Journal of Information and Communication Technology

Volume 16, Issue 59-60, Spring and Summer 2024, pp. 93-107

# Intrusion Detection Based on Cooperation on the Permissioned Blockchain Platform in the Internet of Things Using Machine Learning

**Mohammad Mahdi Abdian[1*], Majid Ghayouri Sales[2], Seyed Ahmad Eftekhari[3]**

[1] Master's Degree In Secure Computing, Computer Department, Imam Hossein University, Tehran, Iran.
[2] Assistant Professor, Computer Department, Imam Hossein University, Tehran, Iran.
[3] Bachelor Degree In Software Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran.

Received: 17 December 2022, Revised: 22 January 2023, Accepted: 05 April 2023
Paper type: Research

**Abstract**

Intrusion detection systems seek to realize several objectives, such as increasing the true detection rate, reducing the detection time, reducing the computational load, and preserving the resulting logs in such a way that they cannot be manipulated or deleted by unauthorized people. Therefore, this study seeks to solve the challenges by benefiting from the advantages of blockchain technology, its durability, and relying on IDS architecture based on multi-node cooperation. The proposed model is an intrusion detection engine based on the decision tree algorithm implemented in the nodes of the architecture. The architecture consists of several connected nodes on the blockchain platform. The resulting model and logs are stored on the blockchain platform and cannot be manipulated. In addition to the benefits of using blockchain, reduced occupied memory, the speed, and time of transactions are also improved by blockchain. In this research, several evaluation models have been designed for single-node and multi-node architectures on the blockchain platform. Finally, proof of architecture, possible threats to architecture, and defensive ways are explained. The most important advantages of the proposed scheme are the elimination of the single point of failure, maintaining trust between nodes, and ensuring the integrity of the model, and discovered logs.

**Keywords**: Intrusion Detection, Blockchain, Internet Of Things, Machine Learning, Intrusion Detection Based On Machine Learning.

[*] Corresponding Author's email: G9713671809@ihu.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره های ۵۹ و ۶۰، بهار و تابستان ۱۴۰۳، صفحه ۹۳ الی ۱۰۷

# تشخیص نفوذ مبتنی بر همکاری در بستر زنجیره‌ی‌بلوکی دارای مجوز در اینترنت‌اشیاء به روش یادگیری ماشین

**محمد مهدی عبدیان[1*]، مجید غیوری ثالث[2]، سید احمد افتخاری[3]**

[1]کارشناسی ارشد رایانش امن، گروه کامپیوتر دانشگاه جامع امام حسین (ع)، تهران، ایران

[2]استادیار گروه کامپیوتر دانشگاه جامع امام حسین (ع)، تهران، ایران

[3]کارشناسی مهندسی کامپیوتر نرم‌افزار، دانشگاه آزاد اسلامی واحد تهران مرکز، تهران، ایران

تاریخ دریافت: ۱۴۰۱/۰۹/۲۶ تاریخ بازبینی: ۱۴۰۱/۱۱/۰۲ تاریخ پذیرش: ۱۴۰۲/۰۱/۱۶

نوع مقاله: پژوهشی

## چکیده

در سیستم‌های تشخیص نفوذ، افزایش نرخ تشخیص‌های درست و کاهش زمان آموزش و تشخیص، کاهش بار پردازشی، نگهداشت مناسب مدل تشخیص‌دهنده و لاگ‌های حاصل، به طوری که توسط افراد غیرمجاز قابل دستکاری یا پاک شدن نباشند حائز اهمیت می‌باشد. بنابراین در این پژوهش، با بهره‌مندی از مزایای زنجیره‌بلوکی و قابلیت ماندگاری آن و با بهره‌مندی از معماری IDS مبتنی بر همکاری چند گره به دنبال رفع مشکلات مطرح شده می‌باشیم. مدل بر اساس الگوریتم درخت تصمیم است که در گره‌های معماری به عنوان موتور تشخیص نفوذ فعالیت می‌کند. معماری متشکل از چندین گره مرتبط در بستر زنجیره‌بلوکی می‌باشد، مدل و لاگ‌های ایجاد شده در بستر زنجیره‌بلوکی ذخیره شده و لذا به راحتی قابل دستکاری یا پاک شدن نیستند. کنار مزایای حاصل از به کارگیری زنجیره‌بلوکی، مساله‌ی میزان حافظه اشغالی و سرعت و زمان انجام تراکنش‌ها توسط زنجیره‌بلوکی نیز مطرح می‌باشند. در این پژوهش مدل‌های ارزیابی برای معماری تک گره و چند گره در بستر زنجیره‌بلوکی، مطرح شده است. در نهایت اثبات معماری و تهدیدات احتمالی نسبت به معماری و راه‌های دفاع تشریح می‌شود. مهمترین مزایای طرح شامل حذف نقطه‌ی شکست واحد، حفظ اعتماد بین گره‌ها و اطمینان از جامعیت مدل و لاگ‌های کشف شده می‌باشد.

**کلیدواژگان**: تشخیص نفوذ، زنجیره‌بلوکی، اینترنت‌اشیاء، یادگیری ماشین، تشخیص نفوذ مبتنی بر یادگیری ماشین.

* رایانامهٔ نویسنده مسؤول: G9713671809@ihu.ac.ir

## ۱- مقدمه

با رشد سریع اینترنت و شبکه‌های مبتنی بر آن، تنش‌های مربوطه نیز افزایش یافته است که یکی از اصلی‌ترین این تنش‌ها امنیت در برابر حملات و نفوذ به شبکه می‌باشد. بدین ترتیب، سیستم‌های تشخیص نفوذ به عنوان نگهبان سیستم‌های کامپیوتری باید توانایی شناسایی و دفاع را در زمان بسیار کوتاه داشته باشند [۱].

مشکلی که روش‌های تشخیص نفوذ رایج و سنتی دارند این است که با داشتن تک گره تشخیص‌دهنده نفوذ که در روش‌های رایج مطرح بوده و هست، دچار ضعف نقطه شکست واحد می‌باشند و این موضوع با زیرساخت اینترنت‌اشیاء که ذاتاً غیرمتمرکز هست، مناسب و همه جانبه نخواهند بود. و با از دسترس خارج شدن این تک گره تشخیص‌دهنده، کل مکانیزم تشخیص از کار می‌افتد، لذا برای مقابله با این مشکلات و همچنین تشخیص حملات پیچیده و توزیع شده مانند حمله‌ی انکار سرویس توزیع شده، نیاز است که از روش‌های تشخیص نفوذ مبتنی بر همکاری استفاده کرد تا چندین گره‌ها تشخیص‌دهنده از تمام ترافیک زیرساخت، اطلاعات جامع‌تر و کامل‌تری در اختیار داشته باشند، این شیوه برای کشف نفوذ در اینترنت و مشخصاً در اینترنت‌اشیاء، رویکرد نسبتاً جدیدی است. همچنین مورد دیگری که روش‌های رایج تک گره و یا روش‌های مبتنی بر همکاری موجود در اینترنت‌اشیاء دارند، این است که به دلیل سبک وزن‌تر بودن و هزینه‌های پردازشی و زمانی کمتر که مناسب برای دستگاه‌های اینترنت‌اشیاء می‌باشد، متکی به روش‌های مبتنی بر امضاء می‌باشند که بر این اساس صرفاً قادر به تشخیص حملات از پیش شناخته شده، با امضای مشخص و از قبل موجود در پایگاه داده‌ی امضاهای خود خواهند بود.

یاناونگ سوی و همکاران در [۲] به پیاده‌سازی یک روش بهینه و سبک وزن مبتنی بر یادگیری ماشین برای دستگاه‌های اینترنت‌اشیاء پرداخته‌اند و به طور خاص پیاده سازی این سیستم را در ماشین raspberry pi انجام داده‌اند. این سامانه‌ی تشخیص نفوذ به صورت منفرد روی یک دستگاه اینترنت‌اشیاء پیاده‌سازی شده و عمل می‌کند. اوساما آلکادی و همکاران در [۳] به تشخیص نفوذ در بستر ارتباطی اینترنت و ابر با دید همکاری بین اجزا پرداخته و از روش یادگیری عمیق با محوریت الگوریتم BiLSTM استفاده کرده‌اند. آن‌ها همچین برای ارزیابی کار خود از مجموعه داده‌ی ذکر شده در مقاله‌ی قبل یعنی UNSW-NB15 استفاده کرده و از زنجیره‌ی بلوکی اتریوم در کار خود بهره برده‌اند. گانتور هارما پوترا و همکاران در یک [۴] طرح و شمای کلی از سیستم تشخیص نفوذ مبتنی بر همکاری در اینترنت‌اشیاء با استفاده از زنجیره‌بلوکی را بیان کرده‌اند که در آن از زنجیره‌بلوکی عمومی اتریوم برای حفظ امنیت و جامعیت داده‌های بین گره‌ها استفاده شده است

ویژی منگ و همکاران در [۵] به بررسی کاربرد زنجیره‌ی بلوکی در سیستم‌های تشخیص نفوذ مبتنی بر همکاری بین گره‌ها در اینترنت‌اشیاء پرداخته‌اند و دو مساله‌ی مهم مدیریت اعتماد و اشتراک‌گذاری داده را بررسی کرده‌اند.. سنا یاکوت و همکاران در [۶] به معرفی و پیاده سازی زنجیره‌ی بلوکی دارای مجوز در اینترنت‌اشیاء و به طور خاص پیاده سازی زنجیره‌ی بلوکی هایپرلجرفابریک در دستگاه raspberry pi پرداخته‌اند و قابلیت‌های امنیتی و حریم خصوصی که از این کار حاصل می‌شود را مورد مطالعه قرار داده‌اند. هین تای تو ترونگ و همکاران در [۷] به معرفی و پیاده سازی زنجیره‌ی بلوکی دارای مجوز هایپرلجرفابریک در اینترنت‌اشیاء پرداخته و سپس قابلیت‌های مختلف آن از جمله امنیت، حریم خصوصی، اشتراک داده و کنترل دسترسی را مورد تجزیه و تحلیل قرار داده‌اند. الساندرو اسفوزین و همکاران در [۸] از اسنورت به عنوان سیستم تشخیص نفوذ در دستگاه‌های رزبری‌پای استفاده کرده‌اند که پیاده‌سازی سبک وزن و مناسبی است، اما با توجه به اینکه برنامه‌ی اسنورت برای تشخیص از روش‌های مبتنی بر امضا استفاده می‌کند لذا نمی‌تواند حملات جدید را به خوبی کشف کند.

