ISSN: 2745-3673
http://www.Rcsj.ir

مجله علمی پژوهش در علوم رایانه (سال هفتم)
شماره ۲۶ / تابستان ۱۴۰۱ / ص ۱۲- ۱

پیش‌بینی آسیب پذیری در برابر حملات در برنامه‌های کاربردی تحت وب

سمیه نصیری
کارشناسی ارشد مهندسی کامپیوتر گرایش نرم افزار، مدرس مدعو دانشگاه فرهنگیان پردیس الزهرا زنجان.

نام نویسنده مسئول:
سمیه نصیری

تاریخ دریافت: ۱۴۰۱/۰۳/۰۱
تاریخ پذیرش: ۱۴۰۱/۰۵/۰۵

چکیده

ساختن وب سایت ایمن برای برنامه نویسان وب وقت گیر، پرهزینه و چالش برانگیز است. محققان برای شناسایی سینک‌های صفحه وب از آنجا که به کاهش زمان و هزینه ایمن سازی برنامه وب کمک می‌کند، مدل‌های مختلف پیش‌بینی آسیب پذیری در برابر حملات وب را معرفی می‌کنند. همچنین روش‌های مختلف یادگیری ماشین توسط مدل‌های پیش بینی آسیب پذیری در برابر حملات موجود برای جلوگیری از مؤلفه‌های آسیب پذیر در برنامه‌های وب استفاده می‌شود. با این حال، اکثر این روش‌ها نمی‌توانند همه آسیب پذیری در برابر حملات‌های وب را به چالش بکشند. بنابراین، در این مقاله روشی با عنوان *NM-PREDICTOR* برای پیش‌بینی آسیب پذیری در برابر حملات در وب سایت‌ها به‌عنوان یک مشکل طبقه‌بندی شده با پیش تعیین کد معتبر یا آسیب پذیر پیشنهاد شده است. علاوه بر این، از طبقه بندی در طبقه‌های مختلف الگوریتم‌های یادگیری ماشین برای قضاوت در مورد حذف اجزای آسیب پذیر استفاده شده است. این روش برروی مجموعه داده‌های سه برنامه کاربردی وب ارزیابی شده ، که ۲۲۳ آسیب پذیری در برابر حملات با کیفیت عالی را که در *PHPMyAdmin*، *Moodle* و *Drupal* یافت می‌شود، ارائه می‌دهد. روش پیشنهادی نسبت به نتایج مطالعات موجود در مورد *Drupal*، *PHPMyAdmin* و *Moodle* مزیت بالایی دارد.

واژگان کلیدی: امنیت شبکه، حملات سایبری، آسیب پذیری در برابر حملات، برنامه‌های تحت وب، مجموعه داده‌ها.

مجله‌ی علمی پژوهش در علوم رایانه، شماره ۲۶، تابستان ۱۴۰۱، ص ۱۲- ۱

ISSN: 2745-3673
http://www.Rcsj.ir

مقدمه

برنامه‌های وب بهترین راه برای ارائه امکانات استاندارد از طریق اینترنت هستند. همکاری فن آوری های متنوعی که در بسیاری از لایه‌های تعمیم یافته استفاده می‌شود، دلیل اصلی آسیب پذیری در برابر حملات در برنامه‌های وب است. در حقیقت، تعداد آسیب پذیری در برابر حملات گزارش شده وب به سرعت در حال افزایش است. نقاط ضعف و اشکالاتی در وب سایت وجود دارد که می‌تواند توسط یک هکر مورد سوء استفاده قرار بگیرد نیز به عنوان آسیب پذیری در برابر حملات وب شناخته می‌شود. مطابق با پروژه امنیتی برنامه وب باز (OWASP)، مهم‌ترین آسیب پذیری در برابر حملات وب شامل XSS، CSRF و Injection SQL است.

برنامه‌های وب بر اساس ماهیت و عملکرد خود داده‌های حساس را هدایت و مدیریت می‌کنند و برای انجام فعالیت‌های مهم اقتصادی و کسب و کارهای مرتبط با آن مانند بانکداری، تأمین مالی آنلاین، خرید آنلاین و حساب‌های رسانه‌های اجتماعی به کار گرفته می‌شوند. امروزه، اکثر معاملات مالی و ارتباطات اجتماعی انجام شده توسط کاربر به برنامه‌های وب وابسته است. با این حال، آسیب پذیری در برابر حملات وب در ارتباط با برنامه وب فعالیت‌های کاربر را در این برنامه‌ها محدود می‌کند. این خطرات شامل تغییر مسیر کاربر به سایت‌های مخرب، درخواست‌های غیر قانونی HTTP، سرقت اطلاعات شخصی از طریق کوکی‌ها و جلسه‌ها، نصب بدافزارها و سایر فعالیت‌های غیرقانونی است. به منظور چیرگی کامل بر این مسائل، در سراسر جهان از مکانیزم های تست نفوذ مختلف با تکنیک‌های متنوعی استفاده می‌کند.

بهره برداری اطلاعاتی از آسیب پذیری در برابر حملات وب که برنامه‌های وب را تهدید می‌کند و تقاضای اضافی برای اقدامات متقابل امنیتی را تأیید و مورد تاکید قرار می‌دهد. روند آزمایش مکانیزم های تست و ردیابی آسیب‌ها دارای ریسک‌های بسیاری است زیرا بیشتر از طریق دستی انجام می‌شود و در نتیجه به دقت زیادی احتیاج دارد. بنابراین، برخی از رویکردهای دیگر برای غلبه بر موضوعات فوق الذکر مانند آزمایش جعبه سفید، آزمایش جعبه سیاه، برنامه نویسی ایمن، تجزیه و تحلیل ایستا، تجزیه و تحلیل پویا، آنالیز ترکیبی و یادگیری مکانیزم و عملکرد ماشین مورد نیاز است.

آزمایش جعبه سفید نوعی روش بررسی و تست است که در آن تستر به کد نرم افزاری دسترسی پیدا کرده و آن را تجزیه و تحلیل می‌کند. در این ازمایشات یک مسئله مثبت ولی کاذب در کد منبع وب سایت وجود دارد. به منظور پشتیبانی از محدوده عملکرد آزمایش کنندگان و غلبه بر تکنیک جعبه سفید، روش دیگری به نام تست جعبه سیاه وجود دارد. در این آزمایش کشف وب، آسیب پذیری در برابر حملات با مشاهده خروجی وب سایت در پاسخ به یک ورودی خاص مورد بررسی و به نسبت تجزیه و تحلیل انجام می‌گیرد.

