-a)p_j(k) \quad \forall j, j \neq i \end{aligned} \quad (1)$$

در این صورت، عامل یادگیر، همین عمل α_i را برای تکرار $k+1$ انتخاب می کند. ب- پاسخ نامطلوب از محیط

$$\begin{aligned} p_i(k+1) &= (1-b)p_i(k) \\ p_j(k+1) &= \frac{b}{r-1} + (1-b)p_j(k) \quad \forall j, j \neq i \end{aligned} \quad (2)$$

به طور تصادفی برحسب بردار P_vector انتخاب می کند. این عملیات به طور همزمان توسط تمام گره های نگهبان انجام می گیرد. به این ترتیب، پس از گذشت یک بازه زمانی t ، دوره زمانی اول از اجرای فاز دوم الگوریتم پیشنهادی خاتمه می یابد. سپس گره ها یک مقصد تصادفی جدید برای خود انتخاب و شروع به حرکت به سوی مقصد می کنند. به این ترتیب، دور بعدی از فاز دوم آغاز می شود. همان طور که گفته شد، فاز دوم، به تعداد η دور اجرا می گردد. از آن جا که در حمله تکرار گره، دشمن یک گره (نظیر u) را ضبط و چندین کپی از آن را در شبکه منتشر می کند، لذا گره ی با شناسه u به تعداد دفعات بیشتری نسبت به گره های غیرکپی در همسایگی گره های نگهبان ظاهر می شود که این سبب می شود عامل های یادگیر احتمال عمل متناظر با این گره کپی را در بردار احتمالات (P_vector) افزایش دهند تا به مقدار 1 نزدیک شود.

۵-۲- فاز سوم (شناسایی گره های کپی)

پس از پایان اجرای فاز دوم الگوریتم پیشنهادی، هر گره نگهبان باید تصمیم بگیرد که آیا گره ی کپی ای شناسایی کرده است یا خیر. اگر هیچ گره کپی ای در شبکه وجود نداشته باشد، در این صورت با توجه به مدل تصافی حرکت گره ها، تعداد دفعات حضور همه گره ها در همسایگی هر گره نگهبان η تقریباً برابر خواهد بود که این سبب می شود مقدار احتمال ها در بردار P_vector تقریباً باهم برابر باشند. ولی اگر دشمن یک گره u را ضبط نموده و چندین کپی از آن در شبکه تزریق کرده باشد، در این صورت، تعداد دفعات ظاهر شدن گره با شناسه u در همسایگی گره های نگهبان بیشتر از سایر گره ها خواهد بود. این سبب می شود مقدار احتمال عمل متناظر با گره u در بردار P_vector عامل یادگیر گره های نگهبان به مراتب بیشتر از مقدار احتمال عمل های متناظر با سایر گره ها باشد. بنابر این، هر گره نگهبان باید با توجه به بردار احتمالات عمل های (P_vector) عامل یادگیر خود اقدام به شناسایی گره های کپی کند. در اینجا یک روال ساده جهت شناسایی گره های کپی، بر اساس بردار P_vector پیشنهاد می شود. شبه کد این روال در شکل (۲) آمده است.

```

1: evaluate  $\mu$ ,  $\mu = \frac{1}{\eta}$ 
2: sort  $P\_vector$  in Descend order
3: malicious_list =  $\phi$ 
4: for  $i = 0$  to  $\eta - 2$ 
    if ( $P\_vector[i] - P\_vector[i+1] > 2\mu$ )
        add Action[i] to malicious_list
    else
        exit for

```

شکل (۲) شبه کد روال شناسایی گره های کپی

در این روال، ابتدا میانگین (μ) بردار P_vector محاسبه می گردد. سپس، این بردار P_vector به صورت نزولی مرتب می شود. در آخر، با یک حلقه، بردار P_vector را پیمایش نموده و چنانچه اختلاف احتمال عمل i ام با احتمال عمل $i+1$ بیشتر از مقدار "دو برابر میانگین بردار P_vector " باشد، عمل متناظر با $P_vector[i]$ ، یعنی $Action[i]$ را به عنوان گره بدخواه کپی علامت می زند. ولی اگر شرط مورد نظر برای عنصر i ام برقرار نباشد، حلقه پیمایش شکسته شده و عناصر بزرگتر از i دیگر بررسی نمی شوند. چراکه عمل های متناظر با گره های کپی، میزان احتمال شان در بردار احتمالات خیلی بیشتر از احتمال دیگر عمل ها خواهد شد. از این رو، تفاضل مقدار احتمال عمل های متناظر با گره های کپی نسبت به عمل های متناظر با دیگر گره های غیرکپی خیلی بیشتر از دو برابر میانگین خواهد شد (بسته به تعداد دوره های فاز دوم و تعداد کپی ها).