چاندراسخار و همکاران در [۹] با استفاده از شبکه‌های عصبی-فازی و ماشین بردار پشتیبان (SVM)، یک سیستم تشخیص نفوذ ارائه کرده‌اند که با معماری چهار لایه، عمل طبقه بندی را بصورت سلسله مراتبی انجام می‌دهد. اقبال و همکاران در [۱۰] به طبقه‌بندی و تشخیص و پیشگیری از نفوذ به‌عنوان یک سرویس در حملات امنیتی ابر پرداخته‌اند. در این پژوهش برای مقابله با این حملات یک چارچوب همکاری سیستم تشخیص نفوذ ارائه شده است. در این پژوهش به اهمیت تشخیص نفوذ و پیشگیری به‌عنوان یک سرویس مشخص پرداخته شده‌است. میهیتری و همکاران در [۱۱] سیستم‌های تشخیص نفوذ مبتنی بر یادگیری گروهی برای ابرها را بررسی کرده‌اند. سیستم‌های تشخیص نفوذ مبتنی بر یادگیری گروهی برای ابرها به طور کلی به دو بخش اصلی انتخاب ویژگی و الگوریتم یادگیری تقسیم می‌شود.

تحقیق حاضر به این دلیل حائز اهمیت است که در آن به دلیل استفاده از روش یادگیری ماشین سبک وزن و بهینه، به طور همزمان امکان تشخیص حملات جدید و پیچیده در دستگاه‌های کم توان اینترنت‌اشیاء میسر می‌شود و همچنین به علت بهره‌مندی از

زنجیره‌ی بلوکی خصوصی، حریم خصوصی اجزا حفظ شده، از حملات داخلی بین گره‌ها جلوگیری می‌شود و همچنین اطلاعات و لاگ‌های شناسایی شده پایدار و ماندگار در سیستم باقی می‌مانند.

با این حال استفاده از روش‌های مبتنی بر همکاری با مشکلاتی چون حملات داخلی و متخاصمانه رو به رو می‌باشند که برای جلوگیری از این حملات و حفظ اعتماد بین گره‌ها و حفظ جامعیت داده‌ها، حفظ جامعیت مدل تشخیص‌دهنده و حفظ جامعیت هشدار (لاگ) های به اشتراک گذاشته شده بین گره‌ها (حذف یا تخریب لاگ‌ها و اطلاعات شناسایی شده که در عملیات پس از سوءاستفاده مورد حمله قرار می‌گیرند)، راه‌کاری نیاز است تا این اطلاعات در یک جا متمرکز نباشند و و به راحتی قابل حذف نباشند (نقطه واحد شکست نداشته باشیم)، لذا برای این منظور زنجیره‌ی‌بلوکی می‌تواند یک راهکار مناسب باشد که مورد توجه قرار گرفته است.

## ۲- مبانی نظری پژوهش

### ۲-۱- اینترنت‌اشیاء

واژه‌ی اینترنت‌اشیاء نخستین بار در سال ۱۹۹۹ میلادی توسط کوین اشتون مطرح شد [۱۲] اینترنت‌اشیاء به طور کلی از اتصال دستگاه‌های مختلف و ناهمگون (غیرهم‌جنس) به یکدیگر شکل می‌گیرد و طبق تعریف عبارت است از: «شبکه‌ای از اتصال انواع گوناگون موجودیت‌ها و در دسترس‌پذیری این موجودیت‌ها در هر مکان و هر زمانی». برای مثال موجودیت‌های اینترنت‌اشیاء عبارت‌اند از: خانه‌های هوشمند، اتوموبیل‌های هوشمند، وسایل پوشیدنی مانند ساعت‌های هوشمند، موبایل‌های هوشمند و موارد متنوع و گوناگون دیگر [۱۳].

مؤسسات تحقیقاتی از جمله موسسه گارتنر و کمپانی سیسکو پیش‌بینی می‌کنند که تعداد گره‌ها در اینترنت‌اشیاء به مرور زمان به صورت تصاعدی بیشتر می‌شوند [۱۳] معماری اینترنت‌اشیاء در چندین لایه شکل گرفته است که اکثر منابع آن را به سه لایه با نام‌های: لایه فیزیکی (ادراکی)، لایه‌ی شبکه (انتقال) و لایه‌ی کاربردی تقسیم می‌کنند، برخی دیگر از منابع این معماری را به چهار لایه تقسیم می‌کنند [۱۴]. که عبارت است از: لایه فیزیکی (ادراکی)، لایه‌ی شبکه (انتقال)، لایه‌ی میانی و لایه‌ی کاربردی، در بخش‌های پیش‌رو به توضیح هر کدام از این لایه‌ها پرداخته شده است.

### ۲-۲- زنجیره‌ی بلوکی

زنجیره بلوکی یک پایگاه داده باز است که یک دفتر کل توزیع شده را که معمولاً در یک شبکه همتا به همتا مستقر می‌شود، نگهداری می‌کند. توسط یک لیست از رکوردها به نام بلوک که حاوی تراکنش است و به طور مداوم در حال رشد می‌باشد، تشکیل شده است [۱۵]. زنجیره‌ی بلوکی یک تکنولوژی نوظهور و مؤثر است که یکی از معروف‌ترین اثرات آن، پدید آوردن انقلابی در رمزارزها بوده است، این تکنولوژی اساساً از یک پایگاه داده توزیع شده‌ی امن که با نام دفتر کل مرکزی شناخته می‌شود، تشکیل شده است. این پایگاه داده حاوی اطلاعات تراکنش‌های مختلف خانواده‌ی زنجیره‌ی بلوکی می‌باشد، و تمام تراکنش‌ها و اعتبار سنجی‌ها در این جا اتفاق می‌افتند. برای مثال زمانی که یک تراکنش قرار است بین دو نقطه صورت گیرد، گره مبدأ این درخواست تراکنش را برای تمام گره‌های موجود در زنجیره‌ی‌بلوکی ارسال می‌کند، سپس هر گره به صورت دوره‌ای مجوعه‌ای از تراکنش‌ها را جمع آوری می‌کند و آنها را در یک بلوک گروه‌بندی می‌کند برای تأیید و ثبت هر تراکنش در این دفتر کل، نیاز به توافق بین اکثریت مشارکت‌کنندگان در شبکه است؛ به طوری که هر تراکنش یا اطلاعات مربوط به آن، پس از ورود به دفتر کل، هرگز نمی‌تواند پاک شود و یا تغییر کند [۱۶] و [۱۷]. زنجیره بلوکی را می‌توان پایگاه داده‌ای در نظر گرفت که اطلاعات، صرفاً به آن اضافه می‌شود و شبکه‌ای از اعضای همتابه‌همتا از آن نگهداری می‌کنند.

### ۲-۳- سیستم تشخیص نفوذ

تشخیص نفوذ به معنای شناسایی استفاده غیرمجاز یا حملات به یک سیستم یا شبکه است. یک سیستم تشخیص نفوذ برای شناسایی و سپس برای منحرف کردن یا بازدارندگی (در صورت امکان) چنین حملاتی طراحی و استفاده می‌شود. مانند دیواره‌های آتش، سیستم‌های تشخیص نفوذ می‌توانند مبتنی بر نرم‌افزار باشند و یا می‌توانند سخت‌افزار و نرم‌افزار را (به شکل دستگاه‌های سیستم تشخیص نفوذ مستقل از پیش نصب‌شده و از پیش پیکربندی‌شده) ترکیب کنند. اغلب، نرم‌افزار سیستم تشخیص نفوذ روی همان دستگاه‌ها یا سرورهایی اجرا می‌شود که دیواره‌های آتش، پراکسی‌ها یا سایر سرویس‌های مرزی شبکه در آنجا کار می‌کنند. اگر یک سیستم تشخیص نفوذ روی همان دستگاه یا سروری که دیواره آتش یا سایر سرویس‌ها در آن نصب شده است اجرا نشود، باید آن دستگاه‌ها را به دقت کنترل کند. اگرچه چنین دستگاه‌هایی تمایل دارند در حاشیه شبکه کار کنند، سیستم‌های تشخیص نفوذ

می‌توانند حملات داخلی و همچنین حملات خارجی را شناسایی کرده و با آنها مقابله کنند [۱۸].

