Journal of Modeling in Engineering

Journal homepage: https://modelling.semnan.ac.ir/

ISSN: 2783-2538

Semnan University



Research Article

Vulnerable Smart Contracts Detection on the Ethereum Blockchain Using an Effective Risk Estimation Metric

UNCORRECTED PROOF

PAPER INFO

Paper history:

Received: 2022-03-13
Revised: 2022-05-12
Accepted: 2022-10-19

Keywords:

Blockchain,

Smart contracts,

Risk estimation,

Linear discriminant analysis,

Nearest neighbor,

Detection rate.

ABSTRACT

Identifying vulnerable smart contracts has a direct impact on blockchain security because it helps users avoid using these contracts. Calculating the risk level is preferable to accurately identifying these types of contracts using classification models because these models have classification errors. In addition, the data required to achieve high accuracy may not be available. Therefore, with the help of a vulnerability risk estimation criterion for smart contracts, users can be helped in decision-making. In this research, the issue of vulnerability risk in smart contracts is introduced. In addition, an effective criterion for its estimation is devised. In this criterion, linear discriminant analysis of smart contracts and distances to their nearest neighbors is exploited to estimate the risk of an unknown smart contract. Although deep learning is not used in the proposed criterion and it requires little training data, it provides a realistic risk estimate. Experiments conducted on a real dataset of Ethereum blockchain smart contracts, including vulnerable and secure contracts, demonstrate the effectiveness of the proposed criterion. Furthermore, the performance of the proposed measure in term of detection rate, accuracy, recall and F1-score is superior to existing risk estimation metrics in other areas, such as apps and URLs.

DOI: https://doi.org/10.22075/jme.2024.31282.2495

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

*** Corresponding author.**
E-mail address:

How to cite this article:

مقاله پژوهشی

شناسایی قراردادهای هوشمند آسیب پذیر در بلاک چین اتریوم به کمک یک معیار مؤثر برای برآورد خطر آنها

اطلاعات مقاله

دریافت مقاله: ۱۳۹۵/۰۳/۲۰
بازنگری مقاله: ۱۳۹۵/۰۶/۲۰
پذیرش مقاله: ۱۳۹۵/۰۷/۱۵

واژگان کلیدی:
بلاکچین،
قراردادهای هوشمند،
برآورد ریسک،
تحلیل تفکیکی خطی،
نزدیک ترین همسایه،
نرخ تشخیص.

چکیده

شناسایی قراردادهای هوشمند آسیب پذیر تأثیر مستقیمی بر امنیت بلاک چین دارد زیرا به کاربران کمک می‌کند از استفاده از این قراردادها اجتناب کنند. محاسبه میزان ریسک نسبت به شناسایی دقیق این نوع قراردادها با استفاده از مدل‌های طبقه بندی، برتری دارد زیرا این مدل‌ها دارای خطاهای طبقه بندی هستند. علاوه بر این، داده‌های لازم برای دستیابی به دقت بالا ممکن است در دسترس نباشند. بنابراین، با کمک معیار برآورد ریسک آسیب پذیری برای قراردادهای هوشمند، می‌توان به کاربران در تصمیم گیری کمک کرد. در این پژوهش، مسئله ریسک آسیب پذیری در قراردادهای هوشمند معرفی شده است. علاوه بر این، یک معیار موثری برای برآورد آن، ابداع شده است. در این معیار، برای تخمین ریسک یک قرارداد هوشمند ناشناخته، از تحلیل تفکیکی خطی قراردادهای هوشمند و فواصل تا نزدیکترین همسایگان آنها بهره برداری می شود. اگرچه یادگیری عمیق در معیار پیشنهادی استفاده نمی‌شود و به داده‌های آموزشی کمی نیاز دارد، اما تخمین ریسک واقع بینانه ای را ارائه می‌دهد. آزمایش‌های انجام‌شده بر روی مجموعه داده‌های واقعی از قراردادهای هوشمند بلاک چین اتریوم، شامل قراردادهای آسیب‌پذیر و ایمن، اثربخشی معیار پیشنهادی را نشان می‌دهد. علاوه بر این، عملکرد معیار پیشنهادی نسبت به معیارهای موجود تخمین ریسک در سایر زمینه‌ها، مانند برنامه‌ها و لینک‌ها، از نظر نرخ تشخیص، دقت، فراخوانی و امتیاز F1 برتری دارد.

DOI: https://doi.org/10.22075/jme.2024.31282.2495

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

UNCORRECTED PROOF

۱-مقدمه[۱]

قرارداد هوشمند یک قرارداد خود اجرا با شرایطی است که به طور مستقیم در خطوط کد نوشته شده است. این قراردادها بر روی یک بلاک چین ذخیره و اجرا می‌شوند و تضمین می‌کنند که کد و اجرای آن شفاف، تغییرناپذیر و ایمن هستند. استفاده از بلاک چین در ارتباطات راه دور، آن‌ها را ایمن‌تر و کارآمدتر می‌کند. این به مدیریت بهتر داده‌ها و تراکنش‌ها کمک می‌کند و سیستم‌ها را امن‌تر و قابل اعتمادتر می‌کند. قراردادهای هوشمند در بلاک چین به طور خودکار شرایط قرارداد را در صورت تحقق، اجرا می‌کنند و نیاز به واسطه‌ها را کاهش می‌دهند و همچنین اعتماد و کارایی را افزایش می‌دهند. ویژگی‌های کلیدی قراردادهای هوشمند شامل اجرای خودکار، شفافیت، تغییر ناپذیری و امنیت هستند. قراردادهای هوشمند طیف گسترده‌ای از کاربردها در خدمات مالی، مدیریت زنجیره تامین، املاک و مستغلات، مراقبت‌های بهداشتی، صنعت حقوقی، صنعت ساختمان، سیستم‌های رأی‌گیری، بازی، سرگرمی و اینترنت اشیاء (IoT) را شامل می‌شوند. قراردادهای هوشمند که بر روی پلتفرم‌های بلاک چین عمل

* پست الکترونیک نویسنده مسئول:

استناد به این مقاله: نحوه استناد فارسی در اینجا درج گردد.

می‌کنند، به دلیل توانایی آن‌ها در اجرای خودکار و اجرای شرایط توافق‌نامه‌ها، به طور فزاینده‌ای محبوب شده‌اند. با این حال، افزایش استفاده از آن‌ها همچنین آسیب پذیری-های امنیتی متعددی را برجسته کرده است. مطالعات اخیر بر روی روش‌های مختلف برای شناسایی و کاهش این آسیب پذیری‌ها، متمرکز شده‌اند.

قراردادهای هوشمند مستعد آسیب‌پذیری‌های امنیتی مختلف هستند که منجر به تحقیقات قابل توجهی در مورد شناسایی و کاهش آن‌ها می‌شود. مهم‌ترین آسیب‌پذیری‌ها در قراردادها حملات ورود مجدد، سرریزها و سرریزهای اعداد صحیح، مسائل کنترل دسترسی و استثنائات کنترل نشده هستند. حملات ورود مجدد زمانی اتفاق می‌افتند که یک قرارداد قبل از به‌روزرسانی وضعیت آن، یک قرارداد خارجی را فراخوانی می‌کند و به قرارداد خارجی اجازه می‌دهد تا دوباره فراخوانی شود و تراکنش‌ها را تکرار کند. سرریزهای اعداد صحیح زمانی اتفاق می‌افتند که عملیات محاسباتی از محدودیت‌های ذخیره‌سازی متغیرها فراتر روند. در مسائل کنترل دسترسی، اجرای ضعیف می‌تواند اجازه دسترسی غیرمجاز به توابع محدود را بدهد. مدیریت خطای نادرست می‌تواند منجر به رفتار یا شکست ناخواسته قرارداد شود که به عنوان استثناهای کنترل نشده شناخته می‌شوند. تعدادی از تکنیک‌ها و ابزارها برای تشخیص آسیب پذیری قراردادهای هوشمند، همانطور که در شکل (۱) نشان داده شده است، توسعه داده شده‌اند. این روش‌ها را می‌توان در پنج گروه دسته بندی کرد: تجزیه و تحلیل استاتیک، اجرای نمادین، آزمون فاز(Fuzz Testing)، وارسی رسمی و یادگیری ماشین(ML). ابزارهای تجزیه و تحلیل استاتیک، کد را بدون اجرای آن بررسی می‌کنند و الگوهایی را جستجو می‌کنند که با آسیب‌پذیری‌های شناخته شده مطابقت دارند. teEther یک ابزار تجزیه و تحلیل است که اجازه ایجاد یک اکسپلویت برای یک قرارداد را تنها با بایت کد باینری آن می‌دهد [۱]. Slither ابزار دیگری است که یک چارچوب تحلیل ایستا برای شناسایی آسیب‌پذیری‌ها و تجسم ساختارهای قرارداد فراهم می‌کند [۲]. ابزارهای اجرای نمادین، تمام مسیرهای اجرایی ممکن را در یک قرارداد برای شناسایی آسیب‌پذیری‌ها شبیه‌سازی می‌کنند. Oyente یک ابزار اجرای نمادین اولیه برای قراردادهای هوشمند است که برای شناسایی چندین نوع آسیب‌پذیری طراحی شده است [۳]. آزمون فاز شامل ارائه ورودی‌های تصادفی یا نیمه تصادفی به یک قرارداد برای کشف رفتارهای غیرمنتظره است. Echidna یک fuzzer قوی برای قراردادهای هوشمند اتریوم است که ویژگی‌های امنیتی مختلف را آزمایش می‌کند [۴]. راستی‌آزمایی رسمی از روش‌های ریاضی برای اثبات صحت قراردادها با توجه به ویژگی‌های مشخص شده، استفاده می‌کند. KEVM توصیف رسمی ماشین مجازی اتریوم (EVM) را برای تأیید قرارداد، ارائه می‌کند [۵]. Isabelle/HOL رسماً از یک دستیار اثبات برای تأیید صحت قراردادهای هوشمند استفاده کرد [۶]. چندین پژوهش، استفاده از یادگیری ماشین را برای پیش‌بینی و شناسایی آسیب‌پذیری‌ها بر اساس داده‌های تاریخی را مورد بررسی قرار داده‌اند. تحقیقات در مورد استفاده از مدل‌های یادگیری ماشین برای تشخیص آسیب‌پذیری در حال ظهور است، که نشان‌دهنده نتایج امیدوارکننده در شناسایی الگوهایی است که به راحتی توسط روش‌های سنتی بدست نمی آیند [۷-۱۰]. مطالعات اخیر به طور فزاینده‌ای بر روی استفاده از تکنیک‌های یادگیری ماشینی برای شناسایی آسیب‌پذیری‌ها در قراردادهای هوشمند متمرکز شده‌اند. هدف این رویکردها افزایش دقت و کارایی تشخیص آسیب‌پذیری با یادگیری الگوها از داده‌های تاریخی و به‌کارگیری آن‌ها در قراردادهای جدید است.