شناسایی می کنند. بر روی هر یک از این گره های نگهبان، یک عامل یادگیر سوار می شود. از دلایل موثر بودن مدل عامل یادگیر پیشنهادی می توان به هوشمندی، سبک وزن و غیر قطعی بودن اشاره کرد که مناسب شبکه های حسگر بی سیم است. الگوریتم پیشنهادی از ۳ فاز تشکیل تشکیل شده است. در فاز اول، عامل های یادگیر پیکربندی می شوند. در فاز دوم، هر گره نگهبان با نظارت بر ترافیک شبکه، عامل یادگیر خود را بر روزرسانی می کند. این فاز خود به η دوره زمانی کوچک تر با طول زمان t شکسته می شود. در هر دوره زمانی t ، گره ها یک مقصد تصادفی برای خود انتخاب نموده و پس از رسیدن به مقصد در آنجا برای مدت زمانی ساکن می مانند و شروع به ارسال پیام های "Hello"، داده ای، درخواست مسیر و... می کنند. پس از آنکه η دور عمل نظارت بر ترافیک شبکه، توسط گره های نگهبان صورت پذیرفت و فاز دوم خاتمه یافت، فاز سوم آغاز می شود که در آن، اقدام به شناسایی گره های کپی می شود. در ادامه به شرح جزئیات این ۳ فاز از الگوریتم پیشنهادی می پردازیم.

۵-۱- فاز اول (پیکربندی عامل های یادگیر)

قبل از گسترش گره ها در محیط عملیاتی، عامل های یادگیر بر روی گره های نگهبان بار می شوند و با فرض این که تعداد گره های حسگر معمولی η باشد، بردار عمل (A_vector) و بردار احتمالات (P_vector) عمل های عامل های یادگیر به صورت رابطه (۳) تنظیم می شوند:

$$A_vector = [1, 2, \dots, \eta]$$

$$P_vector = [\frac{1}{\eta}, \frac{1}{\eta}, \dots, \frac{1}{\eta}]$$

(۳)

درواقع، هر گره حسگر معمولی، بیان گر یک عمل ($action$) برای عامل یادگیر است. سپس، هر عامل یادگیر به طور تصادفی یک عمل (α_i) را انتخاب می کند. عمل انتخاب شده α_i توسط عامل یادگیر موجود در گره نگهبان η در واقع گره ای است که η انتظار دارد در دوره بعدی نظارت بر ترافیک (در این جا، دوره اول از فاز دوم) آن را در همسایگی خود مشاهده کند. پس از انجام این مرحله، یعنی پیکربندی عامل های یادگیر موجود در گره های نگهبان، گره ها به طور تصادفی در محیط گسترش می یابند.

۵-۲- فاز دوم (نظارت بر ترافیک)

هر گره پس از ساکن شدن در یک مکان از شبکه، یک پیام "Hello" منتشر می کند تا خود را به همسایه هایش معرفی نماید و در صورت نیاز اقدام به ارسال پیام های درخواست مسیر، داده ای یا زنده بودن می کند. این سبب می شود هر گره نگهبان آگاه شود که چه گره هایی در حال حاضر (دوره فعلی از فاز دوم) همسایه آن هستند. پس از گذشت t واحد زمانی، یا به عبارت دیگر، پس از پایان دوره زمانی فعلی، عامل یادگیر هر گره نگهبان η پاسخی (مثبت یا منفی) از محیط دریافت می کند. اگر عمل (یا همان گره) انتخاب شده، یعنی α_i ، در همسایگی گره نگهبان η ظاهر شده باشد، این به منزله پاسخ مثبت از محیط است و چنانچه عمل α_i در همسایگی گره نگهبان η ظاهر نشده باشد، این به منزله پاسخ منفی از محیط است. اگر عامل یادگیر پاسخ مثبت از محیط دریافت کند، به عمل α_i مطابق رابطه (۱) پاداش داده و همین عمل α_i را برای دوره بعدی انتخاب می کند. ولی اگر پاسخ منفی از محیط دریافت کند، ابتدا عمل α_i را مطابق رابطه (۲) جریمه نموده، سپس یک عمل را مجدداً از مجموعه عمل های خود (A_vector)