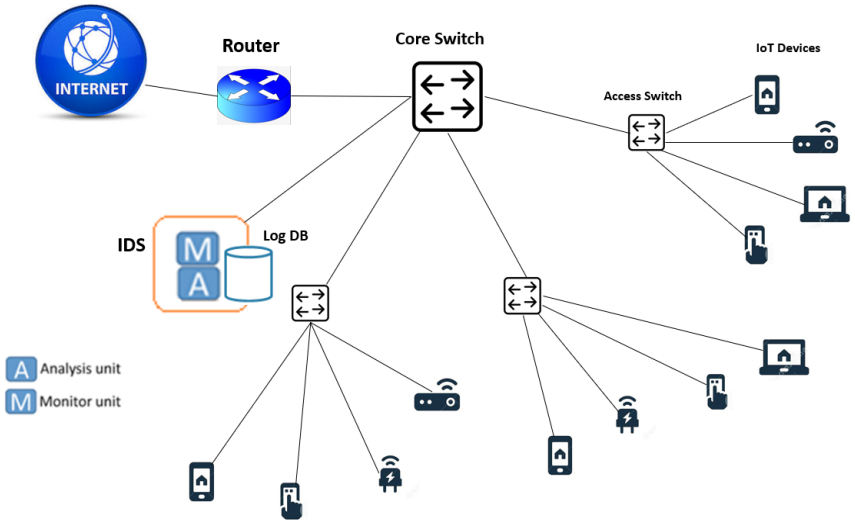
## ۳- راه حل پیشنهادی و پیاده‌سازی

### ۳-۱- معماری تک گره سیستم تشخیص نفوذ در اینترنت اشیاء

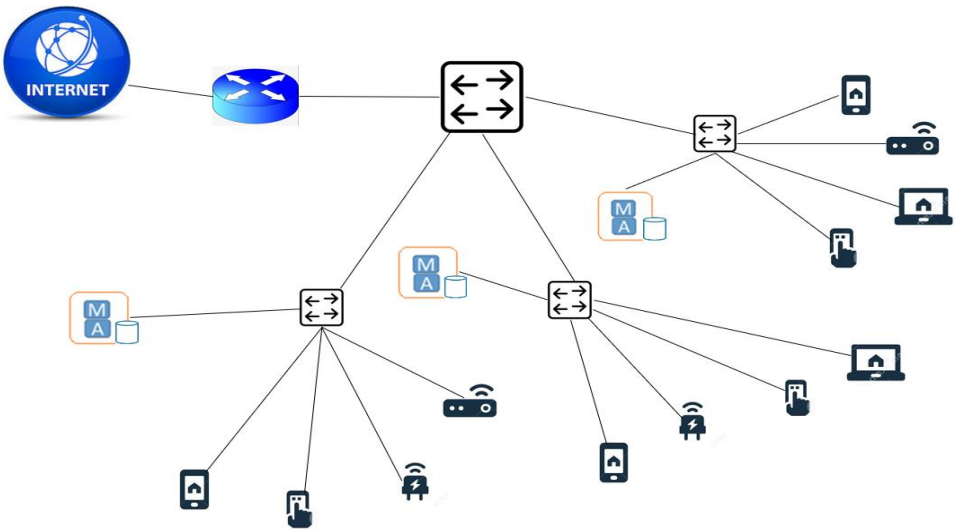
معماری سیستم تشخیص نفوذ تک گره نمایش داده شده در شکل ۱ به عنوان یک نقطه شکست واحد در شبکه لحاظ می‌شود و اگر از دسترس خارج شود یا توسط حمله کنندگان تخریب شده و یا تغییر پیدا کند، مکانیزم امنیتی پایش (تحلیل) ترافیک و تشخیص نفوذ از بین رفته و متوجه این موضوع نیز نخواهیم شد. همچنین اطلاعات و لاگ‌های شناسایی شده در این معماری در پایگاه داده محلی این سیستم منفرد ذخیره می‌شوند.

### ۳-۲- معماری چند گره سیستم تشخیص نفوذ

استفاده از چندین گره برای تشخیص نفوذ پیشنهاد می‌شود که همانطور که مطابق شکل ۲ مشخص می‌باشد، با این کار بار پردازشی گره‌های تحلیل کننده (تشخیص‌دهنده) بین گره‌های متعدد توزیع می‌شود، همچنین نقطه شکست واحد حذف می‌شود و با از کار افتادن یکی از سیستم‌ها کل عملیات تشخیص نفوذ از بین نمی‌رود.




**شکل ۱. معماری تک گره سیستم تشخیص نفوذ**




**شکل ۲. معماری پیشنهادی با چندین گره تشیخص‌دهنده نفوذ (به ازای هر زیر شبکه یک گره تشخیص‌دهنده)**

## ۳-۳- معماری چند گره سیستم تشخیص نفوذ در بستر هایپرلجرفابریک

مطابق شکل ۳ معماری سیستم تشخیص نفوذ مبتنی بر همکاری در بستر زنجیره‌ی بلوکی دارای مجوز در اینترنت اشیاء به روش یادگیری ماشین که هدف این پژوهش می‌باشد را ارائه شد. در این روش، تحلیل حاصل از بررسی ترافیک (لاگ‌ها) برای داشتن دید جامع از کل زیرساخت توسط تک گره‌ها، در قالب تراکنش‌های زنجیره‌بلوکی بین گره‌های تشخیص‌دهنده به اشتراک گذاشته می‌شود. از کنار هم قرار دادن لاگ‌های زیرشبکه‌های مختلف برای کاربردهای جرم‌یابی/حسابرسی[1] با دید جامع و تجمیع[2] لاگ‌ها استفاده می‌شود. همچنین به دلیل به اشتراک گذاری این اطلاعات و ذخیره‌ی آن‌ها در بستر زنجیره بلوکی از حذف لاگ‌ها طی حملات پس از سوءاستفاده جلوگیری می‌شود.

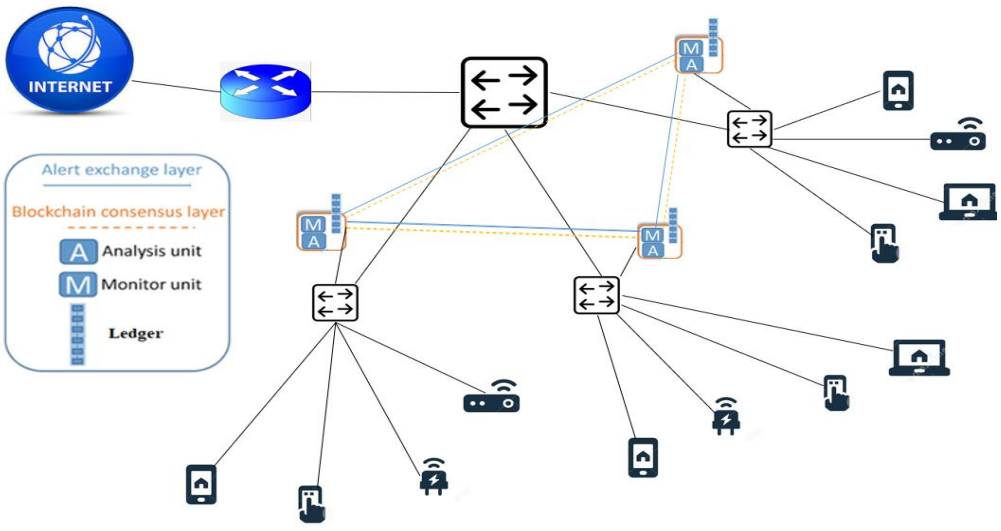
## ۴-۳- تشریح و پیاده‌سازی معماری تک گره سیستم تشخیص نفوذ پیشنهادی

مجموعه داده[3] استفاده شده در این پژوهش، مجموعه داده معروف UNSW-NB15 می‌باشد و دارای حالت‌ها و حملات مختلف در اینترنت‌اشیاء و اینترنت رایج می‌باشد که براساس [۱۹] و به طور کلی دارای ۴۷ ویژگی است که در ردیف‌های ۱ تا ۴۷ ذکر شده‌اند، که مقادیر برخی از آن‌ها غیرعددی و مقادیر برخی دیگر عددی می‌باشند، همچنین علاوه بر داشتن برچسب حمله یا ترافیک حمله، دارای ۹ نوع مختلف حمله می‌باشد آخرین ردیف از این جدول دو برچسب ترافیک سالم و یا مخرب را بیان می‌کند. همچنین انواع حملات موجود در مجموعه داده UNSW-NB15 نیز براساس [۱۹] می‌باشد. مجموعه‌داده در قالب دو فایل CSV با نام‌های train-set.csv و test-set.csv نیز برای نمونه‌های آموزش و تست از [۱۹] و [۲۰] استفاده شدند.

برای کاهش پردازش به علت کثرت ویژگی‌های مجموعه داده، ابتدا تعدادی از ویژگی‌ها را براساس الگوریتم انتخاب ویژگی مبتنی بر همکاری[4] استفاده شده در پژوهش [۲] بر می‌گزینیم که این ۷ ویژگی مطابق جدول ۱ است.