شکل ۱- دسته بندی روش‌های تشخیص آسیب پذیری قرارداد هوشمند

یو استفاده از تکنیک‌های یادگیری عمیق برای شناسایی آسیب‌پذیری‌ها در قراردادهای هوشمند را بررسی کرد[۱۱]. این مطالعه نشان داد که این مدل‌ها می‌توانند به طور مؤثر الگوهای پیچیده را در کد ثبت کنند که منجر به بهبود نرخ تشخیص می‌شود. یک پژوهش جامع در مورد تشخیص آسیب‌پذیری قرارداد هوشمند با استفاده از شبکه‌های

UNCORRECTED PROOF

عصبی گراف(GNN[2])توسط ژوانگ و همکاران (۲۰۲۱) به بررسی کاربرد GNN برای مدل‌سازی وابستگی‌های ساختاری در کد قرارداد هوشمند، پرداخت. این مطالعه نشان داد که GNNها می‌توانند با استفاده از ساختار نمودار کد، از مدل‌های یادگیری ماشین سنتی بهتر عمل کنند و در نتیجه دقت بالاتری در تشخیص آسیب‌پذیری داشته باشند [۱۲]. علاوه بر این، Sendner و همکاران (۲۰۲۳) استفاده از یادگیری انتقالی عمیق را برای تشخیص آسیب‌پذیری قرارداد هوشمند، بررسی کردند و عملکرد مدل‌های تشخیص آسیب‌پذیری را بهبود بخشیدند. این مطالعه با پیش‌آموزش مدل‌ها بر روی مجموعه داده بزرگی از آسیب‌پذیری‌های نرم‌افزاری عمومی و تنظیم دقیق آن‌ها بر روی داده‌های خاص قرارداد هوشمند، به پیشرفت‌های قابل‌توجهی در دقت تشخیص دست یافت و نیاز به داده‌های برچسب‌گذاری شده ویژه قرارداد هوشمند گسترده را کاهش داد [۱۳]. سان و همکاران (۲۰۲۳) تکنیک‌های تحلیل استاتیک را با مدل‌های یادگیری ماشین ترکیب کردند تا قابلیت‌های تشخیص را افزایش دهند. این مطالعه از تجزیه و تحلیل استاتیک برای استخراج ویژگی‌ها از قراردادهای هوشمند استفاده می‌کند و سپس از مدل‌های یادگیری ماشینی، مانند جنگل‌های تصادفی و ماشین‌های بردار پشتیبانی (SVM[3]) برای شناسایی آسیب‌پذیری‌ها استفاده می‌کند. رویکرد ترکیبی در مقایسه با استفاده از تحلیل استاتیکی یا مدل‌های ML به تنهایی، دقت و یادآوری بهبود یافته‌ای را نشان می‌دهد [۱۴]. بکارگیری تکنیک‌های NLP برای تشخیص آسیب‌پذیری قرارداد هوشمند توسط Jeon و همکاران ارائه شد که از متدولوژی های NLP برای تجزیه و تحلیل کد به عنوان متن استفاده می‌کند. این مطالعه از تکنیک‌هایی مانند جاسازی کلمه و مدل‌های ترانسفورماتور (به عنوان مثال، BERT) برای شناسایی آسیب پذیری‌ها استفاده کرد. این رویکرد به ویژه در شناسایی آسیب‌پذیری‌های معنایی که در آن انطباق با روش‌های سنتی دشوار است، مؤثر است [۱۵]. به جای ساخت یک مدل یادگیری ماشینی، در این مطالعه، مسئله تخمین امتیاز ریسک برای قراردادهای هوشمند را بررسی کردیم. برای حل این مسئله، ما معیاری را معرفی کردیم که در آن تحلیل تفکیک خطی به همراه k-نزدیکترین همسایه (KNN)، برای تخمین ریسک قراردادهای هوشمند به کار می‌رود. در ادامه الگوریتمی برای محاسبه این معیار توسعه دادیم. نوآوری های این پژوهش را می توان به صورت زیر خلاصه کرد:

الف)پیشنهاد محاسبه میزان خطر آسیب پذیری قراردادهای هوشمند به جای دسته بندی آنها، ب) ارائه معیاری جدید بر پایه تحلیل تفکیکی خطی به منظور محاسبه خطر آسیب پذیری قراردادهای هوشمند، ج) بهبود دقت تخمین خطر امنیتی معیار پیشنهادی با استفاده از k-نزدیکترین همسایه، د) آزمایش روی داده های واقعی به منظور نشان دادن کارایی معیار پیشنهادی نسبت به چندین معیارهای مشابه قبلی

ادامه این مقاله به شرح زیر سازماندهی شده است. در بخش بعدی، برخی از تحقیقات مرتبط با تشخیص آسیب‌پذیری قرارداد هوشمند بررسی می‌شود. در بخش سوم، مسئله تحقیق بیان شده است. در بخش چهارم، معیار پیشنهادی برای برآورد ریسک قرارداد هوشمند و الگوریتم مربوط به آن معرفی شده است. بخش پنجم ارزیابی‌های تجربی را با استفاده از مجموعه داده‌های واقعی شامل قراردادهای آسیب‌پذیر و ایمن گزارش می‌کند. در نهایت، این مقاله در بخش ششم جمع بندی و نتیجه‌گیری می‌شود.

۲- پیشینه تحقیق

قرارداد هوشمند یک برنامه خود اجرا است که اقدامات لازم برای یک تراکنش بلاک چین را خودکار می‌کند. پس از اجرا، تراکنش‌ها هم قابل ردیابی و هم غیرقابل برگشت هستند. حوزه امنیت قراردادهای هوشمند به سرعت در حال پیشرفت است و محققان به طور مداوم آسیب‌پذیری‌های جدید را شناسایی کرده و روش‌های پیچیده‌ای را برای رفع آنها توسعه می‌دهند. با به کارگیری ترکیبی از یادگیری ماشین، مدل‌های ترکیبی و تکنیک‌های فازی، جامعه بلاک چین گام‌های مهمی در افزایش امنیت و قابلیت اطمینان قراردادهای هوشمند برداشته است. این تلاش‌های مداوم برای پذیرش گسترده‌تر و قابل اعتماد بودن فناوری بلاک

[2] Graph Neural Network

[3] Support Vector Machine

UNCORRECTED PROOF

چین بسیار مهم است. قراردادهای هوشمند (SCs[4]) به یک حوزه جالب توجه در فناوری بلاک چین تبدیل شده‌اند [۱۷]. با این حال، این قراردادهای دیجیتال، با ویژگی‌های متمایز خود مانند تغییر ناپذیری، اجرای خودکار، و چالش-های کد نویسی، در معرض تهدیدات امنیتی متعدد و سوء استفاده‌های بالقوه قرار دارند [۱۸]. محققان با استفاده از رویکردهای یادگیری ماشین، مدل‌های ترکیبی و تکنیک‌های فازی، آسیب‌پذیری‌های قراردادهای هوشمند مختلف از جمله وابستگی سفارش تراکنش (TOD) [۱۹]، ورود مجدد [۲۰] و وابستگی مهر زمانی [۲۰] را شناسایی و به آن‌ها رسیدگی کرده‌اند. TOD، همانطور که توسط سید و همکاران برجسته شده است [۱۹]، به عنوان یک شرایط مسابقه در بلاک چین شناخته شده است. در مقابل، لیو و همکاران [۲۱] استخراج کننده ویژگی به نام HIN را توسعه دادند و روشی را برای تشخیص حلقه‌های بینهایت پیشنهاد کردند. وابستگی سفارش تراکنش (TOD) یکی از آسیب پذیری‌های مهم در قراردادهای هوشمند است. سعید و همکاران [۱۹] TOD را به عنوان یک شرایط مسابقه‌ای که می‌تواند در زنجیره‌های بلوکی رخ دهد، شناسایی کردند. این آسیب‌پذیری زمانی به وجود می‌آید که ترتیب تراکنش‌ها بر نتیجه قرارداد تأثیر می‌گذارد و به طور بالقوه به عوامل مخرب اجازه می‌دهد تا توالی تراکنش‌ها را به نفع خود دستکاری کنند. ورود مجدد، آسیب پذیری قابل توجه دیگری است که توجه زیادی را به خود جلب کرده است. در این آسیب پذیری، یک قرارداد هوشمند بارها و بارها قبل از اتمام اجرای اولیه، قرارداد دیگری را فراخوانی می‌کند که می‌تواند منجر به سوء استفاده شود. محققان روش‌های مختلفی را برای کاهش حملات ورود مجدد مورد بررسی قرار داده‌اند، از جمله استفاده از مدل‌های ترکیبی و تکنیک‌های فازی برای شناسایی و رسیدگی به این مسائل قبل از استقرار قراردادها [۲۰]. وابستگی مهر زمانی آسیب‌پذیری است که در آن اجرای قرارداد به مهرهای زمانی بلاک چین متکی است که می‌تواند توسط ماینرها تا حدی دستکاری شود. این وابستگی می‌تواند منجر به عواقب و سوء استفاده‌های ناخواسته شود. محققان استفاده از روش‌های رسمی را برای کاهش اتکا به مهرهای زمانی و افزایش امنیت قرارداد، پیشنهاد کرده‌اند [۲۰]. در پاسخ به این آسیب‌پذیری‌ها، تکنیک‌های تشخیص مختلفی توسعه داده شده است. این رویکردها معمولاً از یادگیری ماشینی برای تجزیه و تحلیل کد قرارداد و شناسایی آسیب‌پذیری‌های احتمالی استفاده می‌کنند و اقدامی پیشگیرانه برای ایمن کردن قراردادهای هوشمند قبل از استقرار آن‌ها ارائه می‌دهند.