محققان محدودیت‌های عمل پویش جعبه سیاه را در تشخیص آسیب پذیری در برابر حملات به طور مؤثر تحلیل وبه صورت کاملاً عملیاتی نمایش داده‌اند. از طرف دیگر، یک تکنیک بسیار مفید برای اطمینان از امنیت برنامه‌های وب، برنامه نویسی تحت وب ایمن نامیده می‌شود. این مکانیزم شامل شیوه‌هایی مانند رمزگذاری ورودی کاربران، بررسی و کاوش در انواع داده‌ها به همراه امکان پذیرش پرس و جوهای پارامتری شده است که توسعه دهندگان یک برنامه وب برای حفظ امنیت یک برنامه وب از آن‌ها بهره می‌گیرند.

مکانیزم ابزارهای تحلیل ایستا برای بازرسی از کد منبع باینری یا واسط استفاده می‌شود. در مقابل این ابزار تجزیه و تحلیل فازی و پویا کد برنامه وب را برای شناسایی آسیب پذیری در برابر حملات‌ها تجزیه و تحلیل نمی‌کند، اما در هنگام اجرا، داده‌های تزریق شده را تأیید می‌کند. تجزیه و تحلیل ترکیبی از هر دو مکانیزم پویا و ایستا را برای جلوگیری از آسیب پذیری در برابر حملات وب استفاده‌ای کامل خواهد داشت. در یادگیری عملکرد ماشین هم چنین برای تشخیص انواع آسیب پذیری در برابر حملات وب طیف گسترده‌ای از برنامه‌های وب مورد استفاده قرار می‌گیرد. با این حال، می‌تواند برای شناسایی آسیب پذیری در برابر حملات وب در کد منبع با شماره طبقه بندی نیز استفاده شود. روش‌های بی شماری برای تشخیص آسیب پذیری در برابر حملات وب بر اساس یادگیری عملکرد و مکانیزم ماشین ارائه شده است.

در اینجا، یک روش ترکیبی دو لایه به نام NM-PREDICTOR را برای پیش بینی آسیب پذیری در برابر حملات برنامه‌های کاربردی تحت وب پیشنهاد می‌شود. برای بهبود دقت پیش بینی، ویژگی‌های متنی و معیارهای نرم افزاری به طور

ISSN: 2745-3673
http://www.Rcsj.ir

موازی با هم در نظر گرفته و چندین مدل پیش بینی را با یک دیگر ترکیب می‌کند. در مرحله اول، مکانیزم NM-PREDICTOR 6 مدل پیش بینی مختلف را از یک مجموعه ساخت یافته آموزشی ایجاد می‌کند، که توسط معیارهای نرم افزاری و ویژگی‌های متنی آنها ارائه شده است. این مدل‌ها دارای برچسب آسیب پذیری در برابر حملات یا غیرقابل توصیف هستند.

با توجه به یک روش جدید برای پیش بینی (به عنوان آسیب پذیری در برابر حملات یا غیرقابل انکار)، هر یک از شش مدل اصلی به تنهایی قابلیت پیش زمینه احتمال این امر را از آسیب پذیری در برابر حملات پیش بینی می‌کنند. در ردیف دوم، VUL-PREDICTOR مدل پیش بینی دیگری (بر اساس نتایج پیش بینی شده از شش دسته طبقه بندی شده) را می‌سازد. هدف اصلی از این گفتمان ارائه یک الگوریتم ترکیبی برای NM-PREDICTOR جهت پیش بینی تأثیرپذیرتری در ارزیابی نتایج مربوط به مجموعه داده‌های برنامه‌های کاربردی تحت وب است.

بررسی پیشینه و ادبیات

مطالعات بسیاری در مورد پیش بینی آسیب پذیری در برابر حملات وب وجود دارد که درآن ها مدل‌ها و ابزارهای مختلفی برای پیش بینی آسیب پذیری در برابر حملات در پروژه‌های نرم افزاری ارائه شده است.

در مطالعه‌ای جامع توسط Neuhaus و همکارانش ابزار حریص پیشنهاد شده است و این ابزار به طور خودکار آسیب پذیری در برابر حملات موجود در واحدهای بایگانی در هر پایگاه داده ا بررسی می‌کند. ابزار حریص از اطلاعات مکانیزم معدن برای ثبت آسیب پذیری در برابر حملات گذشته از قطعات استفاده می‌کند. علاوه بر این، اجزای شناسایی شده با توجه به نوع آسیب پذیرترین افراد در هر دسته طبقه بندی می‌شوند.

وانگ و همکارانش [۱] مکانیزم خوشه بندی چگالی سریع موسوم به DSVRDC را مورد مطالعه قرار داد و با استفاده از آن روش جدیدی برای ایجاد آسیب پذیری در برابر حملات وب در نظر گرفته‌اند. سفارشات آسیب پذیری در برابر حملات کشف شده با خوشه بندی وابسته به چگالی به ترتیب چیده شده و بررسی دسته‌های طبقه بندی شده توسط تفاوت در نظم S- و روش خوشه بندی چگالی مبتنی بر افت Rd.

شولت و همکاران [۲] ترکیبی از مکانیزم یادگیری ماشین و آنالیز ایستا و ایجاد IPAAS، برای محافظت از تزریق SQL و آسیب پذیری در برابر حملات به صورت متقابل سایت به وجود آوردند. در این روش پیشنهادی مرزهای دسته بندی داده‌ها و همچنین مقادیر محدود ورودی از کد منبع و درخواست HTTP استخراج می‌شود.

در مطالعه دیگری توسط ویجیاپاسکارا و همکاران. [۳]، روش استخراج متن را ارائه کردند تا آسیب پذیری در برابر حملات احتمالی در مجموعه داده‌های آسیب پذیری در برابر حملات عمومی را استخراج کند. این روش یک ماتریس term-document ایجاد می‌کند. علاوه بر این، استراتژی کار جمع آوری داده‌های گزارش شده از بانک اطلاعاتی مبنی بر اشکالات کلی است تا آن را به یک بردار فردی تبدیل کند که بعداً کلمات را در قالب ابتدایی تا حد امکان کوتاه می‌کند.