۵-۴- تشخیص غلط در الگوریتم پیشنهادی

الگوریتم پیشنهادی در تشخیص گره‌های کپی ممکن است دچار اشتباه گردد. یعنی ممکن است برخی گره‌های غیرکپی را به اشتباه به عنوان گره‌های کپی شناسایی کند. حالت منجر به تشخیص غلط زمانی رخ می‌دهد که برخی گره‌های نرمال (نظیر u) به تصادف به تعداد دفعات زیادی در همسایگی یک گره نگهبان خاص (نظیر W) ظاهر شوند که این سبب می‌شود احتمال متناظر با عمل u در P_vector گره نگهبان W افزایش یابد (به سمت ۱ نزدیک شود). در نتیجه، در فاز سوم الگوریتم پیشنهادی، گره u به اشتباه توسط گره نگهبان W به عنوان گره کپی تشخیص داده می‌شود.

۶- ارزیابی کارایی و نتایج شبیه‌سازی

سربار حافظه: از آن‌جا که الگوریتم پیشنهادی فقط بر روی گره‌های نگهبان سوار است، لذا هیچ سربار حافظه‌ای بر گره‌های حسگر معمولی تحمیل نمی‌کند. ولی گره‌های نگهبان به یک فضای $O(\eta)$ جهت ذخیره لیست عمل‌های عامل یادگیر (یعنی A_vector) و یک فضای $O(\eta)$ جهت ذخیره بردار احتمال عمل‌های عامل یادگیر (یعنی P_vector) نیاز دارند. بنابر این، سربار حافظه مربوط به هر گره نگهبان از مرتبه $O(\eta)$ است درحالی که هیچ سربار حافظه‌ای به گره‌های حسگر معمولی تحمیل نمی‌شود. در جدول (۱) سربار حافظه الگوریتم پیشنهادی با دیگر الگوریتم‌ها موجود مقایسه شده است. همان‌طور که در جدول (۱) مشخص است، الگوریتم پیشنهادی از نظر سربار حافظه برتر از سایر الگوریتم‌ها می‌باشد. چراکه سربار حافظه الگوریتم پیشنهادی، به ازای η گره حسگر معمولی برابر صفر و به ازای ϖ گره نگهبان از مرتبه $O(\eta)$ است که باید توجه شود تعداد حسگرهای معمولی خیلی بیشتر از تعداد گره‌های نگهبان است.

جدول (۱) مقایسه سربار الگوریتم پیشنهادی با دیگر الگوریتم‌ها

| الگوریتم | سربار حافظه | سربار ارتباطات |
|-------------------|-----------------------------|-----------------------------|
| LSM[3] | $O(\sqrt{n})$ | $O(n\sqrt{n})$ |
| SET[6] | $O(n/T)$ | $O(n)$ |
| P-MPC, SDC [7] | $O(w)$ | $O(r\sqrt{n}) + O(s)$ |
| RED[4] | $O(d)$ | $O(n\sqrt{n})$ |
| RAWL[9] | $O(\log n \times \sqrt{n})$ | $O(\log n \times \sqrt{n})$ |
| XED[12] | $O(4 \times d \times E[X])$ | $O(1)$ |
| SPRT[15] | $O(n\sqrt{n})$ | $O(n)$ |
| EDD, SEDD[16] | $O(n), O(\xi)$ | $O(n)$ |
| Algorithm[18] | $O(d)$ | $O(n \times \log n)$ |
| TDD, SDD[20] | $O(n)$ | $O(\sqrt{n}), O(d)$ |
| الگوریتم پیشنهادی | $0 \sim O(\eta)$ | 0 |

n : تعداد گره‌ها در شبکه.
 w : تعداد گره‌های شاهده‌ای که ادعای مکانی از L را ذخیره می‌کنند.
 s : تعداد گره‌های حسگر در یک سلول از شبکه.
 $E[X]$: تعداد حرکت‌های مورد انتظار که یک گره s_i نیاز دارد تا دو نسخه متفاوت از گره‌های کپی را ملاقات کند.
 ξ : زیرمجموعه‌ای از گره‌های شبکه ($n < \xi$).
 i : به ازای حسگر معمولی صفر و به ازای گره‌های نگهبان $O(\eta)$
 نکته: در الگوریتم پیشنهادی سربارها (حافظه‌ای و ارتباطی) فقط به گره‌های نگهبان تحمیل می‌شوند ولی در سایر الگوریتم‌ها، همه گره‌ها متحمل سربار می‌شوند.