استفاده از یادگیری ماشین (ML) و یادگیری عمیق (DL) برای شناسایی آسیب‌پذیری‌ها و حملات در قراردادهای هوشمند در سال‌های اخیر جذابیت قابل توجهی پیدا کرده است. این تکنیک‌های پیشرفته، تجزیه و تحلیل کدهای قرارداد پیچیده را برای کشف نقایص امنیتی که روش‌های سنتی ممکن است از دست بدهند، امکان‌پذیر می‌سازد. روش‌های یادگیری ماشینی برای شناسایی آسیب‌پذیری‌ها در قراردادهای هوشمند با تجزیه و تحلیل ویژگی‌ها و الگوهای کد که نشان‌دهنده مسائل امنیتی بالقوه هستند، استفاده شده‌اند. لی و همکاران (۲۰۲۳) یک روش استخراج ویژگی خودکار را برای شناسایی آسیب‌پذیری‌هایی مانند حلقه‌های نامحدود در قراردادهای هوشمند، معرفی کردند. رویکرد آن‌ها شامل استخراج ویژگی‌های معنی‌دار از کد و استفاده از طبقه‌بندی‌کننده‌های یادگیری ماشین برای شناسایی آسیب‌پذیری‌ها است [۲۲]. مطالعات انجام شده، تجزیه و تحلیل استاتیک و پویا را با ML ترکیب کرده‌اند تا تشخیص آسیب پذیری را بهبود بخشند. به عنوان مثال، چن و همکاران ترکیب ابزارهای تحلیل استاتیک با مدل‌های یادگیری ماشین را برای افزایش تشخیص مسائل امنیتی در قراردادهای هوشمند اتریوم پیشنهاد کردند [۲۳]. تکنیک‌های یادگیری عمیق، به‌ویژه آن‌هایی که شامل شبکه‌های عصبی هستند، برای مدل‌سازی روابط پیچیده در کد قرارداد هوشمند و شناسایی آسیب‌پذیری‌های پیچیده، استفاده شده‌اند. RNN ها، به ویژه شبکه‌های حافظه کوتاه مدت (LSTM)، برای تجزیه و تحلیل دنباله کد قرارداد هوشمند، استفاده شده‌اند. به عنوان مثال، Tann و همکاران (۲۰۱۸) از شبکه‌های حافظه کوتاه‌مدت (LSTM) برای شناسایی آسیب‌پذیری‌های ورود مجدد با یادگیری از مجموعه داده‌های برچسب‌گذاری شده کد قرارداد هوشمند، استفاده کردند [۲۴]. با توجه به ساختار گراف مانند قراردادهای هوشمند، GNN ها به طور مؤثر برای تشخیص

[4] Smart Contracts

UNCORRECTED PROOF

آسیب پذیری استفاده شده‌اند. مطالعه‌ای توسط ژوانگ و همکاران (۲۰۲۱) از GNN برای نمایش و تجزیه و تحلیل قراردادهای هوشمند و شناسایی آسیب‌پذیری‌هایی که روش‌های سنتی ممکن است نادیده بگیرند، استفاده کرد [۲۵]. تحقیقات مربوط به استفاده از یادگیری ماشینی و یادگیری عمیق برای تشخیص آسیب‌پذیری قراردادهای هوشمند، نوید رسیدن به کارایی قابل توجهی را می‌دهند. با ترکیب این تکنیک‌های تحلیلی پیشرفته، محققان می‌توانند خطرات امنیتی را به طور مؤثرتری کشف و کاهش دهند و به استحکام و قابلیت اطمینان برنامه‌های بلاک چین کمک کنند.

ارزیابی خطرات امنیتی جزء ضروری امنیت سایبری است که بر شناسایی، تجزیه و تحلیل و اولویت بندی تهدیدات احتمالی برای یک سازمان یا سیستم تمرکز دارد. به جای توسعه یک مدل یادگیری ماشین برای مشخص کردن قراردادهای هوشمند آسیب‌پذیر، ما برای اولین بار به موضوع تخمین ارزش ریسک امنیتی برای این قراردادها می‌پردازیم و معیاری را برای محاسبات مربوط معرفی می‌کنیم. ارزیابی خطرات امنیتی برای سازمان‌ها و سیستم ها به طور گسترده در چارچوب سیستم‌های مدیریت امنیت اطلاعات (ISMS) مورد مطالعه قرار گرفته است [۲۶، ۲۷]. علاوه بر این، ارزیابی ریسک امنیتی برای سیستم‌های تخصصی مانند SCADA [۲۸] و سیستم‌های فیزیکی سایبری [۲۹]، از جمله اینترنت اشیاء، مورد بررسی قرار گرفته است [۳۰]. در واقع، زمینه ارزیابی امنیتی برای سیستم‌های مختلف به خوبی تثبیت شده است [۳۱، ۳۲]. با این حال، حوزه محاسبه ریسک امنیتی برای برنامه‌های کاربردی تلفن همراه و آدرس‌های منابع اینترنت (URL) حوزه‌های تحقیقاتی نزدیک به مطالعه ما هستند [۳۳]. چندین تحقیق، معیارهای ریسک موثری را برای متمایز کردن بدافزارهای اندروید از برنامه‌های کاربردی قانونی پیشنهاد کرده‌اند [۳۳، ۳۴، ۳۵]. این معیارها در درجه اول به مجوزها یا عملکردهای استفاده شده توسط برنامه‌های اندروید متکی هستند. گیتس و همکاران چندین معیار ریسک مبتنی بر مجوز [۳۳] را معرفی کردند که در میان آنها معیاری به نام RSS به طور ویژه، قابل توجه است. در [۳۴]، یک معیار ریسک بر اساس رویکرد نزدیکترین مرکز پیشنهاد شد که از نمونه‌هایی از برنامه‌های مخرب و غیر مخرب اندروید قبلاً شناخته شده، استفاده می‌کند. یک اندازه‌گیری امتیاز ریسک مبتنی بر آنتروپی در [۳۵] فرموله شد، که از اطلاعات به‌دست‌آمده از مجوزها و سایر ویژگی‌ها برای تشخیص برنامه‌های اندرویدی پرخطر از برنامه‌های معمولی استفاده می‌کند. مسئله محاسبه ریسک امنیت URL در [۳۶] معرفی شد و دو معیار برای حل این مشکل پیشنهاد شد که می‌توانند ریسک URLهای ناشناخته را تا حدودی محاسبه کنند. طبقه‌بندی مبتنی بر k- نزدیکترین همسایه و خطای بازشناسی، جهت دسته‌بندی داده‌ها در [۳۷] معرفی شده است. در این روش، ابتدا k نزدیکترین همسایه به نمونه‌های داده آزمون، از هر دسته موجود در داده‌های آموزش، محاسبه می‌گردند. سپس داده‌ی آزمون بر حسب تعداد مختلفی از نزدیکترین همسایه‌ها در هر دسته بازسازی می‌گردد. در نهایت میزان خطای بازشناسی به ازای هر تعداد همسایه، به طور مجزا محاسبه می‌شود.

۳- بیان مسئله تحقیق

هدف اصلی، توسعه یک سیستم مؤثر و کارآمد برای شناسایی آسیب‌پذیری‌ها در قراردادهای هوشمند است. این سیستم باید از تکنیک‌های پیشرفته مانند یادگیری ماشینی برای شناسایی و کاهش خطرات امنیتی احتمالی استفاده کند و از اجرای مطمئن و ایمن قراردادهای هوشمند در پلتفرم‌های بلاک چین اطمینان حاصل کند. هدف، توسعه یک چارچوب جامع برای تخمین ریسک امنیتی مرتبط با آسیب پذیری در قراردادهای هوشمند است. این چارچوب باید از تحلیل‌های پیشرفته و تکنیک‌های یادگیری ماشینی برای تعیین کمیت تأثیر بالقوه و احتمال آسیب‌پذیری‌ها استفاده کند و در نتیجه بینش‌های عملی برای بهبود امنیت قراردادهای هوشمند ارائه دهد. هدف ایجاد یک مکانیسم امتیازدهی ریسک برای تعیین کمیت خطر آسیب‌پذیری‌های قرارداد هوشمند و در نتیجه ارائه یک معیار ارزیابی ریسک واضح و قابل تفسیر است. یعنی برای یک قرارداد هوشمند، چگونه می توانیم احتمال آسیب پذیر بودن را اندازه گیری کنیم؟ نتیجه مورد انتظار شامل یک معیار قوی برای تخمین ریسک امنیتی آسیب‌پذیری‌ها در قراردادهای هوشمند و ارائه امتیازهای ریسک واضح و قابل سنجش است. بر اساس این معیار یا سنجه، یک الگوریتم تخمین ریسک خودکار که می تواند قراردادهای هوشمند را تجزیه و تحلیل کند و ارزیابی ریسک را برای توسعه دهندگان و ذینفعان ارائه دهد، توسعه یافته است.

UNCORRECTED PROOF

استناد به این مقاله: نحوه استناد فارسی در اینجا درج گردد.

پارامترهای این معیار باید با استفاده از مجموعه‌ای از قراردادهای شناخته شده قبلی بدست آید. در واقع، برای هر عضو از مجموعه قراردادهای هوشمند موجود $SCs = \{C_1, C_2, C_3,\ldots,C_n\}$، وضعیت ایمنی یا آسیب‌پذیری از قبل مشخص است. هر قرارداد C_i با استفاده از یک بردار ویژگی $F = \{f_1,f_2,f_3,\ldots,f_n\}$ نمایش داده می‌شود که با استفاده از کد منبع آن استخراج می شود. پس از محاسبه پارامترها، برای هر قرارداد ناشناخته ورودی x، می‌توان ریسک آن را با استفاده از معیار بدست آمده، تخمین زد. در واقع، ارزیابی ریسک، بینش‌های عملی را برای اولویت‌بندی تلاش‌های کاهش آسیب‌پذیری ارائه می‌دهد. این خود منجر به افزایش توانایی توسعه دهندگان و ذینفعان برای درک و کاهش خطرات امنیتی در قراردادهای هوشمند می‌شود.

تخمین ریسک آسیب پذیری به جای ایجاد مدل دسته بندی برای تشخیص قراردادهای هوشمند آسیب پذیر دارای مزایایی است. نخست اینکه ارائه یک عدد برای ریسک، نشان دهنده میزان آسیب پذیری یک قرارداد است که هرچه بزرگتر باشد یعنی احتمال خطر بیشتر است و هرچه کمتر باشد، ایمن‌تر است حال آنکه مدل دسته بندی فقط به صورت صفر و یک آسیب پذیر بودن یا ایمن بودن آن را نشان می دهد. دوم اینکه، بعد از محاسبه خطر قراردادهای هوشمند می‌توان آنها را بر اساس عدد بدست آمده اولویت بندی کرده به نحوی که کاربر یا استفاده کننده بتواند قرارداد هوشمند با کمترین خطر را برای اجرا روی بلاکچین انتخاب کند. اگرچه در گذشته معیارهایی برای محاسبه خطر برنامه‌های اندروید و همچنین خطر آدرس‌های اینترنتی ارائه شده‌اند ولی تا کنون معیاری برای سنجش میزان خطر آسیب پذیری قراردادهای هوشمند، ارائه نشده است.