یک مطالعه مهم دیگر، شار و تان [۴] ابزاری موسوم به PHPMINER-1 را برای شناسایی آسیب پذیری در برابر حملات وب و طبقه بندی روشهای مختلف تصفیه ورودی در کلاس‌های مختلف به عنوان مجموعه‌ای از کد ایستا را پیشنهاد کردند. بهره گرفتن از این ابزار بر استفاده و شناسایی آسیب پذیری در برابر حملات وب با روش کاوش داده‌ها استوار است.

در مطالعه‌ای دیگری توسط شار، تان و برایان [۵] روش ثانویه‌ای را برای تخمین آسیب پذیری در برابر حملات با استفاده از ویژگی‌های پویا با ویژگی‌های ایستا تکمیل کردند. علاوه بر این، آن‌ها از نقشه‌های یادگیری و سنجش نظارت شده برای طبقه بندی استفاده کرده‌اند.

در مطالعه دیگر با استفاده از روش یادگیری ماشین، هوارد و همکارانش [۶] سیستم Psigene را پیشنهاد داد که ویژگی‌هایی را از یک محدوده بزرگ که برای جمع آوری حمله SQL Injection استفاده می‌شود، بازیابی می‌کند تا نحوه توصیف آنها را بررسی کند. سپس برای شناسایی هر یک از این حملات آن‌ها را علامت گذاری می‌کند.

ISSN: 2745-3673
http://www.Rcsj.ir

در بخش قبلی توسط سینگ و همکارانش اشاره شد که روشی را برای شناسایی XSS، SQL injection، RCE و / LFI RFI توسعه داده‌اند. این مطالعه اثری را در جهت بهبود تمامی پارامترهای عملکرد برای افزایش دقت در جلوگیری از آسیب پذیری در برابر حملاتی از قبیل فراخوان، میزان هشدار کاذب و دقت ارائه داده است.

در یک مطالعه دیگری که اخیراً توسط Grieco و همکاران وی انجام شد [۷] روشی را برای جلوگیری از آسیب پذیری در برابر حملات وب توسط تکنیک‌های تبدیل زیر ساخت‌ها به آرایه فازی و استخراج ویژگی‌های متضاد دخیره سازی در حافظه را پیشنهاد کرد. آن‌ها از V-DISCOVER برای پیش بینی مانیتورینگ و نظارت بر عملکرد پویا برای برنامه‌های مختلف استفاده کرده‌اند. علاوه بر این برخی از آسیب پذیری در برابر حملاتی که منجر به نشت حافظه می‌شود را تأیید می‌کند.

در مجموعه مطالعه دیگری که Medeiros و همکارانش انجام دادند. روش جدیدی را برای جلوگیری از ساختار پایه و متن کد منبع توسط الگوریتم استخراج برای شناسایی آسیب پذیری در برابر حملات وب پیشنهاد کرد.

در یک مطالعه مهم دیگر در مورد یادگیری ماشین والدن، استاکمن و اسکانداریاتو [۸]، دو روش شناسایی آسیب پذیری در برابر حملات مؤثر معیارهای نرم افزار و تکنیک‌های استخراج متن را با یکدیگر مقایسه کردند.

در یک تحقیق دیگر، ابونادی و ممدوح [۹] روش تحقیقات تجربی را بررسی کرده‌اند که اثر پیش بینی پروژه متقاطع را برای پیش بینی آسیب پذیری در برابر حملات نرم افزار بررسی می‌کند. مطالعه نویسنده به دسته بندی‌های بدست آمده با روش‌های مختلف تضعیف ماشین بستگی دارد و از آنها برای تقویت تشخیص تجزیه شونده‌ها استفاده می‌کند. بررسی دقیق تحقیق در مورد روش یادگیری ماشینی برای شناسایی آسیب پذیری در برابر حملات وب و محدوده متمرکز در جدول ۱ ارائه شده است.

جدول ۱. مطالعات انجام شده یادگیری ماشین در پیش بینی آسیب پذیری در برابر حملات وب

Research article	Language/ model	Year	Dataset	Classifier	Web vulnerabilities	Performance parameters	Application
Neuhaus et al. [۱۰]	Vulture tool	2007	134 Mozilla vulnerabilities	SVM	Security vulnerabilities	Precision, recall	Mozilla firefox
Wang et al. [۱]	DSVRDC	2011	Open source web server software Apache httpd 2.2.8	Rd-entropy	Security vulnerabilities	Accuracy	C++ programming language
Wijayasekara et al. [۳]	Open bug database	2012	Linux kernel vulnerabilities (Redhat Bugzilla)	Bayesian	SQLi	Accuracy	Hidden impact bugs
Shar and Tan [۴]	PHPMINER-1	2012	Java-based open source applications	Proposed	XSS, SQL	Accuracy	HTML/JavaScript and PHP
Howard et al. [۶]	Psigene system frameork	2014	The exploit database, PacketStorm Security	Logistic regression	SQL injection	Accuracy, Precision	PHP

روش

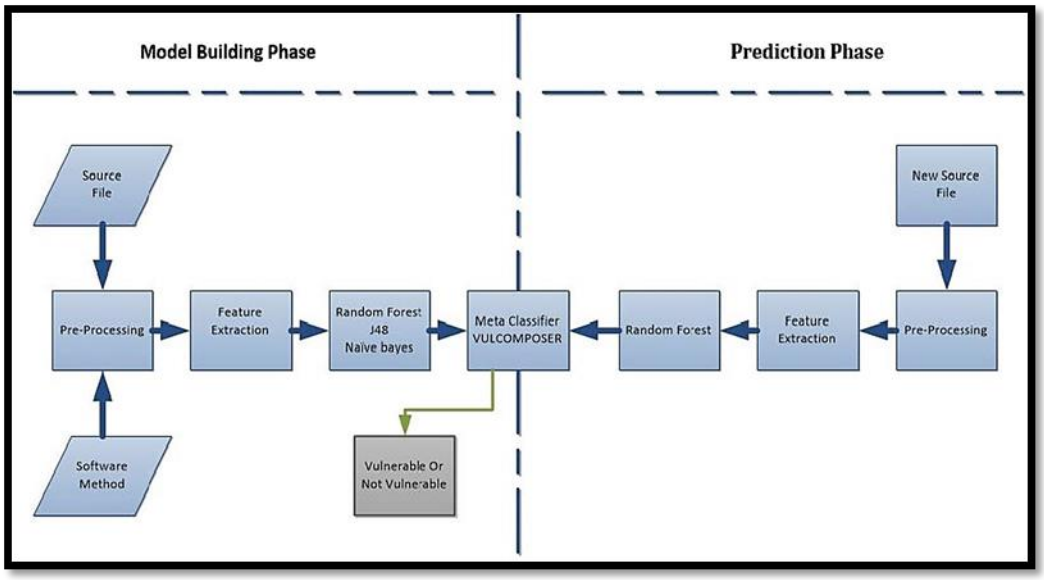
در این بخش، روش پیشنهادی NM-PREDICTOR و مجموعه داده‌ها موازی آن برای ارزیابی الگوریتم و کارآیی آن ارائه داده می‌شود. علاوه بر این، ما طبقه بندی‌های اصلی را با استفاده از NMPRE- DICTOR و نحوه ترکیب این طبقه بندی‌ها در طبقه بندی متا به تفصیل بیان می‌کنیم.