سربار ارتباطات: با توجه به محدودیت‌های انرژی گره‌های حسگر، میزان انرژی مصرفی الگوریتم‌های ارائه شده برای شبکه‌های حسگر یک موضوع مهم است. عملیات ارسال، دریافت و پردازش بسته از مجموعه اعمال مهمی هستند که انرژی مصرف می‌کنند. از آن‌جا که عمل ارسال بسته‌ها نسبت به عملیات پردازش و دریافت بسته‌ها انرژی خیلی بیشتری مصرف می‌کند، لذا محاسبه تعداد بسته‌های ارسالی که به دلیل استفاده از یک الگوریتم خاص به شبکه تحمیل می‌شود (یا همان سربار ارتباطات)، یک معیار مهم جهت ارزیابی کارایی الگوریتم‌های مطرح برای شبکه‌های حسگر است. از آن‌جا که الگوریتم پیشنهادی، فقط از پیغام‌های "Hello"، درخواست مسیر و ... جهت اجرای الگوریتم بهره می‌گیرد و با توجه به اینکه ارسال این دست پیغام‌ها جزء نیازمندی‌های شبکه‌های حسگر متحرک است، لذا هیچ‌گونه سربار ارتباطاتی به گره‌های حسگر معمولی تحمیل نمی‌شود. هم‌چنین گره‌های نگهبان در طی دوره‌های مختلف اجرای الگوریتم پیشنهادی هیچ‌گونه پیغامی در شبکه منتشر نمی‌کنند. از این‌رو، سربار ارتباطی گره‌های نگهبان نیز صفر است. در جدول (۱) هم‌چنین سربار ارتباطات الگوریتم پیشنهادی و سایر الگوریتم‌های موجود مقایسه گردیده است. از حیث سربار ارتباطات، الگوریتم پیشنهادی برتر از دیگر الگوریتم‌ها می‌باشد.

۶-۲- نتایج شبیه‌سازی‌ها

الگوریتم پیشنهادی توسط نرم‌افزار شبیه‌ساز JSIM پیاده‌سازی گردیده و با انجام یک‌سری آزمایش‌ها، کارایی آن در قالب معیارهای احتمال تشخیص و احتمال تشخیص غلط ارزیابی شده است.

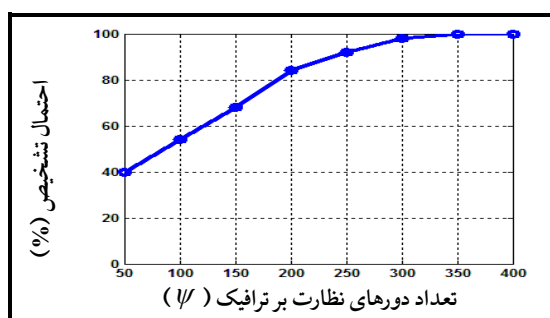
- **احتمال تشخیص:** تعیین می‌کند الگوریتم پیشنهادی با چه احتمالی موفق به شناسایی گره‌های می‌شود.

- **احتمال تشخیص غلط:** احتمال اینکه یک گره غیرکپی به اشتباه توسط یک الگوریتم امنیتی به عنوان گره کپی تشخیص داده شود.

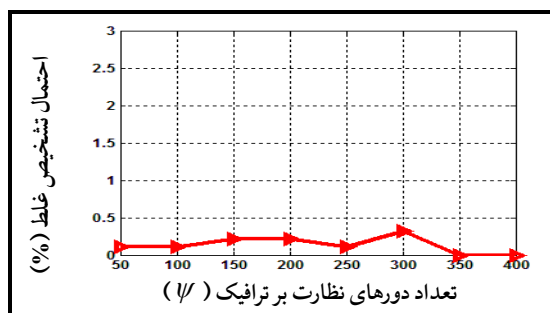
در اجرای شبیه‌سازی‌ها، فرض می‌شود شبکه حاوی n گره حسگر است که به‌طور تصادفی در یک ناحیه دایره‌ای 100×100 متر مربع پراکنده شده‌اند. دشمن M گره قانونی را ضبط نموده و از هر یک از آنها، R گره کپی ایجاد می‌کند. به عبارت دیگر، محیط عملیاتی حاوی $R \times M$ گره بدخواه می‌باشد. هم‌چنین، تعداد گره‌های نگهبان ϖ در نظر گرفته شده است. پارامترهای پاداش و جریمه عامل‌های یادگیر با مقادیر $a = 0.01$, $b = 0.001$ تنظیم شده است. برد رادیویی تمام گره‌ها نیز ۲۰ متر در نظر گرفته شده است. به منظور اطمینان از اعتبار نتایج، هر شبیه‌سازی ۱۰۰ بار تکرار شده و نتیجه نهایی از میانگین نتایج این ۱۰۰ تکرار بدست آمده است.

