Karafan

Quarterly Scientific Journal of Technical and Vocational University

Autumn 2023, Vol. 20, No. 3, p. 553-570

Journal homepage: https://karafan.tvu.ac.ir/?lang=en

TVU

doi:10.48301/KSSA.2024.423558.2754

Original Research

E-ISSN: 2538-4430
ISSN: 2382-9796

Reducing the False Alarm Rates in Detecting Botnets Using the Combination of K-Nearest Neighbors and Stochastic Gradient Descent Algorithms

Aliakbar Tajari Siahmarzkooh[1*]

[1]Assistant Professor, Department of Computer Sciences, Golestan University, Gorgan, Iran.

ARTICLE INFO

Received: 11.05.2023
Revised: 12.24.2023
Accepted: 01.21.2024

Keyword:
Botnet Detection
False Alarm Rate
K-Nearest Neighbors Algorithm
Stochastic Gradient Descent

***Corresponding Author:**
Aliakbar Tajari Siahmarzkooh
Email: *a.tajari@gu.ac.ir*

ABSTRACT

With the increasing expansion of networks connected to the internet, attackers' efforts against these networks have also grown. Therefore, many researchers have proposed solutions to deal with botnets that lead to remote contamination of systems. One of the main problems of existing methods is the high rate of false alarms produced by attack detection systems, including the rate of false positives and false negatives. In the present research, by using machine learning algorithms, these alarm rates were reduced. In the first stage of the proposed solution, the dataset entered a pre-processing stage so that outliers and noise data were identified and discarded. Then, using the K-Nearest Neighbor algorithm, the non-useful features that had no effect in determining the data class were excluded from the dataset. In the next step, the Gradient Descent algorithm was used to accurately detect the class of data and categorize them into normal data or botnet attack. Finally, by performing various tests on the CTU-13 and BoT-IoT datasets in both binary and multi-class modes, the values of the important criteria for evaluating the effectiveness of the botnet attack detection system were obtained. The results showed that in the CTU-13 dataset, in binary and multi-class mode, the false negative rates were 0.01 and 0.04, and the false positive rates were 0.01 and 0.05, respectively; and for the BoT-IoT dataset, in binary and multi-class mode, the false negative rates were 0.02 and 0.05 and the false positive rates were 0.03 and 0.05, respectively. Compared to other existing methods, the proposed method is superior and demonstrates a reduction in the rate of false alarms and improves efficiency.

©2023 Technical and Vocational University, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC 4.0 license) (https://creativecommons.org/licenses/by-nc/4.0/).

EXTENDED ABSTRACT

Introduction

Current networks are constantly exposed to attacks by hackers. These attacks are relatively simple to implement despite their complexity, especially since security penetration testing tools are widely available to both user computer technology and hackers. When the complexity of attacks increases, it is necessary to make intrusion detection solutions more up-to-date and reliable. Therefore, intrusion detection is very necessary and important to ensure the security and integrity of information systems. Analyzing information collected by security solutions, intrusion detection systems use various rules to distinguish between normal and dangerous events.

Soft computing tools and approaches are relatively newer methods that are used in the field of intrusion detection. Machine learning, artificial intelligence, and deep learning algorithms such as neural networks, regression, decision trees, convolutional neural networks, etc., are common soft computing tools and are used today in various fields to provide more efficient solutions to problems with lower costs. In this article, we use the classification system based on machine learning to better display the proposed model.

Methodology

The work steps included three parts: data preprocessing, data reduction, and data classification.

Before the data enters the classification phase, the noise and outlier data are first removed. Normalization of the data is done by changing the range of values to the range of zero to one.

Due to the large amount of data for classification, it is necessary to exclude those features that don't have an effective role in determining the class and category of data from the dataset, and only the features with a higher impact are used for further processing. K-nearest neighbor classification (KNN) is a popular supervised learning algorithm used for classification tasks in machine learning. This algorithm works by placing the k-nearest neighbor to a new data point and classifying it based on the majority of the classes of its neighbors. The mentioned algorithm is simple and effective, but it has problems when dealing with large dimensions of the dataset, which can be computationally expensive if the dataset is voluminous. This algorithm is often used as a basic model to compare with more complex algorithms.