روش پیشنهادی (NMPREIDCTOR) چارچوبی برای جلوگیری از آسیب پذیری در برابر حملات وب مبتنی بر یادگیری ماشین است. ما فرض می‌کنیم که این فرایند پیش بینی به عنوان یک طبقه بندی با توجه به آسیب پذیری در برابر حملات

ISSN: 2745-3673
http://www.Rcsj.ir

پیش بینی شده در یک متن PHP خاص پیش‌بینی می‌شود. شکل ۱ NM-PREDICTOR کلی را با دو مرحله متفاوت ارائه می‌دهد. مرحله اول، مرحله ساخت مدل و مرحله دوم پیش بینی است.

در مرحله ساخت مدل، ابتدا باید یک مدل از منبع آموزشی که نظارت بر یادگیری به عنوان آسیب پذیری در برابر حملات شناخته شده است را ایجاد کرد. در مرحله پیش بینی، از مدل برای پیش بینی اینکه آیا کد منبع جدید آسیب پذیر است یا خیر استفاده می‌شود. در مرحله اول،NM-PREDICTOR تمامی شش طبقه مختلف را بر روی یک مجموعه آموزشی از افراد دارای برچسب که توسط معیارهای نرم افزاری و ویژگی‌های متنی آنها ساخته شده است ارائه می‌دهد. در سطح دوم،-NM PREDICTOR یک طبقه بندی متا (فوق طبقه) ایجاد می‌کند، که شش طبقه اصلی را با هم ترکیب می‌کند.



شکل ۱. نمودار مفهومی از NM-PREDICTOR

برای ساخت VULCOMPOSER، امتیازات حاصل از بررسی دستیابی‌ها توسط ۶ طبقه اصلی برای هر نمونه در مجموعه آموزشی، برای ایجاد یک مجموعه داده جدید جمع آوری می‌شوند. علاوه بر آن، این مجموعه داده برای آموزش VULCOMPOSER با اجرای برنامه یادگیری ماشین به عنوان جنگل تصادفی دسته بندی استفاده می‌شود. برای محاسبه شش امتیاز ثابت، شش طبقه توسط NM-PREDICTOR استفاده می‌شود. اولی این است که برچسب عنوان جدید را پیش بینی کنید و بعد، امتیازات نتیجه گیری از این شش طبقه خواهد بود که به عنوان ورودی برای VULCOMPOSER استفاده می‌شود تا از امتیازات نهایی حاصل، نتیجه نهایی جدید آسیب پذیر باشد. پس از آن، مجموعه ویژگی‌های مشتق شده از عملکرد انواع مختلف الگوریتم‌ها تجزیه و تحلیل می‌شود. سرانجام، پیشنهادی در رابطه با رویکرد ترکیب چندین طبقه ارائه می‌شود. این مجموعه آزمایشی شباهت نزدیکی با همان طرح پیشنهادی توسط مطالعات موجود دارد [۹ و ۱۱].

بانک اطلاعاتی

مجموعه داده‌های (Dataset) مورد استفاده در آزمایشات همان مجموعه داده استفاده شده توسط Welden و همکارانش است که جمع آوری و تجزیه و تحلیل شده است. [۸] و مجموعه داده آسیب پذیری در برابر حملاتPHP Security نامیده شده است [۸]. این مجموعه داده براساس ابلاغیه‌های امنیتی از پایگاه داده ملی آسیب پذیری در برابر حملات، تعداد آسیب پذیری در برابر حملات‌ها را در هر مرحله مشخص می‌کند. این مجموعه داده از سه برنامه وب ۲۲۳ آسیب پذیری در برابر حملات با کیفیت عالی را که درPHPMyAdmin ، Moodle و Drupal یافت می‌شود، ارائه می‌دهد. در صورت آسیب پذیر بودن یا نبودن هر مجموعه داده، مجموعه‌ای از اطلاعات مربوط به آن کلاس را شامل می‌شود.

ISSN: 2745-3673
http://www.Rcsj.ir

از میان ۲۳۳ آسیب پذیری در برابر حملات، ۱۹ مورد از آنها نفوذ کد است که به مهاجمان اجازه می‌دهد به طور تصادفی متغیرهای سمت سرور و هدرهای HTTP را تغییر دهند، ۱۲ مورد از آنها آسیب پذیری در برابر حملات CSRF هستند که به HTML مخرب خارجی اجازه می‌دهند Session های خود را ربوده و برای انجام اقدامات غیرمنطقی اقدام کنند.

اسکریپت‌های متقابل سایت (XSS) آسیب پذیری در برابر حملات وب ۸۶ مورد هستند که با JavaScript های مخرب می‌توانند در مرورگر کاربر تأثیر بگذارند. ۱۴ مورد از آنها در گروه آسیب پذیری در برابر حملات افشای مسیر قرار دارند که به شما امکان می‌دهد مسیر برنامه را بدست آورید. ۷۳ مورد از آنها هم به عنوان موضوعات در ارتباط با مجوزهای دسترسی از جمله اطلاعات افشای، دور زدن امتیاز و آسیب پذیری در برابر حملات مربوط به رمزگذاری گمشده یا نامناسب اجرا شده است طبقه بندی می‌شوند.

۱۹ مورد باقیمانده از آسیب پذیری در برابر حملات متفاوتی هستند که مربوط به حمله به افراد در میانه فیشینگ و سایر بردارهای حمله نا مشخص است. جدول ۲ شامل توضیحات کلی مختصر مربوط به سه کاربرد است که شامل تعداد کل موارد، تعداد کمتر در معرض آسیب پذیری در برابر حملات، میزان P (درصد افراد آسیب پذیر) و تعداد کامل ویژگی‌های متن است. برنامه Moodle یک برنامه کاربردی مهم است که دارای نرخ مثبت بسیار کمتری است. این کیفیت پیش بینی موارد نادر آسیب پذیر را دشوار می‌کند.