**جدول ۱. ویژگی‌های انتخاب شده از UNSW-NB15 با استفاده از CFS برای یادگیری و تست مدل**

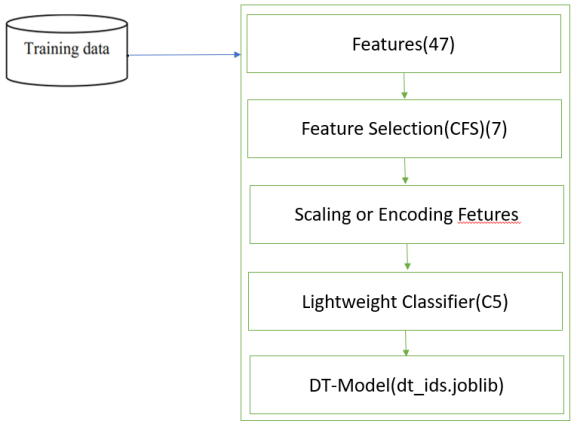
| No. | Name | Type | Description |
|---|---|---|---|
| 1 | sbytes | Integer | Source to destination transaction bytes |
| 2 | sttl | Integer | Source to destination time to live value |
| 3 | dttl | Integer | Destination to source time to live value |
| 4 | service | nominal | http, ftp, smtp, ssh, dns, ftp-data,irc and (-) if not much used service |
| 5 | Sload | Float | Source bits per second |
| 6 | ct_srv_dst | integer | No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26). |
| 7 | ct_dst_sport_ltm | integer | No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26). |




**شکل ۳. معماری پیشنهادی سیستم تشخیص نفوذ مبتنی بر همکاری در بستر زنجیره‌بلوکی در اینترنت‌اشیاء**

[1] Audit
[2] Aggregation
[3] Data Set
[4] CFS: Collaborative Feature Selection

معماری سیستم تشخیص نفوذ منفرد بر اساس الگوریتم درخت تصمیم که نوعی الگوریتم یادگیری ماشین است در شکل ۴ قابل مشاهده می‌باشد.




**شکل ۴. ساخت مدل سیستم تشخیص‌دهنده نفوذ با الگوریتم درخت تصمیم**

با به کارگیری این مدل تشخیص نفوذ در سیستم‌های اینترنت‌اشیاء مورد نظر که شبیه‌سازی از دستگاه رزبری پای نسخه ۴ با دو هسته سی‌پی‌یو و ۴ گیگ رم می‌باشند، معماری تشخیص نفوذ مبتنی بر همکاری مد نظر خود را شکل می‌دهیم.

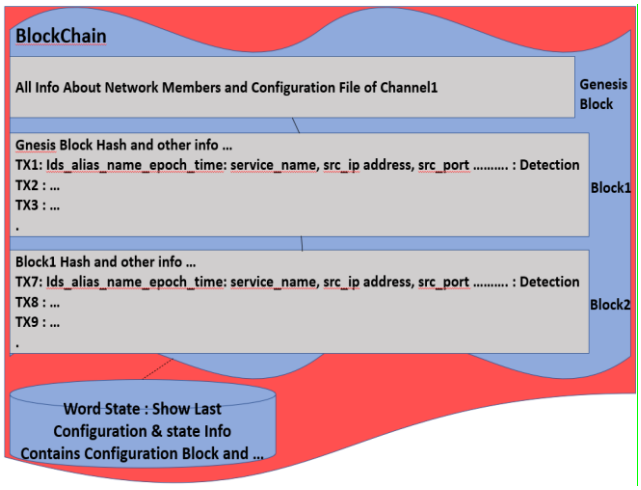
هر دستگاه اینترنت‌اشیاء و در نتیجه کارت شبکه‌ی آن در زیر شبکه‌ای قرار دارد و این کارت شبکه در حالت گوش فرا دادن به کل ترافیک در حالت بی‌قیدوشرط می‌باشد، لذا ترافیک رفت و آمد شده در زیرشبکه توسط این کارت شبکه شنود (پایش ) می‌شود و توسط برنامه‌ی آرگوس (برنامه‌ای مشابه برنامه تی‌سی‌پی‌دامپ که برای خواندن و فیلتر کردن ترافیک گذری از کارت شبکه مورد استفاده قرار می‌گیرد) به سیستم تشخیص نفوذ که به خروجی برنامه argus مربوطه گوش فرا می‌دهد ارسال می‌شود.

همانطور که در شکل ۵ هم مشخص هست در این دفترکل توزیع شده مقدار چکیده مدل به همراه سایر مشخصات کانال مربوطه‌اش ذخیره شده است.

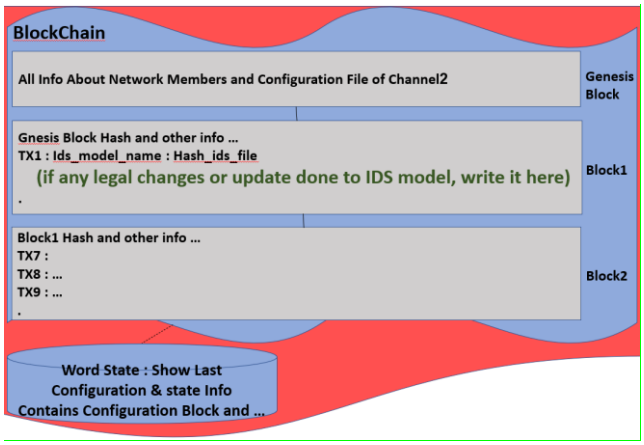
## ۳-۵- تشریح ارتباطات برنامه کاربری هایپرلجرفابریک و مدل تشخیص‌دهنده

مطابق شکل ۶ برنامه کاربردی هایپرلجرفابریک خود را با استفاده از جاوا اسپرینگ بوت[1] و ماژول‌های آن، از جمله امنیت اسپرینگ[2] و میکروسرویس اسپرینگ[3] برای ارائه‌ی سرویسی تحت واسط برنامه کاربری رست[4] مطابق شکل به طریقی طراحی می‌گردد که استفاده کننده خود (در این جا سیستم تشخیص‌دهنده نفوذ) را از طریق Spring Security احراز هویت کرده و به این سیستم تشخیص‌دهنده نفوذ توکنی[5] را تخصیص دهد که برای درخواست‌های آتی خود از این توکنی که از برنامه کاربردی دریافت کرده در سرآیند درخواست[6] خود استفاده می‌کند

همانطور که در شکل ۶ نیز ذکر شده است مدل خود را با استفاده از ابزار درهم‌سازی[7] با استفاده از کتابخانه Pyarmor غیر قابل خواندن و نفوذ ناپذیر می‌کنیم.




(الف)

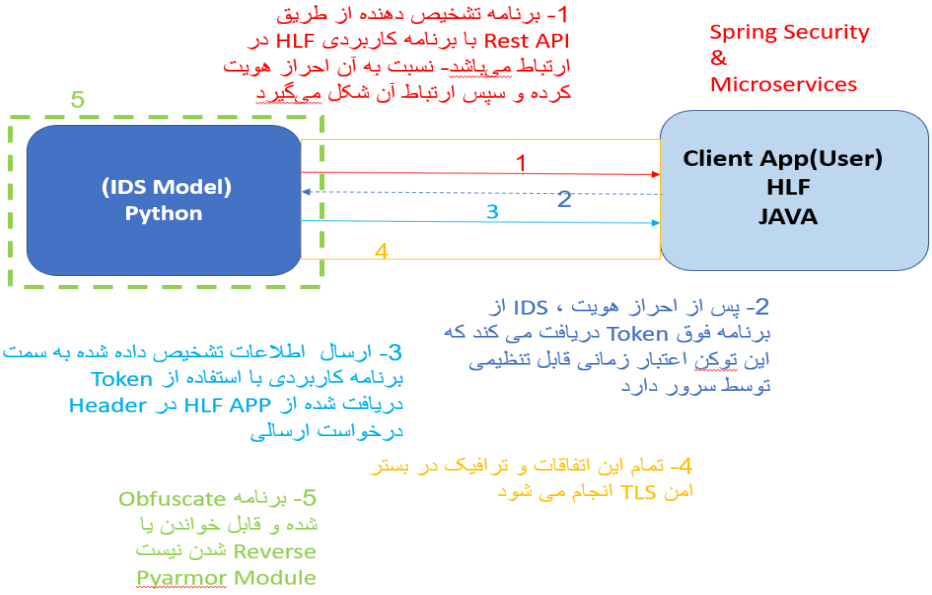



(ب)

**شکل ۵. الف) ساختار دفترکل توزیع شده کانال اول، ب) ساختار دفترکل توزیع شده کانال دوم**

[1]Java Spring Boot
[2]Spring-Security
[3]Spring-Microservice
[4]REST API

[5]Token
[6]Header Request
[7]Obfuscation




**شکل ۶. مراحل ارتباطی مدل تشخیص‌دهنده و برنامه کاربردی هایپرلجرفابریک و مکانیزم‌های امنیتی مربوطه**

## ۳-۶- معماری پیشنهادی با نظر به کانال اول هایپرلجرفابریک

در شکل ۷ توضیحات و اجزای شرح داده شده در کانال اول قابل مشاهده می‌باشند. نمادهای مختص به زنجیره‌ی بلوکی شامل کانال، سرویس عضویت و صدور گواهی و سازمان، مدیر و نیز برنامه کاربردی، همتا، کدزنجیره‌ای، دفترکل توزیع شده، سفارش‌دهنده [۲۱-۲۳] نیز قابل مشاهده است.