۴- روش پیشنهادی

روش ما برای توسعه یک سنجه امتیاز ریسک برای قراردادهای هوشمند شامل پنج مرحله اصلی، یعنی جمع آوری داده‌ها و پیش پردازش، مهندسی ویژگی، آموزش مدل، مکانیسم امتیاز ریسک و پیاده سازی سیستم است. در مرحله جمع‌آوری و پیش‌پردازش داده‌ها، مجموعه داده‌ای جامع از قراردادهای هوشمند، گزارش‌های آسیب‌پذیری و کد منبع قراردادها را جمع‌آوری می‌کنیم. داده‌ها برای استخراج ویژگی‌ها و برچسب‌های مربوطه برای تخمین ریسک، پیش پردازش می‌شوند. در مهندسی ویژگی، تکنیک‌هایی برای استخراج و نمایش ویژگی‌هایی از کد قرارداد هوشمند و داده‌های گذشته که نشان‌دهنده وجود یا عدم وجود خطر امنیتی هستند، توسعه یافته‌اند. در مرحله سوم، یعنی ساخت و آموزش مدل، یک مدل یادگیری ماشین مؤثر را بر روی مجموعه داده برچسب‌گذاری شده، آموزش دادیم. در مکانیسم امتیازدهی ریسک، یک متریک یا مکانیزم امتیازدهی ریسک که پیش‌بینی‌های مدل و دانش خاص حوزه را برای کمی کردن ریسک امنیتی مرتبط با آسیب‌پذیری‌ها ترکیب می‌کند، ایجاد می‌شود. در آخرین مرحله یعنی پیاده سازی و ارزیابی سیستم، ما یک سیستم خودکار را پیاده سازی می‌کنیم که مدل آموزش دیده را برای تجزیه و تحلیل قراردادهای هوشمند جدید و برآورد ریسک امنیتی، ایجاد می‌کند.

ایده روش پیشنهادی مبتنی بر دسته بند k-نزدیکترین همسایه(KNN) است هرچند که هدف ما در اینجا دسته بندی نیست و قصد داریم مقدار خطر آسیب پذیری را محاسبه کنیم. بزرگترین مزیت k-نزدیکترین همسایه، سادگی مفهومی و عدم وجود فرض اولیه درباره توزیع داده‌ها است. بر خلاف مدل‌هایی مانند شبکه عصبی یا ماشین بردار پشتیبان، k-نزدیکترین همسایه یک فرمول یا تابع پیچیده نمی سازد. تصمیم گیری در لحظه تست و بر اساس همسایگی انجام می‌شود. بنابراین تبدیل کردن آن به یک معیار برای محاسبه خطر آسیب پذیری ساده‌تر از سایر مدل هاست. k-نزدیکترین همسایه یک یادگیر تنبل (Lazy Learner)است. یعنی این الگوریتم عملاً هیچ فرآیند آموزش (Training) به معنای مرسوم ندارد. تمام کاری که در مرحله آموزش انجام می‌دهد، ذخیره کردن داده‌ها است که این ویژگی مزایایی دارد. مزیت اول به روزرسانی آسان داده هاست. اگر داده جدیدی داشته باشیم، فقط کافی است آن را به مجموعه داده اصلی اضافه کنیم. نیازی به آموزش مجدد کل مدل از ابتدا (مانند درخت تصمیم یا رگرسیون لجستیک) نیست. مزیت دوم این است که این روش مناسب برای داده‌های با تغییرات پویا است. در محیط‌هایی که داده‌ها به سرعت در حال تغییرند KNN می‌تواند بدون هزینه محاسباتی زیاد برای بازآموزی، خود را با داده‌های جدید تطبیق دهد. مزیت سوم این است که اگر داده‌های آموزشی بسیار زیاد باشند و نویز کمی داشته باشند و همچنین اگر فرمول فاصله معناداری برای مسئله خود انتخاب کنیم،

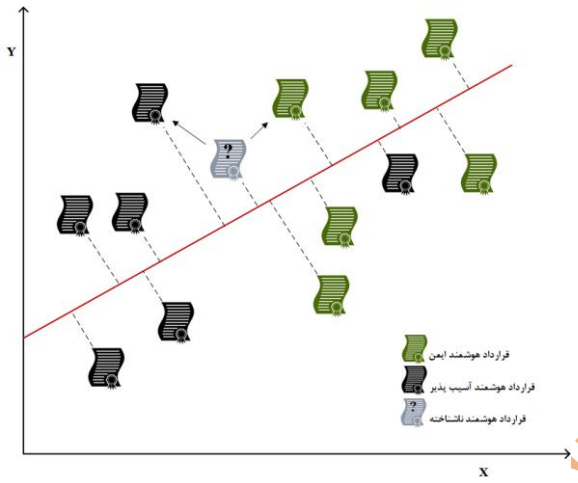
PROOF

KNN می‌تواند به دقت (Accuracy) بسیار بالایی دست یابد. و در آخر مزیت چهارم این است که k –نزدیکترین همسایه فقط به دسته بندی محدود نمی‌شود و به راحتی می‌تواند برای مسائل رگرسیون (پیشبینی مقادیر پیوسته) نیز استفاده شود. در رگرسیون، به جای رأی اکثریت، میانگین مقادیر همسایه‌ها محاسبه می شود. که در این مقاله ما از این مزیت برای ارائه یک معیار محاسبه خطر آسیب پذیری بر اساس این دسته بند استفاده کرده‌ایم.

در روش پیشنهادی می‌خواهیم ارزش ریسک را برای نمونه ناشناخته‌ای از قراردادهای هوشمند مطابق شکل (۲) محاسبه کنیم. در این شکل فضای دو بعدی برای سادگی نشان داده شده است. با این حال، در عمل، بسیاری از ابعاد (ویژگی‌های) قراردادهای هوشمند در کدهای منبع آنها قابل استخراج است. برای معیار پیشنهادی لازم است پارامترهای مورد نیاز برای تخمین ریسک در فضای اصلی و فضای نگاشت شده را محاسبه کنیم. در نهایت ریسک با کمک پارامترهای به دست آمده برآورد می‌شود. این پارامترها مربوط به فاصله نمونه ناشناخته تا نمونه‌های آسیب پذیر و ایمن در هر دو فضا می‌باشد. فضای جدید از طریق تحلیل تفکیکی خطی قراردادهای هوشمند آسیب‌پذیر و ایمن در دسترس، به دست می‌آید. در واقع نمونه ها بر اساس تحلیل تفکیکی خطی به فضای جدیدی نگاشت می‌شوند و بخشی از محاسبه ریسک در این فضای جدید انجام می‌شود. این نگاشت به این دلیل رخ می‌دهد که در فضای جدید، این دو نوع نمونه بیشتر قابل تشخیص از هم هستند و می‌توان ارزش ریسک یک قرارداد هوشمند ناشناخته را با دقت بیشتری محاسبه کرد. علاوه بر این، برای تخمین دقیق‌تر میزان ریسک، در فضای اصلی فواصل بین قرارداد هوشمند ناشناخته و k–نزدیکترین همسایه قراردادهای آسیب پذیر و k–نزدیکترین همسایگان قراردادهای ایمن به طور جداگانه محاسبه می‌شوند. در شکل (۲)، k برای سادگی برابر با یک در نظر گرفته شده است. در واقع، ما نمی‌خواهیم اطلاعات را در فضای کامل و اصلی ویژگی‌ها از دست بدهیم. به طور کلی، ریسک امنیتی یک نمونه ناشناخته با فاصله آن از نمونه‌های آسیب پذیر رابطه معکوس دارد. هر چه فاصله بین نمونه ورودی و نمونه‌های آسیب پذیر قبلی کمتر باشد، خطر بیشتر است. از طرفی ریسک امنیتی این نمونه ارتباط مستقیمی با فاصله از نمونه های ایمن دارد. یعنی هر چه فاصله آن از نمونه‌های ایمن کمتر باشد، خطر کمتری دارد. این ایده هم در فضای اصلی و هم در فضای نگاشت شده، برقرار است. بنابراین، برای محاسبه ریسک نمونه ورودی، پارامترهای مورد نیاز یعنی فواصل و مقادیر به‌دست‌آمده در هر دو فضای اصلی و نگاشت شده، با یک معادله به صورت زیر ترکیب می‌شوند:

$$Risk(x) = \frac{ds+\frac{sd_S}{k}}{dv+\frac{sd_V}{k}} \quad (۱)$$

معادله (۱) خطر آسیب پذیری یک قرارداد ناشناخته را تخمین می زند.



شکل ۲- محاسبه ریسک آسیب پذیری یک قرارداد هوشمند ناشناخته با استفاده از فضاهای اصلی و نگاشت شده توسط تحلیل تفکیکی خطی.

در این فرمول ds و dv به ترتیب فاصله قرارداد نگاشت شده x تا میانگین قراردادهای هوشمند ایمن نگاشت شده و میانگین قراردادهای آسیب پذیر نگاشت شده است. علاوه بر این، sd_s و sd_v به ترتیب مجموع فواصل قرارداد هوشمند x تا k نزدیکترین قراردادهای هوشمند ایمن و k نزدیکترین قراردادهای هوشمند آسیب پذیر در فضای اصلی اولیه هستند. همانطور که در معادله (۱) دیده می‌شود، برای بدست آوردن میانگین، این مقادیر بر k تقسیم شده‌اند.

همانطور که قبلاً اشاره شد فرمول فاصله در روش k–نزدیکترین همسایه بسیار حائز اهمیت است. برای محاسبه فاصله نمونه ناشناخته تا نمونه‌های آسیب پذیر و ایمن می‌توان از فرمول‌های فاصله مختلفی مثل فاصله اقلیدسی، منهتن، مینکوفسکی یا کسینوسی استفاده کرد. به دلیل توان دوم تفاوتها در فرمول فاصله اقلیدسی، این فاصله به مقادیر تفاوت بزرگتر، وزن بیشتری می‌دهد. درحالی که در فاصله منهتن که جمع قدرمطلق تفاوتهاست، تأثیر تفاوت‌ها

UNCORRECTED PROOF

در این فرمول افزایش نمی‌یابد. در نتیجه فاصله اقلیدسی به یک تفاوت بزرگ در یک ویژگی، حساسیت بیشتری نشان می‌دهد تا چندین تفاوت کوچک در چند ویژگی که در مجموع مقدار یکسانی دارند. این می‌تواند در شناسایی نمونه‌هایی که در یک ویژگی بسیار غیرعادی هستند، مفید باشد. بنابراین برای هدف ما که شناسایی قراردادهای هوشمند آسیب پذیر(غیرعادی) است، این فرمول نسبت به سایر فرمول‌های فاصله برتری دارد. از طرف دیگر در فاصله اقلیدسی برخلاف فاصله کسینوسی، جهت بی اهمیت است. یعنی به جهت بردار تفاوت بین دو نقطه، حساس نیست. بنابراین در فاصله اقلیدسی فقط اندازه کلی تفاوت مهم است. وقتی ویژگی‌ها کمّی و پیوسته هستند، فاصله اقلیدسی به خوبی تفاوت‌های عددی بین نمونه‌ها را اندازه می‌گیرد.