جدول ۲. تمام نمرات آسیب پذیری در برابر حملات وب در مجموعه داده

Application	Vulnerable files	Total files	P-rate (%)
Drupal	62	202	30.68%
PHPMyAdmin	27	322	8.39%
Moodle	24	2942	0.82%

ارزیابی الگوریتم

برای آموزش و آزمایش مجموعه داده برای ویژگی‌های نرم افزاری و متن، سه الگوریتم مانند RF، NB و J48 انتخاب شده‌اند. دلیل انتخاب این الگوریتم‌ها، استراتژی مختلف آموزشی است. علاوه براین، کلیه موارد فوق برای کشف مکانیسم آزمایش، آموزش و یادگیری است [۸، ۹، ۱۱و ۱۲]. تنظیم پارامتر برای الگوریتم‌های یادگیری ماشین از مقادیر پارامتر پیش فرض در Weka استفاده می‌کند. برای J48 اجرای الگوریتم درخت تصمیم C4.5 است و ضریب اطمینان را در تبادل اطلاعات ۰.۲۵ تعیین می‌کنیم. برای جنگل تصادفی مقدار درختان ایجاد شده را در ۱۰۰ تنظیم کنید.

حداکثر حد برای رشد درخت وجود ندارد. برای Bayes ساده، از discretization تحت نظارت استفاده می‌کند که در ادامه متغیر را به متغیرهای مجزا یا عادی کاذب تغییر می‌دهد. برای بهینه سازی وضعیت طبقه بندی انجام شده توسط مجریان طرح، الگوریتم‌های متکی بر ماشین‌های مختلف در WEKA با استفاده از مدل‌های متا ترکیب می‌شوند. در این مطالعه، ما نتیجه را با یک طبقه بندی واحد ارزیابی می‌کنیم، که برای افزایش صحت طبقه بندی‌های مختلف چندان مؤثر نبوده است، زیرا هیچ طبقه بندی واحدی به بهترین دقت و طبقه بندی مثبت کاذب نمی‌پردازد [۱۱].

چگونه می‌توان نتایج را ارزیابی کرد؟

این نتایج با توجه به وجود داده‌های آسیب پذیری در برابر حملات وب ثبت می‌شود. برای دستیابی به یک نتیجه بر اساس پارامترهای مختلف یادگیری ماشین مانند دقت، فراخوان و اندازه گیری-f. اندازه گیری دقیق تعداد موارد آسیب پذیر توسط یک مدل آسیب پذیر صورت می‌پذیرد. معیارهای عملکردی اتخاذ شده در این مطالعه الهام گرفته از کار [۸، ۹، ۱۱و ۱۲] هستند. این اقدامات به شرح زیر است:

ISSN: 2745-3673
http://www.Rcsj.ir

ماتریس درهم روشی برای نشان دادن چگونگی درهم بودن کلاس در هنگام پیش بینی است. مقدار دقیق مشخص شده به عنوانTP (مثبت درست) نامگذاری می‌شود و ارزش مثبت طبقه بندی نشده یا غلط به عنوان FP (مثبت نادرست) توصیف می‌شود.

دقت در اندازه گیری همان نمونه‌هایی است که به درستی طبقه بندی شده‌اند. این نشان می‌دهد که مقدار پیش بینی شده با مقدار واقعی چقدر نزدیک است. دقت (accuracy) را می‌توان با روش ارائه شده اندازه گیری کرد.

$$Accuracy = (TP + TN) / TP + FP + FN + TN \quad (۱)$$

دقت نمونه‌هایی که به درستی طبقه بندی شده‌اند را ارزیابی می‌کند. این گزینه مجموعه اطلاعاتی را در مورد میزان محکم بودن پیش بینی را ارائه می‌دهد. دقت را می‌توان با فرمول مطابق شکل زیر محاسبه کرد:

$$Precision = TP / (TP + FP) \quad (۲)$$

بیاد بیاورید تعداد نمونه‌های مثبت است که با دقت طبقه بندی مثبت می‌شوند. همچنین به عنوان حساسیت شناخته می‌شود. می‌توان آن را محاسبه کرد:

$$Recall = TP / (TP + FN) \quad (۳)$$

نمره F1 یک معیار عملکرد است که هر دو دقت را با هم ترکیب می‌کند و با هم فراخوانی می‌کند.

$$Score = (2 * Precision * Recall) / (Precision + Recall) \quad (۴)$$

نتیجه و ارزیابی تجربی

در این بخش، NM-PREDICTOR را با مستندات مطالعات موجود ارزیابی می‌کنیم. این آزمایش بر روی هسته Intel (R) coreTM i5 در حال اجرا در سرعت CPU از ۳.۴۰ گیگاهرتز انجام شده است. سیستم عامل نسخه Microsoft Windows 10 است. برای نتیجه آزمایش از WEKA برای اجرای برنامه‌های مختلف مانند Drupal، PHPMyAdmin و Moodle استفاده می‌شود. برای اعتبارسنجی NM-PREDICTOR و کاهش تعصب انتخاب مجموعه تمرین، اعتبار سنجی متقابل ۱۰ پیچ را در WEKA انجام دهید. اعتبار سنجی متقابل یک ارزیابی استاندارد عملکرد آسیب پذیری در برابر حملات است، که به طور گسترده برای ارزیابی مطالعات مهندسی نرم افزار ازگذشته مورد استفاده قرار می‌گیرد [۱۳-۱۱].

اجزا مربوط به یک برنامه به طور تصادفی به ۱۰ پیچ با اندازه برابر تقسیم می‌شوند. هر قسمت همان درصد از آسیب پذیری در برابر حملات‌ها را با کل نسخه (طبقه بندی استراتژیک) دارد. از این ۱۰ پیچ، ۹ مورد برای آموزش کلاس درس استفاده می‌شود، در حالی که از یک پیچ باقی مانده برای آزمایش اثربخشی کلاس استفاده می‌شود.

در مرحله قبل از پردازش، کد منبع با نشانه گذاری در آنها، با از بین بردن فضای هرز، متوقف کردن کلمات و نشانه گذاری‌ها را حذف می‌کند. برای مطالعه پیشنهادی، اسامی و کلمات شناسایی را در کامنت ها قرار دهید و نامهای شناسه را در زیر علامت پوشش بالایی با استفاده از بیان منظم آن‌ها در ویژگی صوتی یادگیری و بدون نظارت به سمت بردار Weka منحرف نمایید.