The stochastic gradient descent (SGD) classifier is a linear classification algorithm widely used in machine learning applications. This algorithm based on SGD optimization is a widely used optimization solution for training large-scale machine learning models. This algorithm operates by repeatedly adjusting the weights of a linear model along the negative gradient of the loss function. SGD classification is known for its efficiency, scalability, and ability to handle large datasets, making it a popular choice in many real-world applications. In this article, we use this algorithm to classify data into two categories: normal and attack.

Results and discussion

Table 1 shows the values of the evaluation parameters for different percentages of training data samples and different values of k in the k-fold validation mode. The best result for the parameters in the validation mode was obtained when the k-fold validation method was used with the value of k=8, and in the percentage mode when 63% of data samples were used for training and 37% of data were used for testing.

Table 2, like the previous table, shows the values of evaluation parameters for multi-class classification. As can be observed in this table, the best results for the parameters were obtained when using the k-fold validation method with the value of k=7. In general, the results obtained from binary classification were better than multi-class classification.

Table 1. Binary classification results with k-fold cross-validation and percentage split.

Parameter	5-fold Cross validation	8-fold Cross validation	10-fold Cross validation	Train (63%)	Train (74%)	Train (76%)
FPR	0.15	**0.02**	0.07	**0.01**	0.09	0.08
FNR	0.12	**0.01**	0.09	**0.01**	0.07	0.10
Accuracy	94.87	**99.97**	97.38	**99.99**	98.35	96.58
F1 Score	93.05	**99.76**	96.44	**99.81**	97.55	95.81
Error rate	5.13	**0.03**	2.62	**0.01**	1.65	3.42

Table 2. Multi-class classification results with k-fold cross validation and percentage split.

Parameter	5-fold Cross validation	8-fold Cross validation	10-fold Cross validation	Train (63%)	Train (74%)	Train (76%)
FPR	0.13	**0.05**	0.07	0.07	**0.04**	0.08
FNR	0.11	**0.05**	0.08	0.06	**0.05**	0.08
Accuracy	96.27	**99.03**	97.20	98.33	**99.34**	96.28
F1 Score	95.72	**98.76**	97.38	97.41	**99.28**	96.21
Error rate	3.73	**0.97**	2.80	1.67	**0.66**	3.72

Tables 3 and 4 show the comparison between the value of the best result obtained from the most important evaluation criteria in the proposed method with some other methods for binary and multi-class classification modes, respectively. By comparison, it can be concluded that the use of KNN and SGD methods can lead to improvement in the accuracy of intrusion detection and particularly reduction in the rate of false alarms in the detection of botnets in the dataset.

Table 3. Binary classification comparison.

Method	FNR	FPR	Accuracy
DT+KNN	0.06	0.08	99.26
RL+AdaBoost	0.08	0.08	99.39
SVM+NB	0.07	0.06	99.54
CNN+LSTM	0.05	0.07	99.66
RF+ANN	0.05	0.05	99.85
SGD+KNN (Proposed)	**0.01**	**0.01**	**99.99**

Table 4. Multi-class classification comparison.

Method	FNR	FPR	Accuracy
DT+KNN	0.09	0.08	98.38

Method	FNR	FPR	Accuracy
RL+AdaBoost	0.09	0.09	99.12
SVM+NB	0.08	0.07	98.52
CNN+LSTM	0.09	0.09	98.74
RF+ANN	0.08	0.11	99.21
SGD+KNN (Proposed)	**0.04**	**0.05**	**99.34**

Conclusion

With the expansion of the use of the web space by users, the attacks on devices connected to the internet have also increased. Thus far, many researchers have provided solutions to deal with botnets that lead to remote system contamination. However, since their proposed methods suffer from high false alarm rates in detecting botnets, in this article, a combination of two machine learning algorithms called k-nearest neighborhood and stochastic gradient descent was used. One was used to reduce the features and the other to better categorize the data. The results obtained from the experiments and their comparison with some other methods showed that the combination of these two algorithms can improve the accuracy of diagnosis and particularly reduce the rate of false positives and false negatives in detect botnets.