شکل(۳) فلوچارت الگوریتم پیشنهادی شامل تحلیل تفکیک خطی به همراه k-نزدیکترین همسایه برای تخمین ریسک (LNR[5]) را نشان می‌دهد. در این شکل، ورودی‌ها و خروجی الگوریتم مشخص شده است. همانطور که در این فلوچارت مشخص است به کمک قراردادهای هوشمند برچسب خورده آسیب پذیر و ایمن قبلی می‌توان خطر امنیتی یک قرارداد هوشمند ناشناخته را محاسبه کرد. پس از گرفتن ورودی‌های لازم، ماتریس‌های درون کلاسی برای هر کلاس و همچنین ماتریس بین کلاسی برای دو کلاس آسیب پذیر و ایمن، بدست می‌آیند. این ماتریس‌ها برای محاسبه ماتریس تبدیل تفکیکی خطی مطابق روش LDA لازم هستند که در مرحله بعد محاسبه می‌گردد. سپس قرارداد هوشمند ناشناخته و همچنین کل قراردادهای آسیب پذیر و ایمن به کمک ماتریس تبدیل LDA به فضای جدید نگاشت می‌شوند و در فضای جدید فاصله قرارداد ناشناخته x تا قراردادهای ایمن و آسیب پذیر محاسبه می‌گردد. اما برای محاسبه خطر قرارداد ناشناخته، علاوه بر فاصله‌ها در فضای نگاشت شده، مطابق فرمول(۱) مجموع فواصل در فضای اصلی نیز نیاز است که در ادامه فلوچارت، محاسبه می‌گردند. در هر دور از حلقه‌ای که در فلوچارت شکل(۳) نشان داده شده است، به ترتیب فاصله قرارداد ناشناخته ورودی از نزدیکترین قراردادهای هوشمند آسیب پذیر محاسبه شده و به مجموع بدست آمده تا آن دور اضافه می گردد. نزدیکترین همسایه از لیست قراردادهای آسیب حذف می‌گردد تا در دور بعدی نزدیکترین همسایه بعدی مورد توجه قرار گیرد. همین فرایند برای نزدیکترین همسایه از نوع ایمن(غیر آسیب پذیر) نیز انجام می‌گیرد تا در پایان k دور دو مجموع فاصله یعنی مجموع فاصله قرارداد ناشناخته ورودی از k نزدیکترین قراردادهای آسیب پذیر و مجموع فاصله تا k نزدیکترین قرارداد ایمن را داشته باشیم. بعد از پایان حلقه، میانگین‌ها به سادگی با تقسیم بر k محاسبه شده و به همراه فواصل بدست آمده در فضای نگاشت شده که قبل از حلقه بدست آمدند به منظور محاسبه خطر آسیب پذیری مطابق فرمول(۱) به کار می‌روند که با توجه به شکل (۳)، خروجی فلوچارت خواهد بود.



[5] Linear discriminant analysis with Nearest neighbour for Risk estimation

شکل ۳- فلوچارت الگوریتم پیشنهادیLNR

به منظور بررسی جزئیات الگوریتم پیشنهادی، شکل (۴) شبه کد الگوریتم پیشنهادی را نشان می دهد. این الگوریتم مجموعه‌ای از قراردادهای هوشمند قبلاً شناخته شده (SCs)، یک قرارداد هوشمند ناشناخته x و تعداد نزدیکترین همسایگان k را به عنوان ورودی می‌گیرد و مقدار ریسک امنیتی آن را بر می‌گرداند. جدول(۱) نمادهای استفاده شده در الگوریتم را نشان می‌دهد. در خط ۱، داده‌های ورودی نرمال می‌شوند. در خطوط ۲ و ۳ به ترتیب مجموعه‌ای از قراردادهای هوشمند آسیب پذیر و ایمن استخراج شده‌اند. مقادیر میانگین این دو مجموعه به ترتیب در خطوط ۴ و ۵ محاسبه شده است. در خطوط ۶ و ۷، ماتریس‌های پراکندگی درون کلاسی برای هر دو کلاس (مجموعه) قراردادهای هوشمند آسیب پذیر و غیرآسیب پذیر به دست آمده است. پس از محاسبه ماتریس پراکندگی بین کلاسی در خط ۸، ماتریس تبدیل LDA در خط ۹ محاسبه می‌شود. در خطوط ۱۰ و ۱۱، مجموعه‌های نگاشت شده از قراردادهای هوشمند آسیب پذیر و غیرآسیب پذیر به دست می‌آیند. متعاقباً، در خط ۱۲، نگاشت قرارداد هوشمند ورودی ناشناخته x نیز بدست می‌آید. پس از آن، مقادیر میانگین دو مجموعه در خط ۱۳ محاسبه می‌شوند. در خطوط ۱۴ و ۱۵، فاصله نمونه مجهول پیش بینی شده x تا هر دو مقدار میانگین محاسبه می‌شود. این فواصل برای تخمین مقادیر ریسک قرارداد هوشمند ورودی ناشناخته x مورد نیاز است. در دستورات باقی مانده از الگوریتم LNR، دو پارامتر مورد نیاز دیگر نیز به دست می آیند. در خط ۱۶، متغیر sd_v که برای ذخیره مجموع فاصله‌های نزدیکترین همسایه‌ها از نوع قرارداد هوشمند آسیب‌پذیر به قرارداد هوشمند ورودی x لازم است، مقداردهی اولیه می‌شود.

Procedure LNR (SCs, x, k)

begin

1. SCs = normalize (SCs);
2. $VS = \{\forall\ vulnerable\ \ sc \in SCs\}$;
3. $SS = \{\forall\ safe\ \ sc \in SCs\}$;
4. $mv = mean(VS)$;
5. $ms = mean(SS)$;
6. $SW_V = \sum_{x_i \in VS}(x_i - mv) \times (x_i - mv)^T$;
7. $SW_S = \sum_{x_i \in SS}(x_i - ms) \times (x_i - ms)^T$;
8. $SW = SW_V + SW_S$;
9. $W = SW^{-1} \times (mv - ms)$;
10. $Yv = W^T \times VS$;
11. $Ys = W^T \times SS$;
12. $yx = W^T \times x$;
13. $mYv = mean(Yv)$; $mYs = mean(Ys)$;
14. $dv = |yx - mYv|$;
15. $ds = |yx - mYs|$;
16. sd_v=0;
17. **for i** =1 to **k do**
18. $nv = first\ member\ of\ VS$;
19. O_v= ***ED***(x, nv);
20. **for each** $v_i \in VS$ **do**
21. O_i = ***ED***(x, v_i);
22. **if** $O_i \leq O_v$ **then**
23. $nv = v_i$;
24. $O_v = O_i$;
25. **Endfor**;
26. $SV = SV - \{nv\}$;
27. sd_v+= O_v;
28. **Endfor**;
29. $sd_s = 0$;
30. **for i** =1 to **k do**
31. $ns = first\ member\ of\ SS$;
32. O_s = ***ED***(x, ns);
33. **for each** $s_i \in SS$ **do**
34. O_i = ***ED***(x, s_i);
35. **if** $O_i \leq O_s$ **then**
36. $ns = s_i$;
37. $O_s = O_i$;
38. **Endfor**;
39. $SS = SS - \{ns\}$;
40. sd_s+= O_s;
41. **Endfor**;
42. **return** $\frac{ds+\frac{sd_s}{k}}{dv+\frac{sd_v}{k}}$;

end

شکل ۴- شبه کد الگوریتم پیشنهادیLNR

در خطوط ۱۷ تا ۲۸ محاسبات لازم k بار برای به دست آوردن نزدیکترین همسایه‌های آسیب پذیر انجام می‌گردد که به شرح زیر توضیح می‌دهیم. در خط ۱۸، اولین قرارداد هوشمند از مجموعه موارد آسیب‌پذیر به‌عنوان مقداردهی اولیه نزدیک‌ترین همسایه آسیب‌پذیر در نظر گرفته می‌شود. سپس فاصله اقلیدسی آن تا قرارداد ورودی در خط ۱۹ محاسبه می شود.

جدول ۱-نمادهای استفاده شده و معانی آنها در الگوریتم LNR

نماد	معنای نماد
SCs	مجموعه قراردادهای هوشمند شناخته شده (آسیب پذیر و ایمن)
VS	مجموعه قراردادهای هوشمند آسیب پذیر
SS	مجموعه قراردادهای هوشمند ایمن
mv	مقدار میانگین برای مجموعه قراردادهای هوشمند آسیب پذیر

UNCORRECTED PROOF

ms	مقدار میانگین برای مجموعه قراردادهای هوشمند ایمن
SW_v	ماتریس پراکندگی درون کلاسی برای قراردادهای هوشمند آسیب پذیر
SW_s	ماتریس پراکندگی درون کلاسی برای قراردادهای هوشمند ایمن
SW	جمع ماتریس های پراکندگی درون کلاسی
W	ماتریس پراکندگی بین کلاس
Yv	مجموعه قراردادهای هوشمند آسیب پذیر نگاشت شده
Ys	مجموعه قراردادهای هوشمند ایمن نگاشت شده
yx	قرارداد هوشمند ورودی نگاشت شده x
mYv	میانگین قراردادهای هوشمند آسیب پذیر نگاشت شده
mYs	میانگین قراردادهای هوشمند ایمن نگاشت شده
dv	فاصله بین قرارداد نگاشت شده x و mYv
ds	فاصله بین قرارداد نگاشت شده x و mYs
sd_v	مجموع فواصل k-نزدیکترین همسایه از نوع قراردادهای آسیب پذیر
sd_s	مجموع فواصل k-نزدیکترین همسایه از نوع قراردادهای ایمن
nv	نزدیکترین قرارداد هوشمند آسیب پذیر در هر تکرار
ns	نزدیکترین قرارداد هوشمند ایمن در هر تکرار
ED	تابع فاصله اقلیدسی
O_v	فاصله نزدیکترین قرارداد هوشمند آسیب پذیر در هر تکرار
O_s	فاصله نزدیکترین قرارداد هوشمند ایمن در هر تکرار
v_i	هر یک از اعضای VS
s_i	هر یک از اعضای SS

در خطوط ۲۰ تا ۲۵ با بررسی تمام اعضای متعلق به مجموعه قراردادهای آسیب پذیر و محاسبه فاصله اقلیدسی آنها با قرارداد ورودی x، نزدیکترین قرارداد آسیب پذیر را شناسایی می‌کنیم. دستور ۲۶ این قرارداد را از مجموعه قراردادهای آسیب پذیر حذف می‌کند تا در دور بعدی بتوانیم نزدیک ترین همسایگان از این نوع را شناسایی کنیم. سپس در خط ۲۷ مقدار فاصله اقلیدسی قرارداد بدست آمده را به مجموع جزئی فواصل اقلیدسی به نزدیکترین همسایگان شناخته شده اضافه می‌کنیم تا در پایان دورهای باقیمانده بتوانیم مجموع فاصله های اقلیدسی کل مجموعه k-نزدیک ترین همسایه‌های آسیب پذیر را بدست آورده و در متغیر مربوطه ذخیره کنیم. همین فرآیند در خطوط ۲۹ تا ۴۱ برای قراردادهای هوشمند ایمن انجام می‌شود تا در نهایت نزدیکترین همسایه‌ها از نوع امن را نسب به قرارداد ورودی x بدست آوریم و متغیرهای مربوطه را به روز کنیم.