در فرآیند حذف کلمه توقف که مرتباً در حال حذف هستند، کلماتی ظاهر می‌گردند که کمکی به تفکیک یک فایل از دیگری می‌کنند و فهرستی از کلمات متوقف شده را که از گلوله برفی موجود است، استفاده می‌کنند. چندین روش برای اداره کلاس متعادل وجود دارد. برای NM-PREDICTOR از الگوریتم تکنیک پراکندگی اقلیت مصنوعی (SMOTE) استفاده شده

ISSN: 2745-3673
http://www.Rcsj.ir

است. این کلاس با ایجاد موارد تخصصی به جای تعویض آنها، کلاس کوچکتری ایجاد می‌کند. در این مطالعه، مقدار "k" برابر ۵ است. فیلتر Randomize برای تغییر خودسرانه ترتیب عبور به وسیله موارد استفاده می‌شود.

مولد عدد تصادفی هر زمان که مجموعه جدیدی از موارد به آن منتقل شد با مقدار دانه تنظیم می‌شود. ما در آزمایش‌های مختلف ارائه شده توسط WEKA از نمونه گیری، تصادفی وSMOTE employed استفاده کردیم و از این فیلترها برای افزایش دقت NM-PREDICTOR بهره برده می‌شود.

در انتخاب خاصیت‌ها، همه ویژگی‌های برای سنجش نرم افزار و ویژگی متن انتخاب شده است. ویژگی‌های متن نشانه‌هایی است که در مراحل پیش پردازش و فرکانس‌های مرتبط با آنها استخراج می‌شود. ویژگی اندازه گیری نرم افزار شامل یک خط کد، یک خط کد غیرHTML، میزان عملکرد، مشکلات مختلف لانه سازی، عارضه سیکلومتری، اندازه Halstead، تعداد تماسهای خارجی، Fan-in، Fan-out، توابع داخلی به نام، توابع خارجی است. موارد تماس گرفته شده و تماس‌های خارجی به توابع می‌باشد.

جدول ۳. نتیجه عملکرد NMPRPREDCITOR

Dataset	Classifier	Generated experimental results		
Performance parameters		Precision	Recall	F1-score
Drupal	Random forest	0.849	0.851	0.848
	J48	0.828	0.832	0.825
PHPMyAdmin	Random forest	0.532	0.410	0.463
	J48	0.445	0.426	0.435
Moodle	Random forest	0.221	0.138	0.169
	J48	0.249	0.171	0.202

برای این مطالعه، از J48 و جنگل تصادفی برای ساخت طبقه بندی استفاده کرده‌ایم. علاوه بر این، اجرای این طبقه بندی از پارامترهای پیش فرض در WEKA استفاده می‌کند. نتیجه برای NM-PREDICTOR نشان می‌دهد که کلیه دسته‌های استفاده شده در اندازه گیری-F همانطور که در جدول ۳ آورده شده است دقت بالایی را ارائه می‌دهد. برای محاسبه شش نمره قابل اعتماد در نتایج اولیه آزمایش اعتبار سنجی متقابل نتایجی حاصل می‌شود تا دو روش معادن متن و معیارهای نرم افزار در PHPMyAdmin،Drupal و Moodle با هم مقایسه شوند. این نرم افزار پشتیبانی بیشتری برای بینش قابل توجهی در مورد Fو اندازه گیری F در مورد Drupal ۸۴/۸٪، در موردPHPMyAdmin ۵۹/۶٪ و در مورد Moodle ۴۴/۳٪ فراهم کرده است. به نظر می‌رسد که اطلاعات ما به وسیله اندازه گیری F بالاتر از اثبات [۸، ۹و ۱۱] بوده و پشتیبانی می‌شوند.

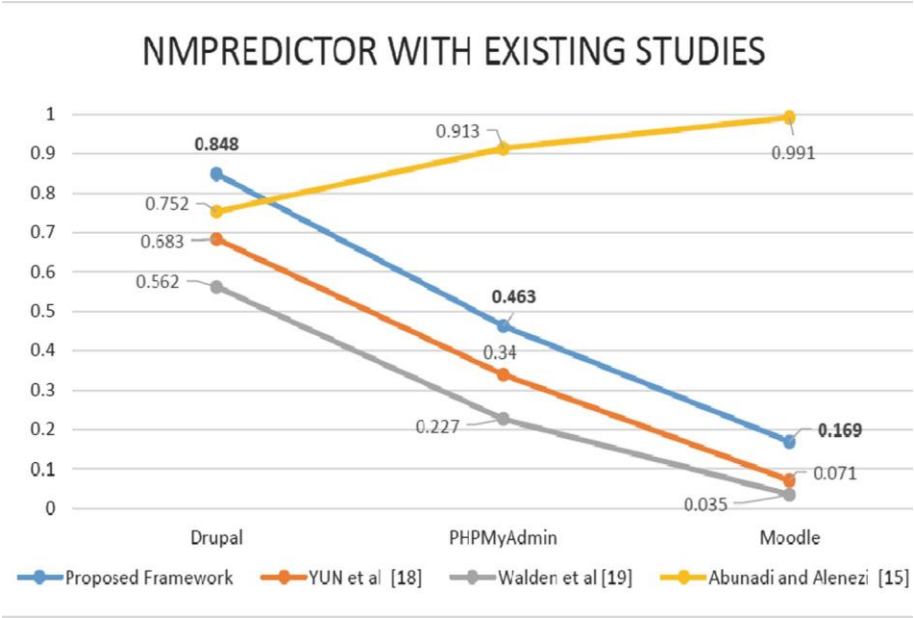
جدول ۴. جدول نمایش مطالعات موجود

Dataset	Classifier	Yun et al. [۱۱]			Abunadi and Alenezi [۹]			Walden et al. [۸]		
Performance parameters		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Drupal	RF	0.672	0.694	0.683	0.747	0.757	0.752	0.473	0.694	0.562
PHP MyAdmin	RF	0.346	0.330	0.340	0.905	0.922	0.913	0.164	0.370	0.227
Moodle	RF	0.250	0.042	0.071	0.987	0.995	0.991	0.018	0.704	0.035

ISSN: 2745-3673
http://www.Rcsj.ir

نتیجه مطالعه موجود برای ژانگ و همکاران [۱۱] ابونادی و آلنزی [۹] والدن و همکارانش [۸] نشان داده شده در جدول ۴ برای والدن و همکاران [۸] مطالعه موجود، دو روش مختلف متن کاوی و معیارهای نرم افزاری را با آزمایش‌های اعتبار سنجی متقابل بر روی سه کاربرد مختلف Drupal، PHPMyAdmin و Moodle مقایسه کرده است.