آزمایش ۱: در این آزمایش پارامترهای $n=100$, $M=5$, $R=10$, $\varpi=10$ تنظیم شده و احتمال تشخیص و احتمال تشخیص غلط گره‌های کپی در الگوریتم پیشنهادی، به ازای $\psi = 50, \dots, 400$ ، ارزیابی گردیده است. شکل‌های (۳) و (۴) نتایج این آزمایش را به ترتیب در قالب معیارهای احتمال تشخیص و احتمال تشخیص غلط نشان می‌دهند. نتایج این آزمایش در شکل (۳) نشان می‌دهد، احتمال تشخیص گره‌های کپی به ازای $\psi = 200$ برابر ۸۴٪، به ازای $\psi = 300$ برابر ۹۸٪ و به ازای $\psi \geq 350$ برابر ۱۰۰٪ است. هم‌چنین، نتایج این آزمایش در شکل (۴) نشان می‌دهد، احتمال تشخیص غلط الگوریتم پیشنهادی به ازای $\psi < 350$

هر گره ضبط شده در شبکه منتشر کند. پیش گرفتن این شیوه، برای دشمن ساده و امکان پذیر است. ولی الگوریتم‌هایی امنیتی در این حالت ممکن است سریع‌تر و دقیق‌تر اقدام به شناسایی گره‌های کپی کنند. نتیجه این آزمایش نیز نشان داد الگوریتم پیشنهادی به ازای R های بزرگتر، سریع‌تر گره‌های کپی را شناسایی می‌کند. چراکه، اگر تعداد کپی‌های زیادی از یک گره خاص، نظیر u ، در شبکه موجود باشد، گره‌های نگهبان تعداد دفعات بیشتری این گره u را ملاقات می‌کنند و در نتیجه احتمال متناظر با عمل این گره u در عامل‌های یادگیر سریع‌تر افزایش می‌یابد و به سمت 1 میل می‌کند. هم‌چنین، نتایج این آزمایش نشان داد، تغییر در پارامتر R ، تأثیر چندانی بر معیار احتمال تشخیص غلط الگوریتم پیشنهادی ندارد و نرخ این معیار تقریباً بین ۳٪ تا ۴٪ می‌باشد.



شکل (۳) تأثیر پارامتر n بر احتمال تشخیص الگوریتم پیشنهادی



شکل (۴) تأثیر پارامتر n بر احتمال تشخیص غلط الگوریتم پیشنهادی

جدول (۲) تأثیر پارامتر n بر کارایی الگوریتم پیشنهادی به ازای $\psi = 350$

| | $n=100$ | $n=200$ | $n=300$ |
|------------------|---------|---------|---------|
| احتمال تشخیص | 100% | 96% | 92% |
| احتمال تشخیص غلط | 0% | 0.9% | 1.4% |

جدول (۳) تأثیر پارامتر n بر کارایی الگوریتم پیشنهادی به ازای $\psi = 500$

| | $n=100$ | $n=200$ | $n=300$ |
|------------------|---------|---------|---------|
| احتمال تشخیص | 100% | 98% | 94.5% |
| احتمال تشخیص غلط | 0% | 0.5% | 1% |

جدول (۴) تأثیر پارامتر R بر کارایی الگوریتم پیشنهادی

| | $R=5$ | $R=10$ | $R=15$ |
|------------------|-------|--------|--------|
| احتمال تشخیص | 70% | 100% | 100% |
| احتمال تشخیص غلط | 0.38% | 0.37% | 0.32% |

آزمایش ۴: هدف این آزمایش، بررسی تأثیر تعداد گره‌های نگهبان، w ، بر کارایی الگوریتم پیشنهادی است. در این آزمایش پارامترهای $M=5$ ، $n=100$ ، $R=10$ ، $\psi=600$ احتمال تشخیص غلط گره‌های کپی در الگوریتم پیشنهادی را به ازای $w=2, \dots, 10$ ارزیابی نموده‌ایم. شکل‌های (۵) و (۶) نتایج این آزمایش را به

کمتر از ۵٪ و به ازای $\psi \geq 350$ برابر ۰٪ است. دلیل این نتایج واضح است. هرچه تعداد دوره‌های نظارت بر ترافیک (یعنی تعداد دوره‌های فاز دوم) افزایش یابد، گره‌های نگهبان دفعات بیشتری گره‌های کپی را در همسایگی خود ملاقات می‌کنند که این سبب می‌شود احتمال عمل‌های متناظر با این گره‌های کپی به سمت 1 و احتمال سایر عمل‌ها (گره‌های غیرکپی) به سمت 0 میل کند. در نتیجه، فاز سوم الگوریتم پیشنهادی، با دقت بیشتری گره‌های کپی را از گره‌های غیرکپی تمیز می‌کند. بنابر این، افزایش پارامتر ψ سبب می‌شود احتمال تشخیص به سمت ۱۰۰٪ و احتمال تشخیص غلط به سمت ۰٪ میل کند.