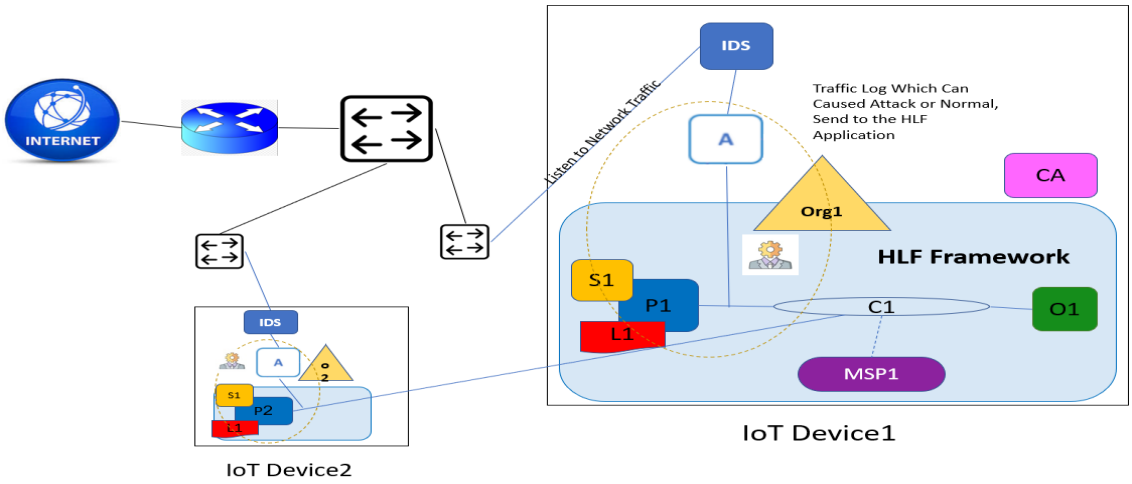
## ۳-۷- معماری پیشنهادی با نظر به کانال دوم هایپر لجرفابریک

کانال دیگری (کانال دوم) وجود دارد که دفترکل توزیع شده آن، مسئولیت ذخیره و نگهداری مقدار چکیده برنامه (مدل) تشخیص‌دهنده را دارد. این مقدار در زبان پایتون با استفاده از کتابخانه hashlib به دست می‌آید، این مقدار به‌عنوان یک تراکنش در دفتر کل توزیع شده مشترک کانال دوم ذخیره می‌شود. برنامه فراخوان برنامه کاربردی هایپرلجرفابریک این کانال که آن را بررسی کننده هش (چکیده)[1] نامیده‌ایم، وظیفه دارد به صورت دوره‌ای در بازه‌های زمانی مختلف و به صورت خودکار اجرا شده، آخرین مقدار هش ذخیره شده در دفتر کل توزیع شده را بخواند، مقدار هش مدل تشخیص‌دهنده روی میزبان خود و سایر گره‌های تشخیص‌دهنده را محاسبه کرده، آن را با مقدار هش خوانده شده مقایسه کند، اگر یکی بودند اعلان صحت عملکرد کرده و در غیر این صورت هشداری را برای مدیر ایجاد می‌کند. همانطور که در شکل مشاهده می‌شود ما این کار را با استفاده از الگوریتم SHA3_256[2] که از نسخه‌های به روز الگوریتم SHA برای محاسبه (گرفتن) چکیده می‌باشد و همچنین توسط ویرایشگر پایتون نیز به عنوان روشی امن و خوب پیشنهاد می‌شود، استفاده کرده‌ایم [۲۴].

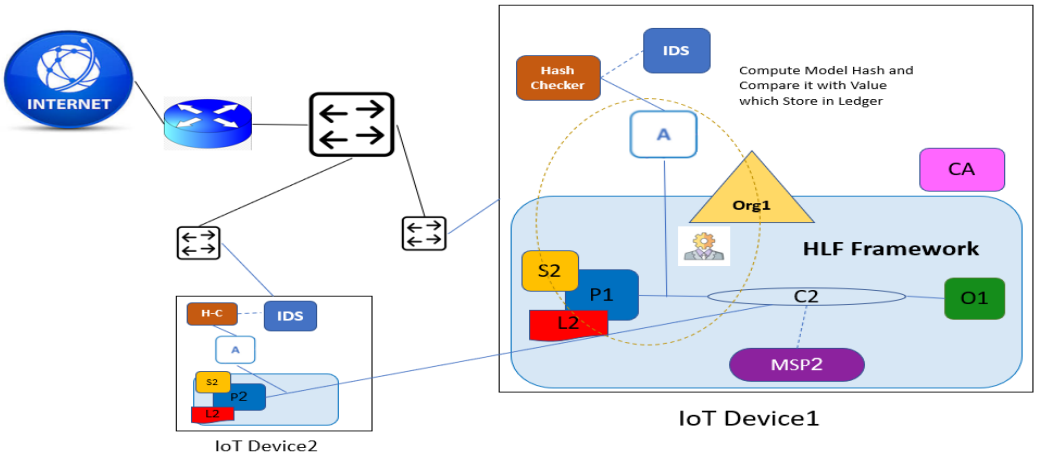
معماری سیستم با نظر به کانال دوم هایپرلجرفابریک به شکل ۸ می‌باشد.

[1]Hash-Checker

[2]Secure Hash Algorithm




**شکل ۷. معماری سیستم تشخیص نفوذ مبتنی بر همکاری در بستر زنجیره‌بلوکی با نظر به کانال اول**




**شکل ۸. معماری سیستم تشخیص نفوذ مبتنی بر همکاری در بستر زنجیره‌بلوکی با نظر به کانال دوم**

هزینه پردازشی و زمانی معماری به علت سبک وزن بودن زنجیره‌بلوکی هایپرلجرفابریک و مدل تشخیص‌دهنده بهینه، پایین است و تنها نگرانی، مساله حافظه می‌باشد، برای این منظور می‌توان روی ترافیک برای ذخیره برخی از ترافیک و عدم ذخیره برخی دیگر از ترافیک فیلتر بندی داشت، که این کار به طور محسوس حجم مورد نیاز برای ذخیره‌سازی را کاهش می‌دهد. همچنین مستندات هایپرلجرفابریک، مدیران را برای امنیت بیشتر و حفظ این اطلاعات به استفاده از کلیدهای کوبرنتیز[1] و ماژول امنیتی سخت‌افزاری[2] توصیه می‌کنند [۲۵].

## ۴- راه حل پیشنهادی و پیاده‌سازی

### ۴-۱- پارامترهای ارزیابی سیستم‌های تشخیص نفوذ

برای ارزیابی و تحلیل روش پیشنهادی مطابق پژوهش رونوئا و همکاران [۲۵] از معیارهایی که در همه‌ی سیستم‌های تشخیص نفوذ بر اساس مثبت صادق[3]، مثبت کاذب[4]، منفی کاذب[5]، منفی صادق[6] رایج است، استفاده می‌شود.

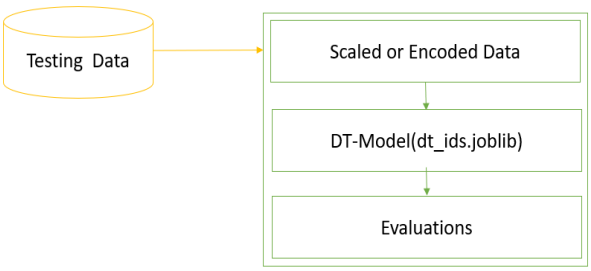
[1]Kubernetes secrets
[2]HSM: Hardware Security Module
[3]TP: True Positive
[4]FP: False Positive
[5]FN: False Negative
[6]TN: True Negative

## ۴-۲- نتایج ارزیابی کارهای پیشین

همچنین نتایج ارزیابی در حالت داشتن دو برچسب شامل حمله یا ترافیک نرمال مطابق [۲۰] بر اساس الگوریتم‌های بیز ساده، ماشین بردار پشتیبان، جنگل تصادفی و درخت تصمیم ذکر شده است. نتایج پژوهش انجام شده توسط الکادی و همکاران [۳] براساس دو برچسب شامل حمله و ترافیک سالم است که در مقایسه با نتایج این تحقیق، مشخص است درصد تشخیص و دقت گزارش شده، در این روش بهبود پیدا کرده است، این افزایش دقت هنگام استفاده از ۶۰ گره پنهان محسوس‌تر می‌باشد، اما همانطور که مشاهده می‌شود زمان تشخیص بسیار بالاتر از روش‌های یادگیری نظارت شده‌ی ذکر شده در کار پیشین می‌باشد.