آن‌ها نتیجه را با دو شاخص اصلی عملکرد کلیدی مانند فراخوانی و بازرسی نشان می‌دهند. علاوه بر این، مقدار پارامتر عملکرد اندازه گیری-F در مورد Drupal برای متن کاوی به ترتیب برای PHPMyAdmin و Moodle به ۵۶/۲٪ است. در مطالعه موجود ابونادی و آلنزی [۹] ادعا کردند که این نتیجه امتیاز بسیار بالایی دارد، اما او نتوانست مستندات لازم برای اثبات کافی در مورد این نتیجه ارائه دهد، زیرا آنها برای ایجاد تعادل یک کلاس از هیچ ترکیبی استفاده نکرده‌اند. برای مطالعه آنها ارزش F1 در مورد Drupal ۷۵/۲٪، در موردPHPMyAdmin ۹۱/۳٪ و در مورد Moodle ۹۹/۱٪ است. برای ژانگ و همکاران [۱۱] رویکردهای نتیجه آزمایشی که به ترتیب تنها از معیارهای نرم افزار دارای ویژگیهای متنی هستند. از جدول، نتیجه موجود برای ۳ VUL-PREDICTOR مربوط به ۳ مجموعه داده، به عنوان امتیازی از امتیاز-F1 برابر ۶۸/۳٪، ۳۴٪ و ۷/۱٪ است. تلاش‌های زیادی انجام شده است تا نتیجه هدف گذاری شده در آن با افزایش دقت بر اساس یادگیری ماشین حاصل گردد.



شکل ۲. نتیجه فریم ورک با مطالعات موجود بر روی جنگل تصادفی

در شکل ۲ نتایج دقت فراخوانی F1 نتیجه تحقیق شده را که به خوبی در کار ارائه شده است، توصیف می‌کند. نتیجه روش پیشنهادی با نتایج موجود ژانگ و همکاران [۱۱]، والدن و همکاران [۸] و ابونادی و آلنزی [۹] مقایسه می‌شود. نتیجه والدن و همکاران. [۸] از آزمون اعتبارسنجی متقابل حاصل شد تا روش‌های متنی و نرم افزاری دو روش در Drupal، PHPMyAdminو Moodle را به هم مربوط کنیم.

اندازه گیری F در نمونه Drupal برای استخراج متن ۶۳/۱٪ بیشتر است. برای مطالعه موجود، ابونادی و آلنزی [۹] ادعا کردند که این نتیجه نمره بالایی است، اما نتوانست مستندات کافی از این نتیجه ارائه دهد زیرا این تحقیق از هیچ نتیجه‌ای برای ایجاد تعادل کلاس استفاده نمی‌کند. برای مطالعه آنها، F1 نمرات مجموعه داده Drupal را در شرایط RF بدست آورده است.

یعنی ۷۵/۲٪، در موردPHPMyAdmin ۹۱/۳٪ است، در صورتی که در Drupal ۹۹/۱ ٪ می‌باشد. برای یون و همکاران روی کردهای نتیجه آزمایشی که به ترتیب فقط از معیارهای نرم افزار دارای ویژگی‌های متنی ساخته می‌شوند امتیاز F1 ۶۸/۳ ٪ برای Drupal، و ۳۴٪ و ۷/۱٪ به ترتیب برای PHPMyAdmin و Moodle است. برای این مطالعه، پشتیبانی بیشتری برای بینش قابل توجه در مورد اندازه گیری F در مورد Drupal ۸۴/۸٪، در مورد PHPMyAdmin 56.9% و در مورد Moodle ۴۴/۳٪ است.

ISSN: 2745-3673
http://www.Rcsj.ir

روش پیشنهادی ما نشان می‌دهد که یک مزیت نسبت به نتیجه ژانگ و همکارانش [۱۱] دارد. ابونادی و آلنزی [۹] و والدن و همکاران [۸] در مورد Drupal برای PHPMyAdmin وMoodle نتیجه ما خیلی بهتر از ژانگ و همکاران [۱۱]. و والدن و همکاران. [۸] بوده اما تحقیقات ابونادی و آلنزی [۹] از هیچ نتیجه‌ای برای تعادل یک کلاس استفاده نکردند.

تعهد به رد تهدیدات

تعدادی از تهدیدها در مورد صحت مطالعه ما وجود دارد، که در زیر می‌پردازیم.

اعتبار سازه: ما با استفاده از یک مجموعه داده در دسترس عموم که در کار قبلی استفاده شده است، نگرانی‌های مربوط به اعتبار سازه را کاهش دادیم. این مجموعه داده شامل برنامه‌های واقعی Android و برچسب‌های آسیب پذیری در برابر حملات پرونده‌ها در آن برنامه‌ها است. مجموعه داده اصلی متأسفانه حاوی پرونده‌های منبع نبود. با این حال، ما برای بازیابی پرونده‌های منبع مربوط از مخزن کد آن برنامه‌ها، از اطلاعات (به عنوان مثال جزئیات برنامه، شماره نسخه و تاریخ) ارائه شده با مجموعه داده با دقت استفاده کرده‌ایم.

ما سعی کردیم با استفاده از اقدامات عملکرد استاندارد برای پیش بینی آسیب پذیری در برابر حملات، تهدیدهای مربوط به اعتبار هم جوشی را به حداقل برسانیم. اما ما اذعان داریم که تعدادی از آزمونهای آماری را می‌توان برای تأیید اهمیت آماری نتیجه گیری‌های ما به کار برد. اگرچه ما نتوانسته‌ایم از آن آزمایش‌های آماری در کارهای قبلی در پیش بینی آسیب پذیری در برابر حملات استفاده شود، اما ما قصد داریم این تحقیقات را در کارهای بعدی خود انجام دهیم.

مجموعه داده‌ای که ما استفاده کردیم حاوی برچسب‌های آسیب پذیری در برابر حملات فقط برای پرونده‌های منبع جاوا است. در عمل، سایر پرونده‌ها (به عنوان مثال پرونده‌های آشکار) XMLممکن است حاوی اطلاعات امنیتی مانند حقوق دسترسی باشند. یک تهدید دیگر مربوط به پیش بینی نسخه متقابل است که ما در آن آزمایش انجام شده را در تکرار کردیم و اجازه دادیم که دقیقاً همان پرونده‌ها بین نسخه‌ها وجود داشته باشد. این ممکن است نتایج را تحت الشعاع قرار دهد، اما تمام مدل‌های پیش بینی که در آزمایش خود با آنها مقایسه کرده‌ایم از این مزیت بهره مند می‌شوند.