کارافن
فصلنامه علمی دانشگاه فنی و حرفه‌ای
پاییز ۱۴۰۲، دوره ۲۰، شماره ۳، ۵۷۰-۵۵۳
آدرس نشریه: https://karafan.tvu.ac.ir/
doi:10.48301/KSSA.2024.423558.2754

مقاله پژوهشی
شاپای الکترونیکی: ۲۵۳۸-۴۴۳۰
شاپای چاپی: ۲۳۸۲-۹۷۹۶

کاهش نرخ هشدارهای نادرست در تشخیص بات‌نت‌ها با ترکیب الگوریتم‌های k- نزدیکترین همسایگی و گرادیان کاهش تصادفی

علی اکبر تجری سیاه مرزکوه[1*]

۱- استادیار، گروه علوم کامپیوتر، دانشگاه گلستان، گرگان، ایران

اطلاعات مقاله

دریافت مقاله: ۱۴۰۲/۰۸/۱۴
بازنگری مقاله: ۱۴۰۲/۱۰/۰۳
پذیرش مقاله: ۱۴۰۲/۱۱/۰۱

کلید واژگان:
تشخیص بات‌نت
نرخ هشدار نادرست
الگوریتم k- نزدیکترین همسایگی
گرادیان کاهش تصادفی

***نویسنده مسئول:** علی اکبر تجری سیاه مرزکوه
پست الکترونیکی:
a.tajari@gu.ac.ir

چکیده

با گسترش روزافزون شبکه‌های متصل به اینترنت، حملات مهاجمان به این شبکه‌ها نیز رشد کرده است. بنابراین، محققان زیادی برای مقابله با بات‌نت‌ها که از راه دور منجر به آلودگی سیستم‌ها می‌شوند راهکارهایی را ارائه کرده‌اند. یکی از معضلات اصلی روش‌های موجود، نرخ بالای هشدارهای نادرست تولید شده توسط سیستم‌های تشخیص حمله از جمله نرخ مثبت کاذب و منفی کاذب است. در این مقاله برای کاهش نرخ هشدارهای نادرست از ترکیب دو الگوریتم یادگیری ماشین استفاده می‌شود. در مرحله اول راهکار پیشنهادی، مجموعه داده وارد یک مرحله پیش‌پردازش می‌شود تا داده‌های پرت و نویز شناسایی شده و کنار گذاشته شوند. پس از آن با استفاده از الگوریتم k- نزدیکترین همسایگی، ویژگی‌های غیر مفید که در تعیین کلاس داده‌ها اثری ندارند از مجموعه داده کنار گذاشته می‌شوند. در مرحله بعدی، برای تشخیص دقیق کلاس داده‌ها و دسته‌بندی آنها به داده عادی یا حمله بات‌نت، از الگوریتم گرادیان کاهش تصادفی استفاده می‌گردد. در پایان، با انجام آزمایش‌های مختلف بر روی مجموعه داده‌های CTU-13 و BoT-IoT در هر دو حالت دودویی و چند کلاسه، مقادیر معیارهای مهم ارزیابی کارآیی سیستم تشخیص حملات بات‌نت به‌دست می‌آیند. نتایج نشان می‌دهد که در مجموعه داده CTU-13، در حالت دودویی و چند کلاسه به‌ترتیب نرخ منفی کاذب ۰.۰۱ و ۰.۰۴ و نرخ مثبت کاذب ۰.۰۱ و ۰.۰۵ و برای مجموعه داده BoT-IoT، در حالت دودویی و چند کلاسه به‌ترتیب نرخ منفی کاذب ۰.۰۲ و ۰.۰۵ و نرخ مثبت کاذب ۰.۰۳ و ۰.۰۵ به‌دست می‌آید که در مقایسه با سایر روش‌های موجود از برتری برخوردار است و نشان می‌دهد که روش پیشنهادی منجر به کاهش نرخ هشدارهای نادرست و در نتیجه بهبود کارآیی می‌شود.

©2023 Technical and Vocational University, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC 4.0 license) (https://creativecommons.org/licenses/by-nc/4.0/).