## ۴-۳- مدل ارزیابی معماری تک گره با استفاده از مجموعه داده تست

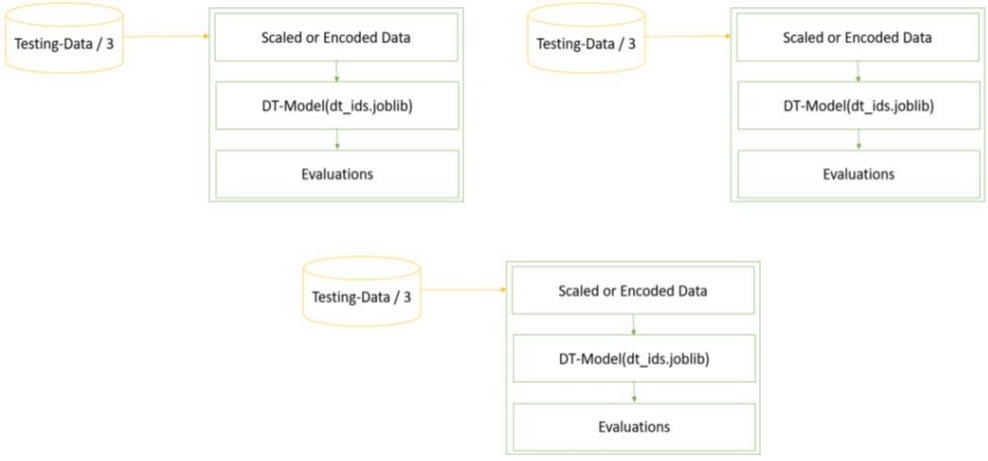
برای ارزیابی مدل و طرح ارائه شده در پژوهش صورت گرفته، مطابق شکل ۹ و با استفاده از مجموعه داده تست عمل می‌کنیم. ماشین (گره) تشخیص‌دهنده نفوذ، داده‌های آزمایش را دریافت کرده، این داده‌ها را بر اساس ۷ ویژگی ذکر شده در بخش قبلی که با استفاده از الگوریتم انتخاب ویژگی CFS گزینش شده‌اند، ارزیابی می‌کند و نتایج را گزارش می‌دهد. کل داده‌های تست در این‌جا مورد استفاده قرار می‌گیرند که همانطور که پیشتر ذکر شد شامل ۸۲۰۰۰ ردیف ترافیکی می‌باشند. نتایج ارزیابی از این حالت را به اختصار با عنوان درخت تصمیم منفرد نمایش می‌دهیم.




**شکل ۹. ارزیابی مدل ساخته شده با کل داده‌های تست**

## ۴-۴- مدل ارزیابی معماری چند (سه) گره با استفاده از مجموعه داده تست

مشخصاً برای ارزیابی مدل در حالت توزیع شده مثلاً با داشتن ۳ گره، مطابق شکل ۱۰ داده‌ی تست خود را به صورت تصادفی[1] به سه قسمت مساوی تقسیم کرده و سپس نتایج ارزیابی را قید می‌کنیم. انتظار می‌رود زمان تشخیص با توجه به تقسیم شدن داده‌های تست کمتر شده و در واقع گلوگاه[2] زمانی در این‌جا ماشینی خواهد بود که با کمترین سرعت ارزیابی تشخیص خود را به پایان می‌رساند. در این‌جا ۳/۱ از کل داده‌های تست برای هر گره مورد استفاده قرار می‌گیرند که شامل ۲۷۳۳۳ ردیف ترافیکی می‌باشند. نتایج ارزیابی از این حالت را به اختصار با عنوان درخت تصمیم با سه گره[3] نمایش می‌دهیم.

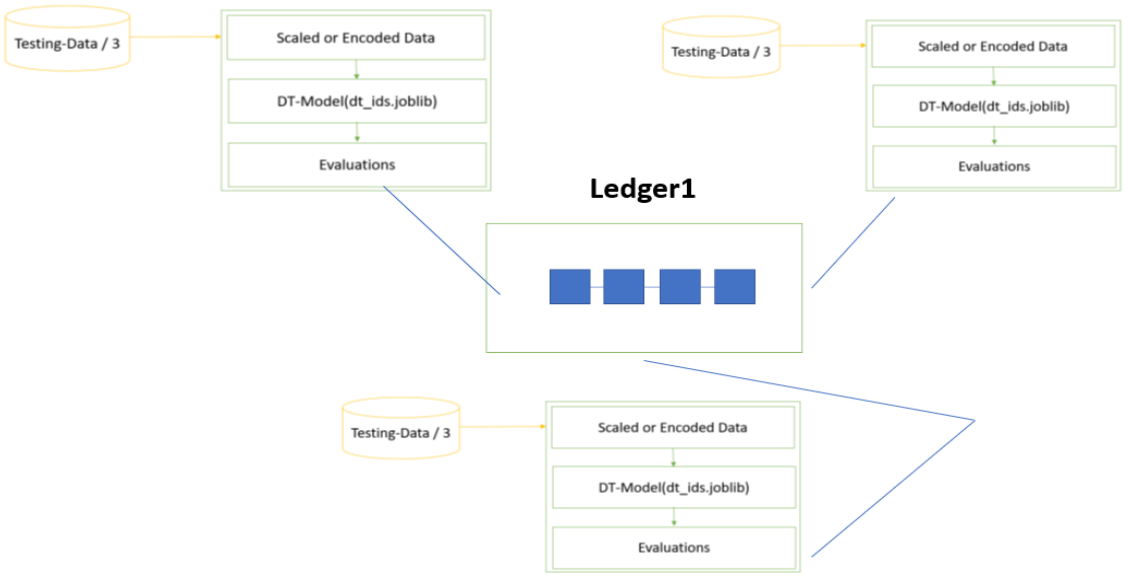



**شکل ۱۰. تقسیم داده‌های تست به سه قسمت و سپس محاسبه زمان تشخیص انجام شده**

[1] Random
[2] Bottleneck
[3] 3-DT

### ۴-۵- مدل ارزیابی معماری پیشنهادی با سه گره در بستر هایپرلجرفابریک با استفاده از مجموعه داده تست

مطابق شکل ۱۱ به ارزیابی زمان ذخیره‌سازی و انتشار[1] لاگ‌های کشف شده در قالب تراکنش‌های هایپرلجرفابریک می‌پردازیم. نتایج ارزیابی از این حالت را به اختصار با عنوان درخت تصمیم-هایپرلجرفابریک[2] نمایش داده شدکه هدف اصلی پژوهش می‌باشد.




**شکل ۱۱. تشخیص و سپس ذخیره‌سازی و انتشار لاگ‌ها و اطلاعات کشف شده روی دفترکل توزیع شده همتاها**

### ۴-۶- جداول ارزیابی بر اساس دقت تشخیص

نتایج حاصل از کار ارائه شده در این پژوهش در کنار نتایج حاصل از مقالات پیشین ذکر شده، در جداول (۲) و (۳) نمایش داده شده‌اند. مطابق جدول ۲ زنجیره‌بلوکی اتریوم به دلیل سرعت پایین تراکنش‌ها، هزینه زمانی بالایی دارد، اما مدل پیشنهادی ما به دلیل استفاده از زنجیره‌بلوکی هایپرلجرفابریک و سرعت تراکنش بالا، زمان بسیار کمتری نیاز دارد. البته قابل ذکر است که سرعت تراکنش‌های روش پیشنهادی می‌تواند با تغییر پارامترهای گره‌های سفارش‌دهنده (ترتیب‌دهنده) بسیار بالاتر برود.

### ۴-۷- تأخیر ایجاد بلوک و به اشتراک گذاری آن در هایپرلجرفابریک

در این پژوهش یک لاگ و اطلاعات حاصل از تحلیل ترافیک، حداکثر بعد از ۲ ثانیه در بستر زنجیره‌بلوکی بین همتاها و در لجر به اشتراک گذاشته شده و ثبت می‌شود و از بین بردن آن غیر ممکن خواهد بود.

### ۴-۸- زمان مصرفی ذخیره‌سازی لاگ‌ها

زمان طی شده برای تشخیص و سپس ذخیره‌سازی لاگ حملات کشف شده حاصل از تحلیل ترافیک مطابق جدول ۳ می‌باشد.

همان طور که در جدول ۳ مشاهده می‌شود، با افزایش تعداد گره‌های تشخیص‌دهنده، زمان کاهش می‌یابد. به طوری که زمان کل هنگام داشتن ۳ گره نسبت به زمانی که یک گره تشخیص‌دهنده داریم، یک سوم شده است. لذا با داشتن ۹ گره و یا ۱۲ گره تشخیص‌دهنده و برای این حجم از بردارهای حمله، زمان تشخیص و ذخیره‌سازی در هایپرلجرفابریک به ترتیب به حدود ۷۲ ثانیه برای ۹ گره و حدود ۵۰ تا ۶۰ ثانیه برای ۱۲ گره تقلیل می‌یابد.

[1] Propagation

[2] DT-HLF

```
# Batch Timeout: The amount of time to wait before creating a batch
BatchTimeout: 2s

# Batch Size: Controls the number of messages batched into a block
BatchSize:

    # Max Message Count: The maximum number of messages to permit in a batch
    MaxMessageCount: 10

    # Absolute Max Bytes: The absolute maximum number of bytes allowed for
    # the serialized messages in a batch.
    AbsoluteMaxBytes: 99 MB

    # Preferred Max Bytes: The preferred maximum number of bytes allowed for
    # the serialized messages in a batch. A message larger than the preferred
    # max bytes will result in a batch larger than preferred max bytes.
    PreferredMaxBytes: 512 KB
```