تعداد زیادی برنامه کاربردی را در نظر گرفتیم که از نظر اندازه، پیچیدگی، دامنه، محبوبیت و تاریخچه تجدید نظر متفاوت است. با این حال ما تصدیق می‌کنیم که مجموعه داده‌های ما ممکن است نماینده انواع برنامه‌های Android نباشد. تحقیقات بیشتر برای تأیید یافته‌های ما برای سایر برنامه‌های Android و همچنین سایر برنامه‌های کاربردی مانند برنامه‌های وب و برنامه‌های نوشته شده به زبان‌های برنامه نویسی دیگر مانند PhP و سی + +.

نتیجه گیری

در این مقاله، ما روش NM-PREDICTOR را پیشنهاد می‌کنیم تا آسیب پذیری در برابر حملات‌ها را در برنامه‌های وب پیش بینی کرده و پیش بینی آسیب پذیری در برابر حملات را به عنوان یک مشکل طبقه بندی با از پیش تعریف کردن اینکه اگر یک PHP یک فرد آسیب پذیر باشد، تهیه کنیم. این روش یک الگوریتم پیش بینی است که به عنوان مثال دو لایه برچسب را آسیب پذیر یا غیرقابل آسیب می‌داند. در اولین مرتبه، NM-PREDICTOR شش طبقه مختلف را در یک مجموعه آموزشی از برچسب‌ها ایجاد می‌کند که توسط معیارهای نرم افزاری و ویژگی‌های متنی آنها ساخته شده است. در سطح دوم، NM-PREDICTOR یک طبقه بندی متا ایجاد می‌کند، که شش طبقه کلاس را تشکیل می‌دهد. مجموعه داده‌های سه برنامه وب ۲۲۳ آسیب پذیری در برابر حملات با کیفیت برتر را که در PHPMyAdmin، Moodleو Drupal یافت می‌شود، ارائه می‌دهد. نتایج مطالعه ما نشان می‌دهد که روش پیشنهادی برای دستیابی به امتیاز بهتر اندازه گیری F در مقایسه با ژانگ و همکاران [۱۱]، ابونادی و آلنزی [۹] و والدن و همکاران [۸] به خوبی عمل کرده است. در آینده، ما قصد داریم علاوه بر سه پروژه در نظر گرفته شده در این کار، در مورد آسیب پذیری در برابر حملات بیشتر از پروژه‌های اضافی نیز آزمایش کنیم. علاوه بر این، به منظور بهبود کاراییNM-PREDICTOR بیشتر، یک جهت بررسی ویژگی‌های اضافی است که می‌توان علاوه بر معیارهای نرم افزاری و ویژگی‌های متنی که در این کار استفاده می‌شود، بهره بگیریم. علاوه بر این، ما قصد داریم تا به بررسی تأثیر کدها [۱۴و

ISSN: 2745-3673
http://www.Rcsj.ir

۱۵] بپردازیم تا از آن به عنوان ویژگی استفاده شود و ویژگی‌های کامپوزیتی دیگری را نیز ایجاد کند که می‌تواند اثربخشی یک الگوریتم طبقه بندی را بهبود بخشد.

ISSN: 2745-3673
http://www.Rcsj.ir

منابع و مراجع

[1] T. Scholte, W. Robertson, D. Balzarotti, and E. Kirda, “Preventing input validation vulnerabilities in web applications through automated type analysis”, In: 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC), pp. 233-243. 2012.

[2] D.Wijayasekara, M. Manic, J. L. Wright, and M. McQueen, “Mining bug databases for unidentified software vulnerabilities”, In: 2012 5th International Conference on Human System Interactions (HSI), pp. 89-96, 2012.

[3] L.K. Shar, and H.B.K. Tan, “Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities”, In: Proceedings of the 34th International Conference on Software Engineering, pp. 1293-1296, 2012.

[4] L.K. Shar, and H.B.K. Tan, “Predicting common web application vulnerabilities from input validation and sanitization code patterns”, In: 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 310-313, 2012.

[5] G.M. Howard, C.N. Gutierrez, F.A. Arshad, S. Bagchi, and Y. Qi, “pSigene: webcrawling to generalize SQL injection signatures”, In: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 45-56, 2014.

[6] G. Grieco, G.L. Grinblat, L. Uzal, S. Rawat, J. Feist, and L. Mounier, “Toward large-scale vulnerability discovery using machine learning”, In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pp. 85-96, 2016.

[7] D. Taibi, A. Janes, and V. Lenarduzzi, “How developers perceive smells in source code: a replicated study”, Inf. Softw. Technol, 92, 223-235, 2017.

[8] S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller, “Predicting vulnerable software components”, In Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 529-540, 2007.

[9] Medeiros, N. Neves, and M. Correia, “Detecting and removing web application vulnerabilities with static analysis and data mining”, IEEE Trans. Reliab, Vol. 65, No. 1, pp. 54-69, 2016.

[10] Y. Wang, Y. Wang, and J. Ren, “Software vulnerabilities detection using rapid density-based clustering”, J. Inf. Comput. Sci., Vol. 8, No. 14, pp. 3295-3302, 2011.

[11] J. Walden, J. Stuckman, and R. Scandariato, “Predicting vulnerable components: software metrics vs text mining”, In: 2014 IEEE 25th International Symposium on Software Reliability Engineering (ISSRE), pp. 23-33, 2014.

[12] S. Gupta, and B.B. Gupta, “Cross-site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art”, Int. J. Syst. Assur. Eng. Manag., Vol. 8, No. 1, pp. 512-530, 2017.

[13] Y. Zhang, D. Lo, X. Xia, B. Xu, J. Sun, and S. Li, “Combining software metrics and text features for vulnerable file prediction”, In: 2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 40-49, 2015.

[14] F. Palomba, G. Bavota, M. Di Penta, F. Fasano, R. Oliveto, A. De Lucia, “A large-scale empirical study on the lifecycle of code smell co-occurrences”, Inf. Softw. Technol, 99, 1-10, 2018.

[15] Web Vulnerability Scanners: A Case Study, Angel Rajan, Emre Erturk, Eastern Institute of Technology, Hawke’s Bay.