مقدمه

شبکه‌های کنونی به‌طور مداوم در معرض حملات مهاجمان قرار می‌گیرند. این حملات با وجود پیچیدگی برای پیاده‌سازی بسیار ساده هستند، به‌ویژه به این دلیل که ابزارهای آزمایش نفوذ امنیتی به‌طور گسترده برای هر دو فناوری رایانه کاربر و هکرها در دسترس هستند. زمانی‌که پیچیدگی حملات افزایش می‌یابد لازم است تا راهکارهای تشخیص نفوذ نیز به همان میزان به‌روزتر و قابل اعتمادتر ساخته شوند [۱]. از جمله حملاتی که به شبکه‌ها صورت می‌گیرد شامل موارد زیر می‌باشد: حملات سیل‌آسا که منجر به عدم سرویس‌دهی می‌شود، پویش پورت‌ها که نقاط آسیب‌پذیر سیستم را جستجو می‌کند، حدس کلمات عبور (تلاش برای ورود غیر مجاز به سیستم)، یا حملات سرریز بافر که می‌تواند شامل یک سوء استفاده برای دسترسی به منابع و ریشه باشد [۲]. بنابراین تشخیص نفوذ، امری بسیار ضروری و مهم برای تأمین امنیت و یکپارچگی سیستم‌های اطلاعاتی به‌شمار می‌آید. سیستم‌های تشخیص نفوذ با تجزیه و تحلیل اطلاعات جمع‌آوری شده توسط راهکارهای امنیتی، از قوانین مختلفی برای ایجاد تمایز بین رویدادهای عادی و خطرناک استفاده می‌کند.

دو دسته کلی برای سیستم‌های تشخیص نفوذ وجود دارد [۳]: (۱) تشخیص نفوذ مبتنی بر امضا یا سوء استفاده که بر روی مجموعه‌ای از حملات از قبل شناخته شده آموزش داده می‌شود و (۲) تشخیص نفوذ مبتنی بر ناهنجاری که مدلی را برای رفتار عادی ایجاد می‌کند و هرگونه دسترسی با ویژگی‌های متفاوت با حالت عادی را به‌عنوان حمله شناسایی می‌کند. نقطه ضعف اصلی روش‌های وابسته به امضا این است که روش داده شده فقط حملات شناخته شده را شناسایی می‌کند و نیاز به نگهداری مستمر در پایگاه داده دارد. مشکل اصلی روش تشخیص مبتنی بر رفتار نیز این است که در شرایطی منجر به تولید هشدارهای نادرست به‌دلیل خطای مدل می‌شود. در این مقاله نیز قصد داریم تا نرخ هشدارهای نادرست را کاهش دهیم.

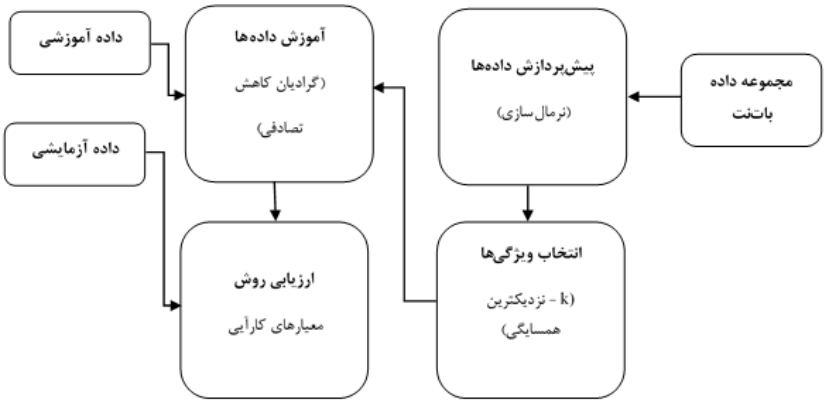
با داشتن مجموعه داده‌هایی که از شبکه گردآوری شده‌اند و با اعمال راهکارهای تشخیص نفوذ، عملاً داده‌ها به دو دسته داده‌های عادی و حمله طبقه‌بندی می‌شوند که بر اساس این طبقه‌بندی، اقداماتی توسط مدیران صورت می‌گیرد. سیستم‌های تشخیص نفوذ را می‌توان به روش دیگری نیز طبقه‌بندی کرد. رایج‌ترین طبقه‌بندی، این سیستم‌ها را به سیستم تشخیص نفوذ مبتنی بر میزبان و سیستم مبتنی بر شبکه طبقه‌بندی می‌کند [۳]. به‌طور کلی، یک سیستم تشخیص نفوذ مبتنی بر شبکه با استفاده از یک گره شبکه با یک کارت رابط شبکه و یک رابط برای مدیریت به‌صورت جداگانه پیکربندی می‌شود. این نوع سیستم تشخیص نفوذ اغلب در یک بخش یا در امتداد مرز شبکه قرار می‌گیرد تا با نظارت بر ترافیک شبکه، حملات را بررسی نماید. بر خلاف این نوع، سیستم مبتنی بر میزبان با استفاده از برنامه‌های کوچک نصب شده روی هر گره کار می‌کند که قابلیت نظارت بر سیستم عامل را دارد، داده‌ها را در فایل‌های گزارش وارد می‌کند و در صورت لزوم هشدار مورد نظر را ایجاد می‌کند. بنابراین، این نوع سیستم تشخیص نفوذ فاقد امکانات نظارت بر شبکه به‌عنوان یک موجودیت کلی است.