**شکل ۱۲. نحوه‌ی ساخت بلوک‌های هایپرلجرفابریک بر أساس زمان و یا حجم تراکنش‌ها [۲۶ و ۲۷]**

**جدول ۲. نتایج ارزیابی در حالت داشتن دو برچسب شامل حمله یا ترافیک نرمال**

| HN | Num-of-Nodes | Accuracy | T.* Time | Prediction Time | Training Time | Model | Article |
|---|---|---|---|---|---|---|---|
| - | 1 | 75.3 | - | 0.08 S | 5.53 S | NB | K. Yogesh [132] 19 |
| | 1 | 88.6 | - | 0.06 S | 60.64 S | SVM | K. Yogesh 19 |
| - | 1 | 89.3 | -- | 0.10 S | 16.48 S | RF | K. Yogesh 19 |
| - | 1 | 93.3 | - | 0.11 S | 19.5 S | DT | K. Yogesh 19 |
| 60 | 1 | 99.41% | - | 89.1 S | 1901.2 S | BiLSTM | O. Alkadi [130] 20 |
| 60 | Saved in Ethereum 20-TPS | 99.41% | 2615 S | 89.1 S | - | ETH-BiLSTM | O. Alkadi ETH 20 |
| - | 1 | 90.20 RP | - | 0.096 S RP | 17.43 S | Single-DT | Proposed Single RP |
| - | 3 | 90.20 RP | - | 0.038 S RP | - | 3-DT | Proposed 3Nodes RP 22 |
| - | Saved in Hyper-Ledger-Fabric 2000-TPS | 90.20 RP | 26.15 S | 0.038 S | - | HLF-DT | Proposed 3Nodes-HLF RP-HLF 22 |

*Transaction

**جدول ۳. زمان تشخیص و ذخیره‌سازی بر أساس دو برچسب حمله/سالم**

| Data Base Type | Num-of-Nodes | Time To Detect and Store | Model | Article |
|---|---|---|---|---|
| Usual Local File | 1 | 51.36 Second | Single-DT | Proposed 1Node RP 22 |
| Usual Local File | 3 | 17.12 Second | 3-DT | Proposed 3Nodes RP 22 |
| Distributed Ledger HLF | 1 Detector Saved in Hyper-Ledger-Fabric | 650 Second | HLF-DT | Proposed 1Nodes-HLF RP-HLF 22 |
| Distributed Ledger HLF | 3 Detector Saved in Hyper-Ledger-Fabric | 216 Second | HLF-DT | Proposed 3Nodes-HLF RP-HLF 22 |

## ۴-۹- سربار حافظه ذخیره‌سازی لاگ‌ها

سربار حافظه را با بررسی مجموعه داده تست متشکل از ۸۲۰۰۰ ردیف ترافیک در اختیار و ۵۲۳۰۸ بردار حمله و با در نظر گرفتن سه معماری پیشنهادی در پژوهش، شامل معماری تک گره، معماری با سه گره و معماری با ۳ گره در بستر زنجیره‌بلوکی هایپرلجرفابریک مطابق جدوال پیش‌رو بررسی شد. حافظه‌ی مصرفی هنگام ذخیره لاگ حملات کشف شده حاصل از تحلیل ترافیک با استفاده از مجموعه داده تست متشکل از ۸۲۰۰۰ ردیف ترافیک و تعداد ۵۲۳۰۸ بردار حمله و بر اساس دو برچسب ترافیک نرمال و حمله مطابق جدول ۴ می‌باشد.

**جدول ۴. حافظه مصرفی ذخیره‌سازی بر أساس دو برچسب حمله/سالم**

| Data Base Type | No. of-Nodes | Disk-Overhead | Model | Article |
|---|---|---|---|---|
| Usual Local File | 1 | 4 MB | Single-DT | Proposed 1Node RP 22 |
| Usual Local File | 3 | 1.5 MB | 3-DT | Proposed 3Nodes RP 22 |
| Distributed Ledger HLF | 1 Node Saved in Hyper-Ledger-Fabric | 304 MB | HLF-DT | Proposed 1Nodes-HLF RP-HLF 22 |
| Distributed Ledger HLF | 3 Node Saved in Hyper-Ledger-Fabric | 304 MB | HLF-DT | Proposed 3Nodes-HLF RP-HLF 22 |

با افزایش گره‌های تشخیص‌دهنده حافظه مصرفی تقریباً ثابت می‌ماند. حافظه و سی‌پی‌یو استفاده شده توسط گره‌های تشخیص دهنده در معماری پیشنهادی و در حالت ماکزیمم خود می‌باشد.

## ۴-۱۰- جداول ارزیابی زمان و حافظه مصرفی برای تشخیص و ذخیره‌سازی یک و بیست حمله

جدول ۵ سربار حافظه و همچنین سربار زمانی برای تشخیص یک، ده و بیست حمله در حالت داشتن دو برچسب شامل ترافیک سالم و حمله را نمایش می‌دهد.

**جدول ۵. سربار حافظه و همچنین سربار زمانی برای تشخیص یک، ده و بیست حمله در حالت داشتن دو برچسب**

| Data Base Type | No. of Attacks | Disk-Overhead | Time-Overhead | Model | Node |
|---|---|---|---|---|---|
| Usual Local File<br>Distributed Ledger HLF | 1 | 3 kB<br>5.1 KB | 0.02 s<br>2.03 s | Single-DT<br>HLF-DT | RP<br>RP-HLF |
| Usual Local File<br>Distributed Ledger HLF | 10 | 4 kB<br>52 kB | 0.03 s<br>0.31 s | Single-DT<br>HLF-DT | RP<br>RP-HLF |
| Usual Local File<br>Distributed Ledger HLF | 20 | 8 kB<br>108 kB | 0.04 s<br>0.51 s | Single-DT<br>HLF-DT | RP<br>RP-HLF |

## ۴-۱۱- اثبات معماری (تهدیدات احتمالی نسبت به مدل پیشنهادی و راه‌های دفاع)

در شکل ۱۳ و مطابق بردار ۱ حمله، یکی از چالش‌های زنجیره‌بلوکی هایپرلجرفابریک بدست آوردن سطح دسترسی مدیر کنترل کننده (ایجاد کننده) هایپرلجرفابریک می‌باشد. از جمله اقدامات امنیتی برای این کار می‌توان به محافظت فیزیکی، به‌کارگیری نام کاربری و گذرواژه بسیار قدرتمند اشاره کرد. همچنین خود هایپرلجرفابریک به استفاده از روش‌های رمزنگاری برای رمزنگاری این فایل‌ها و همچنین استفاده از ماژول امنیتی سخت‌افزاری برای محافظت بیشتر از این فایل‌ها توصیه می‌کند. همچنین با عملیات رمزنگاری و ... این کار غیرممکن‌تر خواهد شد.

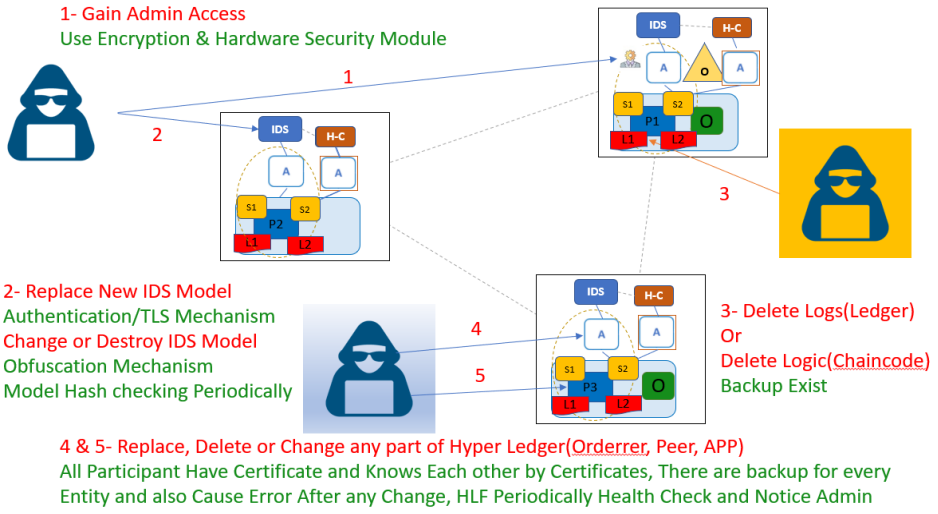
بردار حمله ۲ که به حمله به سیستم (مدل) تشخیص‌دهنده اشاره دارد، به دلیل عدم امکان تغییر یا به روز رسانی یا تخریب مدل تشخیص‌دهنده بدون هماهنگی و اجماع سایر گره‌ها، از حملات متخاصمانه[1] و داخلی که از روی عمد با تغییر یا تضعیف سیستم تشخیص نفوذ موجب راه یافتن حملات و ترافیک مخرب و عدم شناسایی این ترافیک مخرب به زیرساخت و سایر گره‌ها می‌شود، جلوگیری می‌شود.

بردار حمله ۳ که حذف کدزنجیره‌ای و یا اطلاعات و لاگ‌های کشف و گزارش شده و ذخیره شده در دفترکل توزیع شده را هدف قرار داده است، به دلیل ثبت و به اشتراک گذاری کدزنجیره‌ای روی همتاها و همچنین به اشتراک گذاری لاگ‌ها و اطلاعات حملات به صورت توزیع شده روی دفترکل توزیع شده موجود روی همتاها، امکان تغییر یا پاک کردن لاگ‌ها روی تک گره اثر گذار نخواهد بود، زیرا این اطلاعات در چندین سیستم و به صورت غیر متمرکز ذخیره شدند و لذا با این سیاست در نظر گرفته شده، برخی حملات از جمله حملات بعد از سوءاستفاده که شامل پاک کردن لاگ‌ها و ردپاهای پس از عملیات مخرب، توسط حمله کننده می‌باشد، غیرممکن یا سخت می‌شود، این امر همچنین به پایدار بودن اطلاعات و لاگ‌های شناسایی شده برای استفاده در عملیات جرم‌یابی دیجیتال کمک می‌کند.