آزمایش ۲: هدف این آزمایش، ارزیابی تأثیر تعداد گره‌ها، n ، بر کارایی الگوریتم پیشنهادی است. در این آزمایش پارامترهای $w=10$ ، $M=5$ ، $R=10$ ، $\psi=350$ احتمال تشخیص غلط گره‌های کپی در الگوریتم پیشنهادی، به ازای $n=100, 200, 300$ ، ارزیابی گردیده و نتایج حاصل در جدول (۲) آمده است. نتایج این آزمایش نشان داد، با افزایش تعداد گره‌ها در شبکه، احتمال تشخیص گره‌های کپی کاهش و احتمال تشخیص غلط افزایش می‌یابد. دلیل این نتیجه این است که هرچه تعداد گره‌ها، n ، در شبکه افزایش یابد، به نسبت آن تعداد عمل‌های عامل‌های یادگیر نیز افزایش می‌یابد. از طرفی، هرچه تعداد عمل‌های عامل یادگیر افزایش یابد، سرعت همگرایی عامل یادگیر کندتر می‌شود. به عنوان مثال، زمانی که ۱۰۰ گره در شبکه وجود داشته باشد احتمال تشخیص و احتمال تشخیص غلط الگوریتم پیشنهادی به ترتیب ۱۰۰٪ و ۰٪ می‌شود. ولی زمانی که ۳۰۰ گره در شبکه وجود داشته باشد احتمال تشخیص و احتمال تشخیص غلط الگوریتم پیشنهادی به ترتیب ۹۲٪ و ۱/۴٪ است. البته اگر تعداد دوره‌های فاز دوم، یعنی ψ را افزایش دهیم ($\psi > 350$)، احتمال تشخیص گره‌های کپی بیشتر از ۹۲٪ می‌شود و به سمت ۱۰۰٪ میل می‌کند و احتمال تشخیص غلط کمتر از ۱/۴٪ و به سمت ۰٪ میل می‌کند. به منظور تصدیق این ادعا، همین آزمایش را به ازای $\psi=500$ تکرار نموده و نتایج حاصل را در جدول (۳) نشان داده‌ایم. می‌بینیم که با افزایش ψ ، کارایی الگوریتم بالاتر می‌رود.

آزمایش ۳: هدف این آزمایش، ارزیابی تأثیر تعداد کپی‌های منتشر شده از هر گره، یعنی R ، بر کارایی الگوریتم پیشنهادی است. در این آزمایش پارامترهای $w=10$ ، $M=5$ ، $n=100$ ، $\psi=300$ احتمال تشخیص غلط الگوریتم پیشنهادی، به ازای $R=5, 10, 15$ ، ارزیابی گردیده است. جدول (۴) نتایج حاصل از این آزمایش را نشان می‌دهد. نتیجه این آزمایش نشان می‌دهد با افزایش پارامتر R ، احتمال تشخیص گره‌های کپی نیز افزایش می‌یابد. در ادامه، دلیل این نتیجه را با یک نکته مهم شرح می‌دهیم:

دشمن به دو صورت می‌تواند با راه‌اندازی حمله گره‌های کپی، عملکرد شبکه را مختل کند. شیوه اول این است که دشمن تعداد زیادی گره قانونی درون شبکه را ضبط کرده و از روی هر کدام فقط تعداد اندکی (به عنوان مثال ۲ یا ۳ کپی) گره کپی ایجاد و در شبکه منتشر کند. در این حالت، الگوریتم‌های امنیتی به سختی و حتی ممکن است قادر به شناسایی گره‌های کپی نباشند. ولی پیش گرفتن این روش برای دشمن سخت و زمان‌بر است. چراکه باید گره‌های زیادی را ضبط، کدگذاری، برنامه‌ریزی مجدد و کنترل کند. شیوه دوم این است که دشمن تعداد گره‌های اندکی را ضبط کند ولی تعداد کپی‌های زیادی (به عنوان مثال ۱۰ کپی) از