در نتیجه، با اجرای k دور به همین ترتیب، مجموع فاصله k نزدیکترین قراردادهای آسیب پذیر و مجموع k فاصله نزدیکترین قراردادهای ایمن با قرارداد ورودی x به ترتیب در متغیرهای sd_v و sd_s ذخیره می‌شوند. پس از به دست آوردن پارامترهای مورد نیاز، در نهایت در خط ۴۲، ریسک آسیب پذیری یا امنیتی قرارداد ورودی x با استفاده از فرمول (۱) همانطور که قبلا توضیح داده شد، محاسبه می‌گردد و به عنوان حاصل کار این الگوریتم برگردانده می‌شود.

۵- ارزیابی و مقایسه

برای ارزیابی الگوریتم پیشنهادی، علاوه بر LNR، یک متریک پیشنهادی دیگر بنام تحلیل تفکیکی خطی برای محاسبه ریسک (LDR[6]) را پیاده‌سازی کردیم که فقط شامل بخش LDA از متریک پیشنهادی LNR است. برای ارزیابی تطبیقی، در این مطالعه از مهمترین معیارهای امتیاز ریسک ارائه شده در مطالعات قبلی استفاده شده است. این معیارها رویکردهای امتیازدهی ریسک مبتنی بر نادر بودن (RSS) [۳۳]، مبتنی بر نمونه (IRS) [۳۴]، مبتنی بر آنتروپی (ERS) [۳۵]، FRU و NRU [۳۶] هستند. سه معیار اول برای تخمین ریسک امنیتی برنامه‌های اندروید و دو معیار آخر برای تخمین ریسک امنیتی URLهای ناشناخته طراحی شده‌اند. با این حال، هر پنج معیار از معیارهای قوی ارائه شده در زمینه محاسبه ریسک امنیتی حملات و آسیب پذیری‌ها هستند. از آنجایی که هیچ معیاری در تحقیقات گذشته برای محاسبه ریسک آسیب پذیری قراردادهای هوشمند ارائه نشده است، ما این معیارها را برای ارزیابی و مقایسه با معیارهای پیشنهادی LNR و LDR پیاده سازی کرده‌ایم. تمام پیاده سازی‌ها و ارزیابی ها با استفاده از MATLAB 2023 انجام شد. همه آزمایش‌ها بر روی یک رایانه شخصی دارای پردازنده Intel Core i9، ۱۶ گیگابایت رم و دارای نسخه ۶۴ بیتی ویندوز ۱۰ انجام شد. برای ارزیابی و مقایسه این معیارها، از مجموعه داده معرفی شده در [۱۶] که متشکل از قراردادهای هوشمند آسیب پذیر و ایمن روی بلاکچین اتریوم است، استفاده شده است. این مجموعه داده شامل ۷۰ ویژگی استخراج شده از ۳۶۶۷۰ قرارداد هوشمند است که ۹۷۵۶ مورد آسیب پذیر و ۲۶۹۱۴ مورد ایمن هستند.

[6] Linear Discriminant analysis for Risk estimation

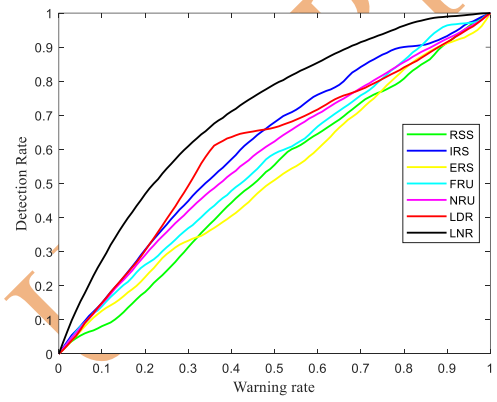
UNCORRECTED PROOF

بنابراین، در این مجموعه داده تعداد قراردادهای هوشمند ایمن تقریباً سه برابر تعداد قراردادهای آسیب پذیر است. مجموعه داده مورد استفاده و پیاده‌سازی‌های ما از LNR، LDR و سایر معیارهای امتیاز ریسک که در این تحقیق برای مقایسه پیاده سازی شده‌اند، در آدرس گیتهاب https://github.com/mdeypir/LNR در دسترس هستند.

برای ارزیابی مؤثر و مقایسه روش پیشنهادی با رویکردهای قبلی، آزمایش‌های مختلفی را از زوایای مختلف انجام دادیم. در این تست‌ها بر روی میزان تشخیص، دقت، یادآوری و امتیاز F1 تمرکز کردیم که در این قسمت به تفصیل توضیح داده شده‌اند. همانطور که قبلاً اشاره شد، معیارهای اندازه‌گیری و برآورد ریسک مؤثر، باید مقادیر ریسک بالایی را برای قراردادهای هوشمند آسیب‌پذیر و مقادیر نسبتاً پایینی را برای قراردادهای ایمن ایجاد کنند تا به وضوح بین این دو نوع تمایز قائل شوند. در نتیجه، آزمایش‌ها بر نرخ‌های تشخیص این معیارها تأکید می‌کنند. یک معیار موفق معیاری است که مقادیر ریسک امنیتی بالاتری را به قراردادهای آسیب پذیر اختصاص دهد. یعنی اگر ارزش ریسک را برای لیستی از هر دو نوع قرارداد ایمن و آسیب پذیر با استفاده از یک معیار قوی محاسبه کنیم و سپس آنها را به ترتیب نزولی مقدار ریسک بدست آمده رتبه بندی کنیم، بیشتر قراردادهای آسیب پذیر باید در بالای لیست حاصل، ظاهر شوند.

در آزمایش اول، هدف ما مقایسه نرخ تشخیص دو معیار پیشنهادی، LNR و LDR، در برابر پنج معیار قبلی است. برای این کار، ما از ۹۰٪ قراردادهای هوشمند مجموعه داده برای ساخت مدل‌های خود استفاده کردیم و هر معیار را با ۱۰٪ باقی مانده آزمایش کردیم. این رویکرد به ما امکان داد تا ریسک امنیتی را با استفاده از مدل توسعه‌یافته محاسبه کنیم و متعاقباً مقادیر ریسک را به ترتیب نزولی رتبه‌بندی کنیم. در هر آزمایش، درصدهای مختلفی را از بالای فهرست به دست آمده انتخاب کردیم تا مشخص کنیم چه نسبتی از آن قراردادها آسیب پذیر هستند. ما از این درصدهای انتخاب شده به عنوان نرخ هشدار یاد می‌کنیم، در حالی که نسبت قراردادهای آسیب پذیر شناسایی شده، نرخ شناسایی نامیده می شود. اگرچه مقادیر ریسک محاسبه‌شده ممکن است در معیارهای مختلف متفاوت باشند، این روش ارزیابی تضمین می‌کند که تفاوت‌ها در محدوده‌های مقادیر بر مقایسه‌ها تأثیر نمی‌گذارد، زیرا مقدار ریسک، نسبی است نه مطلق. واضح است که یک معیار قوی‌تر باعث می‌شود که درصد بیشتری از قراردادهای آسیب‌پذیر در بالای فهرست ظاهر شوند و در نتیجه منجر به افزایش نرخ شناسایی شود. در هر سناریو، نرخ تشخیص را بر اساس نرخ هشدار مربوطه محاسبه کردیم.

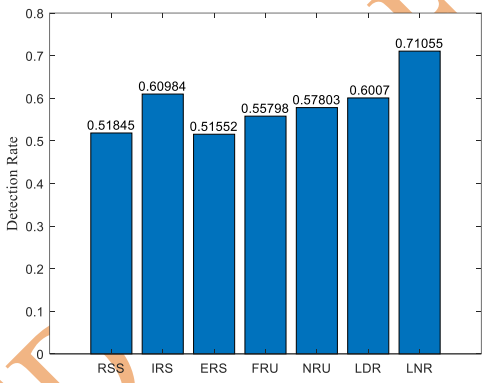
در آزمایش‌های انجام شده، کارایی روش پیشنهادی LNR را با مقدار k=10 با سایر روش‌ها مقایسه می‌کنیم. شکل(۵) منحنی‌های مربوط به معیار پیشنهادی و سایر معیارها را روی مجموعه داده توصیف شده فوق، نشان می‌دهد. در این شکل، محور افقی نرخ هشدار و محور عمودی نرخ تشخیص است. همانطور که در این شکل نشان داده شده است، الگوریتم LNR پیشنهادی به طور قابل توجهی کارآمدتر از سایر روش‌های در نظر گرفته شده است. IRS و LDR به ترتیب رتبه های دوم و سوم کارایی را دارند. با کمک تجزیه و تحلیل تفکیک خطی، روش‌های پیشنهادی LNR و LDR انواع قرارداد ایمن و آسیب‌پذیر را به فضای جدیدی که قابلیت تفکیک بیشتری دارد، نگاشت می‌کنند. در این فضای جدید، برآورد ریسک با دقت بیشتری انجام می‌شود. علاوه بر این، در LNR پیشنهادی، کل ویژگی‌های فضای اصلی نیز برای محاسبه ریسک از طریق روش نزدیک‌ترین همسایه در نظر گرفته می‌شوند.