بردارهای حمله ۴ و ۵ همانطور که مطابق شکل نمایش داده شده است تخریب و از بین بردن و یا جانشانی (جعل) اجزای اصلی هایپرلجرفابریک شامل همتا، برنامه کاربردی، سرویس سفارش‌دهنده (ترتیب‌دهنده) را هدف قرار داده است.. در مورد از بین بردن هریک از اجزا روی یک میزبان باید عنوان کرد که برای هر کدام از گره‌های سفارش‌دهنده و همتا اطلاعات پشتیبان در سایر میزبان‌ها وجود داشته و در صورت از بین رفتن هریک از آن‌ها می‌توان از اطلاعات جایگزین استفاده کرد. همچنین طبق گفته مستندات هایپرلجرفابریک، سلامت اجزای تعریف شده در هایپرلجر فابریک به صورت دوره‌ای و با پیام‌های ارتباطی مثل پیام سلام و ... برای بررسی زنده و فعال بودن اجزا که در دیگر ساختارها مانند سیسکو نیز وجود دارند بررسی می‌شوند، در صورت وجود مشکل و عدم پاسخ از گره‌ها و یا عدم توانایی اتصال، می‌تواند تحت عنوان

[1] Adverserial

پیغام خطا به مدیر سیستم گزارش شود.




**شکل ۱۳. بردارهای حمله احتمالی برای مدل ارائه شده و روش‌های پیش‌گیری از این حملات**

## ۵- نتیجه‌گیری

نتایج نشان داد مزایای راه‌حل پیشنهادی شامل چند مورد است. به دلیل استفاده از چندین گره تشخیص‌دهنده به جای تک گره، گره‌ها دید جامع‌تر و کامل‌تری (بدون هدر رفت یا گم شدن بسته‌های ترافیکی) از کل شبکه دارند، همچنین به دلیل توزیع شدن بار مانیتورینگ و تشخیص، سربار کمتری به تگ گره وارد می‌شود. حذف نقطه‌ی شکست واحد به دلیل توزیع شدن بار ترافیکی و پردازش‌های تحلیل و تشخیص روی چندین گره. حفظ اعتماد به دلیل تعریف شدن و اعطای مجوز به گره‌ها با استفاده از زنجیره‌بلوکی دارای مجوز هایپرلجرفابریک و بهره‌مندی از مکانیسم‌های آن از جمله گواهی‌نامه دیجیتال، ارتباط امن توسط پروتکل امنیت لایه ترابرد و ...

حفظ جامعیت لاگ‌های ذخیره شده برای استفاده در عملیات جرم‌یابی، جلوگیری از حملات پس از سوء استفاده و جلوگیری از پاک شدن لاگ‌های شناسایی شده به دلیل وجود فایل‌های پشتیبان و غیرمترکز ذخیره شده روی همتاهای مختلف. سربار زمانی کم تشخیص، ذخیره و انتشار لاگ‌ها و اطلاعات شناسایی شده در قالب تراکنش‌ها به دلیل سرعت بالای هایپرلجرفابریک نسبت به سایر بسترهای زنجیره‌بلوکی (افزایش سرعت تشخیص، انتشار و ذخیره‌سازی با افزایش تعداد گره‌ها). محافظت و جلوگیری از تغییر منطق (کدزنجیره‌ای) به دلیل داشتن کدزنجیره‌ای پشتیبان در چندین همتا.

جلوگیری از جایگزینی، جعل یا تغییر مدل تشخیص‌دهنده به علت استفاده از مکانیزم‌های احراز هویت و درهم سازی کد. اطمینان از یکسان بودن مدل روی تمام گره‌ها و عدم تخریب یا تزریق کد در مدل به دلیل بررسی‌های دوره‌ای و مقایسه مقدار چکیده مدل موجود و فعال روی هر میزبان با مدل مورد تائید و ذخیره شده (موجود از قبل) در دفترکل توزیع شده. به روز رسانی مدل تشخیص‌دهنده با داشتن مجموعه داده جامع و فراگیر و قابل اعتماد و پایدار با توجه به ترافیک واقعی گذری از کل زیرساخت و اطلاعات و حملات کشف شده از این ترافیک. قابلیت گسترش پذیری بالا و ساده به علت استفاده از چهارچوب هایپرلجرفابریک، مقابله با مشکلات حافظه به دلیل وجود اختیارات مدیران، امکان حذف لاگ‌ها و داده‌های بدون کاربرد و بسیار قدیمی توسط مدیران. جلوگیری از تغییرات ناخواسته در زیرساخت هایپرلجرفابریک بوسیله استفاده از مکانیزم‌های رمزنگاری و ماژول امنیتی سخت‌افزاری.

همچنین نقاط ضعف روش پیشنهادی شامل چند مورد است. اگر هنگام یادگیری و ساخت مدل، یادگیری مخربی صورت بگیرد، مدل و در نتیجه نتایج حاصل از آن فاقد اعتبار هستند. فرآیند به روز رسانی مدل تشخیص‌دهنده، فرآیندی است که به صورت غیر خودکار انجام می‌شود (عامل انسانی). اگر بیش از $N/2 +1$ گره‌ها از دسترس خارج شده و یا از بین بروند، عملیات اجماع با مشکل مواجه می‌شود. به دلیل استفاده از زنجیره‌ی‌بلوکی و روش تشخیص نفوذ مبتنی بر یادگیری ماشین و ...، همچنان کمی سربار پردازشی، زمانی و حافظه بالا می‌باشد.

## مراجع


[1] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for internet of Things (IoT) ", *Journal of ISMAC*, vol. 2, no. 04, pp. 190-199, 2020.

[2] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Implementing Lightweight IoT-IDS on Raspberry Pi Using Correlation-Based Feature Selection and Its Performance Evaluation", *Springer International Conference on Advanced Information Networking and Application, AINA: Advanced Information Networking and Applications*, Vol. 926, pp. 458-469, 2019.

**[3]** O. Alkadi, N. Moustafa, B. Turnbull, and K. R. Choo, "Deep Blockchain Framework-enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", *IEEE Internet of Things Journal*, pp. 1-12, 2020.

[4] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Poster Abstract: Towards Scalable and Trustworthy Decentralized Collaborative Intrusion Detection System for IoT," *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 256-257, 2020.

[5] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When Intrusion Detection Meets Blockchain Technology: A Review," *IEEE Access*, vol. 6, pp. 10179-10188, 2018.

[6] S. Yakut, Ö. Şeker, E. Batur, and G. Dalkılıç, "Blockchain Platform for Internet of Things," *2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey*, pp. 1-6, 2019.

[7] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards Secure and Decentralized Sharing of IoT Data", *2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA*, pp. 176-183, 2019.

[8] A. Sforzin, F. G. Mármol, M. Conti, and J. -M. Bohli (2016), "RPiDS: Raspberry Pi IDS — A Fruitful Intrusion Detection System for IoT," *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 2016, 2016.

[9] A. M. Chandrasekhar, and K. Raghuveer, "Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers", *2013 International Conference on Computer Communication and Informatics IEEE*, pp. 1-7, 2013.

[10] S. Iqbal, M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K. K. R. Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service.," *Journal of Network and Computer Applications*, pp. 98-120, 2016.

[11] P. Mehetrey, B. Shahriari, and M. Moh, "Collaborative ensemble-learning based intrusion detection systems for clouds.," *2016 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 404-411, 2016.

[12] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," *IEEE Symposium on Computers and Communication (ISCC), Larnaca*, pp. 180-187, 2015.

[13] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *10th International Conference for Internet Technology and Secured Transactions (ICITST) IEEE, London*, pp. 336-341, 2015.

[14] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam*, pp. 492-496, 2017.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system,"2009. http://bitcoin.org/bitcoin.pdf.

[16] M., Crosby, N. P. Pattanayak, S. Verma, and V. Kalyanaraman, "BlockChain Technology: Beyond Bitcoin,," *Applied Innovation Review*, 2016.

[17] S. Zamani, Z. Moezkarimi, and Z. Golmirzaei (2019), "Classifying, Comparing, and Analyzing Blockchain Platforms," *International Conference on Web Research, Tehran, Iran*.

[18] T. W. Shinder, "The Best Damn Firewall Book Period," Elsevier, 2011.

[19] "research.unsw.edu.au," [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

[20] K. Yogesh, M. Karthik, T. Naveen, and S. Saravanan, "Design and Evaluation of Scalable Intrusion Detection System Using Machine Learning and Apache Spark," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), pp. 1-7, 2019.

[21] "https://hyperledger-fabric.readthedocs.io/en/release-2.2/peers/peers.html," 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/peers/peers.html

[22] "hyperledger-fabric," 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/membership/membership.html.

[23] "hyperledger-fabric," hyperledger-fabric, 2018. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/ledger/ledger.html.

[24] "pycryptodome," [Online]. Available: https://pycryptodome.readthedocs.io/en/latest/src/hash/sha3_256.html.

[25] "https://hyperledger-fabric.readthedocs.io," 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/deployment_guide_overview.html.

[26] C. A. Ronao, and S. B. Cho, "Mining SQL queries to detect anomalous database access using random forest and PCA", In International conference on industrial, engineering and other applications of applied intelligent systems (pp. 151-160). Springer, Cham., 2015.

[27] "https://hyperledger-fabric.readthedocs.io/en/release-2.2," 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/create_channel/create_channel_config.html?highlight=batchtimeout#orderer