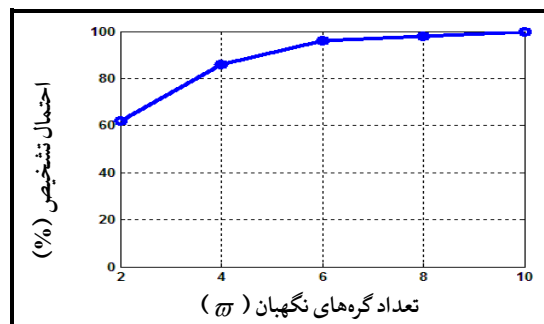


- [3] Parno B., Perrig A., and Gligor V. D., "Distributed Detection of Node Replication Attacks in Sensor Networks", in: Proc. of the IEEE Symposium on Security and Privacy, 2005.
- [4] Conti M., Pietro R. D., and Mancini L. V., "A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks", In: Proc. of the ACM MobiHoc, 2007.
- [5] Conti M., Pietro R. D., Mancini L. V., and Mei A., "Distributed Detection of Clone Attacks in Wireless Sensor Networks", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, 2010.
- [6] Choi H., Zhu S., and Porta T. F. La, "SET: Detecting Node Clones in Sensor Networks", in: Proc. of the SecureComm '07, pp. 341-350, 2007.
- [7] Zhu B. and et al., "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks", in: Proc. of the Annual Computer Security Applications Conference (ACSAC), 2007.
- [8] Kim C., Park C., Hur J., Lee H., and Yoon H., "A Distributed Deterministic and Resilient Replication Attack Detection Protocol in Wireless Sensor Networks", in: Proc. of the Communications in Computer and Information Science Volume 56, pp 405-412, 2009.
- [9] Zeng Y., Cao J., Zhang S., Guo S., and Xie L., "Random-Walk Based Approach to Detect Clone Attacks in Wireless Sensor Networks", In: Proc. of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 28, NO. 5, 2010.
- [10] Yu C.-M., Lu C.-S., Kuo S.-Y., "CSI: Compressed Sensing-Based Clone Identification in Sensor Networks", in: Proc. of the 8th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, 2012.
- [11] C. KIM, SHIN S., PARK C. and et al., "A Resilient and Efficient Replication Attack Detection Schema for Wireless Sensor Network", IEICE TRANS. INF. & SYST., VOL. E92-D, NO. 7, 2009.
- [12] Yu C. M., Lu C. S., and Kuo S. Y., "Mobile Sensor Network Resilient Against Node Replication Attacks" In: Proc. of the IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2008.
- [13] Ho J.-W., Wright M., and Das S., "Fast detection of replica node attacks in mobile sensor networks using sequential analysis", In: Proc. of the IEEE INFOCOM, pp. 1773 - 1781, 2009.
- [14] Ho J.-W., Wright M., and Das S., "Fast Detection of Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Hypothesis Testing", IEEE TRANS. ON MOBILE COMPUTING, VOL. 10, NO. 6, 2011.
- [15] Unnikrishnan D. and et al., "Detecting Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Probability Ratio Test", in: Proc. of the 13th International Conference on Distributed Computing and Networking (ICDCN), Hong Kong, China, January 3-6, 2012.
- [16] Yu C.-M., Lu C.-S., and Kuo S.-Y., "Efficient and Distributed Detection of Node Replication Attacks in Mobile Sensor Networks", In: Proc. of the IEEE Vehicular Technology Conf. Fall (VTC Fall), 2009.
- [17] Gowtham B., Sharmila S., "Location Traced Hybrid Detection of Node Replication Attack in Mobile Wireless Sensor Network", In: Proc. of the Special Issue of International Journal of Computer Applications (0975 - 8887) on Information Processing and Remote Computing - IPRC, August 2012.
- [18] Deng XM, Xiong Y., "A new protocol for the detection of node replication attacks in mobile wireless sensor networks", Journal of Computer Science and Technology 26(4), pp. 732-743, 2011.
- [19] Yxainaxniga Y. and et al., "Single Hop Detection of Node Clone Attacks in Mobile Wireless Sensor Networks", In: Proc. of the International Workshop on Information and Electronics Engineering (IWIEE), Vol. 29, 2012.
- [20] Xing K. an Cheng X., "From Time Domain to Space Domain: Detecting Replica Attacks in Mobile Ad Hoc Networks", In: Proc. of the IEEE INFOCOM, 2010.
- [21] Bloom B H., "Space/time trade-offs in hash coding with allowable errors", Commun. ACM, Vol. 13(7), pp. 422-426, 1970.
- [22] Narendra K. S. and Thathachar M. A. L., "Learning automata: An introduction", in: Proc. of the Prentice Hall, 1989.
- [23] Narendra K. S. and Thathachar M. A. L., "Learning automata a survey", IEEE Transactions on Systems, Man and Cybernetics, Vol. 4, no. 4, 1974.
- [24] Shi E., Perrig A., "Designing secure sensor networks", IEEE Wireless Communications, Vol. 11, pp. 38-43, 2004.
- [25] Tumrongwittayapak C. and Varakulsiripunth R., "Detecting Sinkhole Attacks in Wireless Sensor Networks", in: Proc. of the ICROS-SICE International Joint Conference, Fukuoka International Congress Center, Japan, 2009.
- [26] Piro C., Shields C. and Levine B. N., "Detecting the Sybil Attack in Mobile Ad hoc Networks", in: Proc. of the Securecomm and Workshops, 2006.