شکل۵- مقایسه نرخ تشخیص معیارها برای نرخ های مختلف هشدار

CORRECTED PROOF

برای مقایسه بهتر کارایی معیارها برای همه مقادیر نرخ هشدار، شکل(۶) مساحت زیر منحنی (AUC[7]) را نشان می‌دهد. همانطور که از شکل (۵) و شکل (۶) استنباط می‌شود، به طور کلی روش پیشنهادی دارای نرخ تشخیص بالاتری است که نشان می‌دهد معیار قوی تری برای محاسبه ریسک امنیتی و در نهایت شناسایی قراردادهای هوشمند آسیب پذیر است.



شکل ۶: مقایسه مساحت زیر منحنی(AUC)

به منظور محاسبه ماتریس درهم‌ریختگی، آزمایشی برای محاسبه مثبت واقعی و کاذب و همچنین منفی واقعی و کاذب انجام دادیم. پس از محاسبه ماتریس درهم‌ریختگی، نمرات دقت(Precision)، فراخوانی (Recall)و امتیاز F1 همه معیارهای محاسبه ریسک روی مجموعه داده محاسبه شده و نتیجه در جدول (۲) گزارش شده است. برای این منظور، مقادیر به شرح زیر در نتایج تعیین می‌شوند. مثبت واقعی نشان‌دهنده بدافزارهایی است که با موفقیت شناسایی شده‌اند. مثبت کاذب نشان‌دهنده تعداد برنامه‌های بی‌خطری است که به اشتباه به عنوان بدافزار شناسایی شده‌اند. منفی‌های کاذب تعداد بدافزارهایی را نشان می‌دهند که به اشتباه به عنوان برنامه‌های بی‌خطر شناسایی شده‌اند. دقت، نسبت تعداد بدافزارهایی است که با موفقیت شناسایی شده‌اند به تعداد کل بدافزارهای شناسایی شده. معادله این نسبت به صورت زیر است:

$$Percision = \frac{TP}{TP + FP} \quad (۲)$$

فراخوانی برابر است با تعداد بدافزارهایی که با موفقیت شناسایی شده‌اند به تعداد تمام بدافزارهایی که با موفقیت شناسایی شده‌اند یا به صورت اشتباه به عنوان بی‌ضرر طبقه‌بندی شده‌اند:

$$Recall = \frac{TP}{TP + FN} \quad (۳)$$

امتیاز F1 که میانگین هارمونیک حاصل از دقت و فراخوانی است، نشان می‌دهد که مدل چقدر در تمایز بین دو کلاس توانمند عمل می‌کند:

$$\text{F1-Score} = 2 \times \frac{Percision \times Recall}{Percision + Recall} \quad (۴)$$

جدول ۲- مقایسه‌ی دقت، فراخوانی و امتیاز F1

معیار	دقت	یادآوری	امتیازF1
LNR	0.7778	0.7663	0.772
LDR	0.5932	0.5821	0.5875
NRU[۳۶]	0.5616	0.5506	0.5561
FRU[۳۶]	0.5512	0.5405	0.5458
ERS[۳۵]	0.5192	0.5103	0.5147
IRS[۳۴]	0.6272	0.6252	0.6262
RSS[۳۳]	0.5183	0.5101	0.5142

همانطور که در جدول (۲) مشاهده می‌شود، LNR پیشنهادی بهترین دقت، فراخوانی و امتیاز F1 را به دست می‌آورد. دلیل این امر این است که این معیار مقدار ارزش ریسک را دقیق‌تر از سایر معیارها تخمین می‌زند. بنابراین، مطابق نتایج، در مقایسه با سایر معیارهای اندازه‌گیری ریسک، مثبت‌های واقعی بیشتر، مثبت‌های کاذب و منفی‌های کاذب کمتری به دست می‌آورد. دلیل برتری معیار LNR پیشنهادی نسبت به معیارهای قبلی این است که در LNR، ارزش ریسک قراردادها بر اساس اطلاعات هر دو فضای اصلی و نگاشت شده به کمک تحلیل تفکیکی خطی، به طور همزمان برآورد می‌شود. در حالی که فواصل لازم بر اساس k-نزدیکترین همسایه در فضای با بعد اصلی اولیه محاسبه می‌شود، پس از نگاشت نمونه‌ها نیز با استفاده از تحلیل تفکیک خطی، فواصل لازم در فضای کم بعد

[7] Area Under the Curve

کاهش‌یافته که نمونه‌ها از هم جداتر هستند، محاسبه می‌شوند. با استفاده از اطلاعات به دست آمده در هر دو فضا، مقادیر ریسک نهایی با استفاده از معیار پیشنهادی برای قراردادهای ناشناخته، دقیق‌تر محاسبه می شوند.

تخمین ریسک آسیب پذیری در روش پیشنهادی نسبت به ایجاد مدل های دسته بندی ارائه شده در [۱۶] برای تشخیص قراردادهای هوشمند آسیب پذیر، دارای برتری هایی است. نخست اینکه ارائه یک مقدار برای ریسک آسیب پذیری، نشان دهنده میزان آسیب پذیری یک قرارداد است که هرچه بزرگتر باشد یعنی احتمال خطر بیشتر است و هرچه کمتر باشد، آن قرارداد ایمن‌تر است حال آنکه مدل دسته بندی فقط به صورت صفر و یک آسیب پذیر بودن یا نبودن آن را نشان می دهد که سخت گیرانه است و توصیف پذیری کمتری دارد. دوم اینکه، در روش پیشنهادی بعد از محاسبه خطر قراردادهای هوشمند همسان می‌توان این قراردادها را بر اساس عدد بدست آمده ریسک اولویت بندی کرده به نحوی که کاربر یا استفاده کننده بتواند قرارداد هوشمند با کمترین خطر را برای اجرا روی بلاکچین انتخاب کند، در حالی که در مدل های دسته بندی این امکان وجود ندارد. مورد سوم اینکه، اگرچه روش پیشنهادی نیز مانند مدل های دسته بندی دارای خطا در تخمین مقدار ریسک محاسبه شده است اما نوع خطا در روش پیشنهادی متفاوت بوده و نسبت به مدل های دسته بندی کمتر کاربر را به اشتباه می اندازد زیرا برخلاف مدل های دسته بندی، روش پیشنهادی هیچگاه یک قرارداد آسیب پذیر را به عنوان یک قرارداد ایمن قطعی(یا برعکس) معرفی نمی کند، بلکه تنها ممکن است به اشتباه، مقدار ریسک کم(یا زیادی) برای آن بدست آورد.

در مورد معایب روش پیشنهادی می توان گفت که اگرچه روش پیشنهادی نسبت به معیارهای قبلی دقت بیشتری دارد اما بدست آوردن ماتریس تبدیل LDA و سپس نگاشت نمونه ها به فضای جدید و همچنین محاسبه نزدیکترین همسایه ها از دو نوع آسیب پذیر و ایمن، سربار محاسباتی ایجاد می کند. از طرف دیگر با وجود برتری روش پیشنهادی از نظر دقت، یادآوری و امتیاز F1، با توجه به مقادیر بدست آمده هنوز هم نیاز به افزایش کارایی احساس می شود و تا رسیدن به یک معیار مطلوب راه زیادی باقی مانده است.

۶- نتیجه گیری

صنایع و مشاغل متعددی از جمله مخابرات، داروسازی، ساختمان، لوازم الکترونیکی، تجارت الکترونیک، مراقبت-های بهداشتی، بیمه و بخش های مختلف آنها مانند زنجیره-های تأمین و شهرهای هوشمند، از راه حل‌های مبتنی بر بلاک چین برای رسیدگی به مسائل مربوط به اعتماد بهره-برداری کرده‌اند [۳۸، ۳۹]. قراردادهای هوشمند به عنوان توافق‌نامه‌های خود اجرا ذخیره می‌شوند و در یک شبکه بلاک چین تکرار می‌شوند و شفافیت و تغییر ناپذیری را تضمین می‌کنند. با این حال، قراردادهای هوشمند تعدادی آسیب پذیری بالقوه دارند که بر مزایای استفاده از آنها در بلاک چین تأثیر می‌گذارد. تشخیص آسیب‌پذیری قرارداد هوشمند برای حفظ امنیت و قابلیت اطمینان برنامه‌های بلاک چین، بسیار مهم است. برآورد ریسک امنیتی مرتبط با آسیب‌پذیری‌های قرارداد هوشمند برای حفظ یکپارچگی و قابل اعتماد بودن برنامه‌های بلاک چین ضروری است. در این مطالعه، علاوه بر معرفی مسئله تخمین ریسک برای قراردادهای هوشمند، یک معیار جدید، بنام LNR، برای برآورد مؤثر ریسک قراردادهای هوشمند ناشناخته پیشنهاد شده است. این معیار از اطلاعات فضاهای اصلی و نگاشت شده از طریق تحلیل تفکیکی خطی نمونه‌های شناخته شده قبلی قراردادهای هوشمند، استفاده می‌کند. امروزه یادگیری عمیق کاربردهای گسترده‌ای پیدا کرده است[۴۰، ۴۱]. همانطور که قبلاً اشاره شد، مدل‌های بر اساس یادگیری عمیق به منظور شناسایی قراردادهای هوشمند آسیب پذیر ارائه شده‌اند[۱۱، ۱۳، ۴۲، ۴۳] که البته این مدل‌ها به منظور دسته بندی قراردادها هستند و برای محاسبه خطر آسیب پذیری به نحو دقیق‌تر بر اساس یادگیری عمیق، نیاز به ارائه معیارهای جدیدی داریم. ما در آینده قصد داریم از یادگیری عمیق و مدل های مربوطه به منظور سنجش ریسک آسیب پذیری قراردادهای هوشمند در بلاکچین استفاده کنیم و معیارهای مؤثرتری در این زمینه ارائه دهیم. با توجه به اینکه برای آموزش اینگونه مدل‌ها نیازمند مجموعه داده‌های بزرگ هستیم، می‌بایست ابتدا مجموعه داده‌های بزرگ و با کیفیتی از قراردادهای هوشمند ایمن و آسیب پذیر را گردآوری کنیم تا در نهایت بتوانیم میزان ریسک آسیب پذیری را دقیق‌تر تخمین بزنیم. بنابراین جمع آوری داده و ایجاد یک مجموعه داده جامع از قراردادهای

CORRECTED

هوشمند آسیب پذیر و ایمن به نحوی که بتوان مدل های عمیق را آموزش دهیم، برای تحقیقات آینده لازم است. همچنین استفاده از مدلهای زبانی بزرگ به منظور استخراج نوع و توصیف آسیب پذیری، از متن قراردادهای هوشمند از دیگر کارهای آینده است.