ابزارها و رویکردهای محاسبات نرم، روش‌های نسبتاً جدیدتری هستند که در زمینه تشخیص نفوذ به‌کار می‌روند. الگوریتم‌های یادگیری ماشین، هوش مصنوعی و یادگیری عمیق مانند شبکه‌های عصبی، رگرسیون، درخت تصمیم، شبکه عصبی کانالوشنال و مواردی از این دست، ابزارهای رایج محاسبات نرم هستند و امروزه در زمینه‌های مختلف برای ارائه راه‌حل‌های کارآمدتر برای مشکلات با هزینه‌های کمتر مورد استفاده قرار می‌گیرند [۴].

در واقع، تحقیقات جدید رویکردهای جدیدی را برای مساله تشخیص نفوذ اتخاذ می‌کنند. این تحقیقات به‌جای تلاش برای شناسایی یک نفوذ بر مبنای مقایسه امضای آن با یک پایگاه داده، اغلب بر ایجاد مدلی از رفتارهای عادی متمرکز شده‌اند که می‌تواند برای هر نوع رفتاری که متناقض با رفتار عادی است اعمال شود. برای ارائه چنین راهکارهایی نیاز است تا رفتار ترافیک عادی مورد بررسی دقیق قرار گیرد تا در نهایت مدلی ایجاد شود که بتواند ترافیک شبکه را به یکی از دو دسته عادی یا حمله طبقه‌بندی کند. تنها در این صورت است که می‌توان حملات و انواع آنها را با دقت بالا

شناسایی کرد. در این حوزه تحقیقاتی، روش‌های یادگیری ماشین، ابزار کمکی و ارزشمندی محسوب می‌شوند، زیرا قادر به مدل‌سازی سیستم‌های پیچیده و پویا هستند. این امر به دلیل ویژگی‌های ذاتی آنها برای بازنمایی رفتار داده‌ها است. ما نیز در این مقاله، از سیستم طبقه‌بندی مبتنی بر یادگیری ماشین برای نمایش بهتر مدل پیشنهادی استفاده می‌کنیم. فلوچارت کلی روش پیشنهادی در شکل ۱ نشان داده شده است.

نحوه تنظیم مطالب ارائه شده در مقاله به شرح زیر است: بخش ۲ به بررسی و شرح چند نمونه از کارهای مرتبط در حوزه تشخیص نفوذ می‌پردازد. در بخش ۳ به ارائه روش پیشنهادی به‌کار رفته در این مقاله می‌پردازیم. بخش ۴، سیستم پیشنهادی را مورد ارزیابی قرار می‌دهد و بخش ۵ نتیجه‌گیری کار پیشنهادی را شرح می‌دهد.



شکل ۱. فلوچارت روش پیشنهادی برای تشخیص باتنت.

کارهای مرتبط

در این بخش به توضیح چند نمونه کار تحقیقاتی انجام شده برای تشخیص باتنت می‌پردازیم که اغلب آنها از یادگیری ماشین در ساخت مدل استفاده می‌کنند.