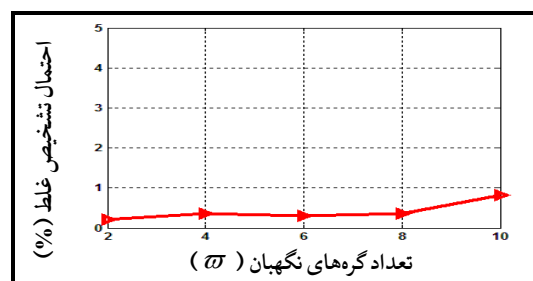
زیر نویس‌ها

- ¹ node replication attack
- ² Replica node
- ³ keying materials
- ⁴ Observer Nodes
- ⁵ P-Model
- ⁶ Sensor Nodes
- ⁷ Watchdog Nodes
- ⁸ Keep alive message

ترتیب در قالب احتمال تشخیص و احتمال تشخیص غلط نشان می‌دهند. نتایج این آزمایش نشان می‌دهد با افزایش تعداد گره‌های نگهبان در شبکه، احتمال تشخیص گره‌های کپی نیز افزایش می‌یابد. دلیل این نتیجه این است که با افزایش تعداد گره‌های نگهبان در شبکه، نواحی بیشتری از شبکه تحت پوشش نظارت این گره‌های نگهبان قرار می‌گیرد. از این رو، احتمال شناسایی تمامی گره‌های بدخواه در شبکه افزایش می‌یابد. از طرف دیگر، همان‌طور که در بخش ۵-۴ اشاره شد، این احتمال وجود دارد برخی گره‌های نرمال (نظیر u) به تصادف به تعداد دفعات زیادی در همسایگی یک گره نگهبان خاص (نظیر W) ظاهر شود که این سبب می‌شود احتمال متناظر با عمل گره u در P_vector گره نگهبان W افزایش یابد (به سمت ۱ نزدیک شود). در نتیجه، در فاز سوم الگوریتم پیشنهادی، گره u به اشتباه توسط گره نگهبان W به عنوان گره کپی تشخیص داده می‌شود. افزایش تعداد گره‌های نگهبان در شبکه، سبب افزایش حالت منجر به تشخیص غلط می‌شود. زیرا احتمال اینکه یک گره غیرکپی به تصادف و به دفعات زیادی در همسایگی یک گره نگهبان ظاهر شود، افزایش می‌یابد.



شکل (۶) تأثیر پارامتر W بر احتمال تشخیص الگوریتم پیشنهادی



شکل (۷) تأثیر پارامتر W بر احتمال تشخیص غلط الگوریتم پیشنهادی

۷- نتیجه‌گیری

در این مقاله، یک الگوریتم جدید، هوشمند و سبک‌وزن به کمک عامل‌های یادگیر جهت شناسایی گره‌های کپی در شبکه‌های حسگر متحرک ارائه گردید. الگوریتم پیشنهادی توسط شبیه‌ساز JSIM پیاده‌سازی گردید و نتایج آزمایش‌ها نشان داد، الگوریتم پیشنهادی کارایی مطلوبی در شناسایی گره‌های کپی دارد. مشکل الگوریتم پیشنهادی، سرعت کم در همگرایی عامل‌های یادگیر است (به خصوص هنگام زیاد شدن تعداد گره‌ها در شبکه). به کارگیری عامل‌های یادگیر سلولی می‌تواند یک راه‌کار موثر جهت برطرف نمودن این مشکل باشد که در کار بعدی خود به آن می‌پردازیم.

مراجع

- [1] Yick J., Mukherjee B. and Ghosal D., "Wireless sensor network survey", Computer Networks 52, pp. 2292-2330, 2008.
- [2] Karlof C. and Wagner D., "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", AdHoc Networks, 2003.