مراجع

[1] Krupp, Johannes, and Christian Rossow. "{teEther}: Gnawing at ethereum to automatically exploit smart contracts." In *27th USENIX security symposium (USENIX Security 18)*, pp. 1317-1333. 2018.

[2] Feist, Josselin, Gustavo Grieco, and Alex Groce. "Slither: a static analysis framework for smart contracts." In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pp. 8-15. IEEE, 2019.

[3] Luu, Loi, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. "Making smart contracts smarter." In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 254-269. 2016.

[4] Grieco, Gustavo, Will Song, Artur Cygan, Josselin Feist, and Alex Groce. "Echidna: effective, usable, and fast fuzzing for smart contracts." In *Proceedings of the 29th ACM SIGSOFT international symposium on software testing and analysis*, pp. 557-560. 2020.

[5] Hildenbrandt, Everett, Manasvi Saxena, Nishant Rodrigues, Xiaoran Zhu, Philip Daian, Dwight Guth, Brandon Moore et al. "Kevm: A complete formal semantics of the ethereum virtual machine." In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 204-217. IEEE, 2018.

[6] Amani, Sidney, Myriam Bégel, Maksym Bortin, and Mark Staples. "Towards verifying ethereum smart contract bytecode in Isabelle/HOL." In *Proceedings of the 7th ACM SIGPLAN international conference on certified programs and proofs*, pp. 66-77. 2018.

[7] Shakya, Supriya, Arnab Mukherjee, Raju Halder, Abyayananda Maiti, and Amrita Chaturvedi. "Smartmixmodel: machine learning-based vulnerability detection of solidity smart contracts." In *2022 IEEE international conference on blockchain (Blockchain)*, pp. 37-44. IEEE, 2022.

[8] Eshghie, Mojtaba, Cyrille Artho, and Dilian Gurov. "Dynamic vulnerability detection on smart contracts using machine learning." In *Proceedings of the 25th international conference on evaluation and assessment in software engineering*, pp. 305-312. 2021.

[9] Sürücü, Onur, Uygar Yeprem, Connor Wilkinson, Waleed Hilal, S. Andrew Gadsden, John Yawney, Naseem Alsadi, and Alessandro Giuliano. "A survey on ethereum smart contract vulnerability detection using machine learning." *Disruptive Technologies in Information Sciences VI* 12117 (2022): 110-121.

[10] Xu, Yingjie, Gengran Hu, Lin You, and Chengtang Cao. "A novel machine learning-based analysis model for smart contract vulnerability." *Security and Communication Networks* 2021, no. 1 (2021): 5798033.

[11] Yu, Xingxin, Haoyue Zhao, Botao Hou, Zonghao Ying, and Bin Wu. "Deescvhunter: A deep learning-based framework for smart contract vulnerability detection." In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE, 2021.

[12] Zhuang, Yuan, Zhenguang Liu, Peng Qian, Qi Liu, Xiang Wang, and Qinming He. "Smart contract vulnerability detection using graph neural networks." In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pp. 3283-3290. 2021.

UNCORRECTED PROOF

[13] Sendner, Christoph, Huili Chen, Hossein Fereidooni, Lukas Petzi, Jan König, Jasper Stang, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, and Farinaz Koushanfar. "Smarter Contracts: Detecting Vulnerabilities in Smart Contracts with Deep Transfer Learning." In *NDSS*. 2023.

[14] Wu, Huaiguang, Hanjie Dong, Yaqiong He, and Qianheng Duan. "Smart contract vulnerability detection based on hybrid attention mechanism model." *Applied Sciences* 13, no. 2 (2023): 770.

[15] Jeon, Sowon, Gilhee Lee, Hyoungshick Kim, and Simon S. Woo. "Smartcondetect: Highly accurate smart contract code vulnerability detection mechanism using bert." In *KDD workshop on programming language processing*, vol. 16, p. 225. 2021.

[16] HajiHosseinKhani, Sepideh, Arash Habibi Lashkari, and Ali Mizani Oskui. "Unveiling vulnerable smart contracts: Toward profiling vulnerable smart contracts using genetic algorithm and generating benchmark dataset." *Blockchain: Research and Applications* 5, no. 1 (2024): 100171.

[17] Zhang, Lejun, Weijie Chen, Weizheng Wang, Zilong Jin, Chunhui Zhao, Zhennao Cai, and Huiling Chen. "Cbgru: A detection method of smart contract vulnerability based on a hybrid model." *Sensors* 22, no. 9 (2022): 3577.

[18] Liu, Chao, Han Liu, Zhao Cao, Zhong Chen, Bangdao Chen, and Bill Roscoe. "Reguard: finding reentrancy bugs in smart contracts." In *Proceedings of the 40th international conference on software engineering: companion proceeedings*, pp. 65-68. 2018.

[19] Sayeed, Sarwar, Hector Marco-Gisbert, and Tom Caira. "Smart contract: Attacks and protections." *Ieee Access* 8 (2020): 24416-24427.

[20] Liu, Zhenguang, Peng Qian, Xiaoyang Wang, Yuan Zhuang, Lin Qiu, and Xun Wang. "Combining graph neural networks with expert knowledge for smart contract vulnerability detection." *IEEE Transactions on Knowledge and Data Engineering* 35, no. 2 (2021): 1296-1310.

[21] Liu, Lin, Wei-Tek Tsai, Md Zakirul Alam Bhuiyan, Hao Peng, and Mingsheng Liu. "Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum." *Future Generation Computer Systems* 128 (2022): 158-166.

[22] Li, Lina, Yang Liu, Guodong Sun, and Nianfeng Li. "Smart contract vulnerability detection based on automated feature extraction and feature interaction." *IEEE Transactions on Knowledge and Data Engineering* 36, no. 9 (2023): 4916-4929.

[23] Chen, Ting, Xiaoqi Li, Xiapu Luo, and Xiaosong Zhang. "Under-optimized smart contracts devour your money." In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)*, pp. 442-446. IEEE, 2017.

[24] Tann, Wesley Joon-Wie, Xing Jie Han, Sourav Sen Gupta, and Yew-Soon Ong. "Towards safer smart contracts: A sequence learning approach to detecting security threats." *arXiv preprint arXiv:1811.06632* (2018).

[25] Zhuang, Yuan, Zhenguang Liu, Peng Qian, Qi Liu, Xiang Wang, and Qinming He. "Smart contract vulnerability detection using graph neural networks." In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pp. 3283-3290. 2021.

[26] Hoffmann, Romuald, Maciej Kiedrowicz, and Jerzy Stanik. "Risk management system as the basic paradigm of the information security management system in an organization." In *MATEC Web of Conferences*, vol. 76, p. 04010. EDP Sciences, 2016.

[27] Shameli-Sendi, Alireza, Rouzbeh Aghababaei-Barzegar, and Mohamed Cheriet. "Taxonomy of information security risk assessment (ISRA)." *Computers & security* 57 (2016): 14-30.

UNCORRECTED PROOF

[28] Cherdantseva, Yulia, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby, and Kristan Stoddart. "A review of cyber security risk assessment methods for SCADA systems." *Computers & security* 56 (2016): 1-27.

[29] Lyu, Xiaorong, Yulong Ding, and Shuang-Hua Yang. "Safety and security risk assessment in cyber-physical systems." *IET Cyber-Physical Systems: Theory & Applications* 4, no. 3 (2019): 221-232.

[30] Nurse, Jason RC, Sadie Creese, and David De Roure. "Security risk assessment in Internet of Things systems." *IT professional* 19, no. 5 (2017): 20-26.

[31] Gray, Carey. "Information Security Policies, Procedures and Standards: Guidelines for Effective Information Security Management." *ITNOW* 45, no. 2 (2003): 30-30.

[32] Landoll, Douglas. *The security risk assessment handbook: A complete guide for performing security risk assessments*. CRC press, 2021.

[33] Gates, Christopher S., Ninghui Li, Hao Peng, Bhaskar Sarma, Yuan Qi, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. "Generating summary risk scores for mobile applications." *IEEE Transactions on dependable and secure computing* 11, no. 3 (2014): 238-251.

[34] Deypir, Mahmood, and Abbas Horri. "Instance based security risk value estimation for Android applications." *Journal of information security and applications* 40 (2018): 20-30.

[35] Deypir, Mahmood. "Entropy-based security risk measurement for Android mobile applications: M. Deypir." *Soft Computing* 23, no. 16 (2019): 7303-7319.

[36] Deypir, Mahmood, and Toktam Zoughi. "Novel Security Metrics for Identifying Risky Unified Resource Locators (URLs)." *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 48, no. 3 (2024): 1209-1227.

[37] Hajizadeh, Rassoul, and Mohammad Ali Hosseinzadeh. "A Classifier Based on K-Nearest Neighbors Using Weighted Summation of Reconstruction Errors." *Journal of Modeling in Engineering* 22, no. 76 (2024): 55-68. (in Persian).

[38] Moussaoui, Meroua, Emmanuel Bertin, and Noel Crespi. "Blockchain and Smart Contracts for Telecommunications: The Whys and Wherefores." In *Blockchain and Smart-Contract Technologies for Innovative Applications*, pp. 259-279. Cham: Springer Nature Switzerland, 2024.

[39] Faraji, Amir, and Shima Homayoun Arya. "Proposing an Integrated Time-Cost Management Model based on Building Information Modeling (BIM) and Blockchain Technology (BCT) Smart contract Development Approach in the Construction Industry." *Journal of Modeling in Engineering* 21, no. 74 (2023): 191-206. (in Persian).

[40] Zarbafi, Sahar, Kourosh Kiani, and Razieh Rastgoo. "Spoken Persian digits recognition using deep learning." *Journal of Modeling in Engineering* 21, no. 74 (2023): 163-172. (in Persian).

[41] Bagherzadeh, Fahimeh, and Razieh Rastgoo. "Deepfake image detection using a deep hybrid convolutional neural network." *Journal of Modeling in Engineering* 21, no. 75 (2023): 19-28. (in Persian).

[42] Gupta, Namya Aankur, Mansi Bansal, Seema Sharma, Deepti Mehrotra, and Misha Kakkar. "Detection of vulnerabilities in blockchain smart contracts using deep learning." *Wireless Networks* 31, no. 1 (2025): 201-217.

[43] Bresil, Michael, Pwc Prasad, Md Shohel Sayeed, and Umar Ali Bukar. "Deep Learning-based Vulnerability Detection Solutions in Smart Contracts: A Comparative and Meta-Analysis of Existing Approaches." *IEEE Access*, no. 13 (2025):1-26.

UNCORRECTED PROOF