در مقاله [۵]، یک رویکرد پرسپترون چند لایه مبتنی بر یادگیری ماشین برای تشخیص باتنت پیشنهاد شده است که به چالش شناسایی باتنت‌های غیر قابل مشاهده‌ای که می‌توانند از تجزیه و تحلیل مبتنی بر امضای سنتی فرار کنند می‌پردازد. این روش، متشکل از ماژول‌های پیش‌پردازش و طبقه‌بندی است که از الگوریتم‌های یادگیری ماشین برای شناسایی حمله باتنت استفاده می‌کند. نویسندگان برای شناسایی باتنت‌ها از تجزیه و تحلیل مبتنی بر رفتار با استفاده از ویژگی‌های مبتنی بر جریان استفاده کردند که سرآیند بسته را تحلیل می‌کند. چارچوب پیشنهادی به دقت ۹۲٪ و نرخ منفی کاذب ۱.۵٪ دست پیدا کرده است. نتایج این تحقیق، قدرت روش‌های یادگیری ماشین برای تشخیص باتنت را نشان می‌دهد و اهمیت تشخیص حمله باتنت مبتنی بر تحلیل رفتار را برجسته‌تر می‌کند.

در مقاله [۶]، با استفاده از روش‌های یادگیری عمیق، طبقه‌بندی بسته‌های داده ترافیک HTTP صورت گرفته است. هدف این تحقیق، بهبود دقت تشخیص و کاهش موارد مثبت کاذب است. روش پیشنهاد شده در این مقاله به دقت ۹۶.۳٪ در مجموعه داده آزمایش دست پیدا کرده است. در مقاله [۷]، یک روش یادگیری ماشین برای شناسایی حملات باتنت پیشنهاد شده است. نویسندگان، یک محیط صنعتی هوشمند بر اساس معماری سخت‌افزاری دستگاه اینترنت اشیا ساختند و از برنامه یادگیری ماشین وکا[1] با الگوریتم جنگل تصادفی برای دستیابی به دقت ۹۶٪ استفاده

[1] Weka

کردند. مدل پیشنهادی، امکان‌سنجی بالایی را در تشخیص حملات بات‌نت برای شبکه امنیتی کارخانه‌های هوشمند نشان داده است.

در مقاله [۸]، روشی پیشنهاد شد که هدف آن مقابله با چالش شناسایی بات‌نت‌های P2P[1] است که ویژگی‌های منحصر به‌فردی دارند که تشخیص آنها را دشوار می‌کند. طبقه‌بندی کننده‌های یادگیری ماشین برای تحلیل ویژگی‌های ترافیک شبکه و شناسایی بات‌نت‌های مرتبط با P2P استفاده می‌شوند. مجموعه داده‌های ISOT و CTU-13 برای آموزش و آزمایش مدل طبقه‌بندی کننده یادگیری ماشین استفاده شد. روش پیشنهادی از الگوریتم درخت تصمیم برای انتخاب ویژگی‌ها استفاده می‌کند و به دقت ۹۸.۷٪ دست یافته است.

در مقاله [۹]، تشخیص حمله بات‌نت با استفاده از روش ترکیبی CNN- LSTM[2] برای فرم‌های کاربردی اینترنت اشیا پیشنهاد شد. آزمایش‌های گسترده‌ای با استفاده از یک مجموعه داده واقعی به نام N-BaIoT، شامل الگوهای داده عادی و مخرب، استخراج شده از یک سیستم واقعی انجام شد. نویسندگان، الگوریتم جدیدی به نام یادگیری ترکیبی عمیق را پیشنهاد کردند که یک شبکه عصبی کانولوشنال و حافظه کوتاه‌مدت (CNN-LSTM) را برای شناسایی حملات بات‌نت ترکیب می‌کند. نتایج آزمایش، اثربخشی مدل پیشنهادی را در تشخیص حملات بات‌نت با نرخ دقت ۹۰.۸۸٪ و ۸۸.۶۱٪ نشان می‌دهد.

در مقاله [۱۰]، شناسایی حملات بات‌نت در دستگاه‌های اینترنت اشیا با استفاده از روش‌های یادگیری ماشین پیشنهاد شد. روش پیشنهادی شامل استفاده از درخت تصمیم، مدل XgBoost و مدل رگرسیون لجستیک است که بر روی مجموعه داده UNSW-NB15، آزمایش و ارزیابی شدند. روش درخت تصمیم در این مقاله با ۹۴٪ دقت، عملکرد بهتری داشت.

در مقاله [۱۱]، موضوع بات‌نت‌های اینترنت اشیا و تأثیر بالقوه آنها بر قابلیت اطمینان سیستم‌های اینترنت اشیا مورد تمرکز قرار گرفته است که با منابع محدود دستگاه‌های اینترنت اشیا تشدید می‌شود. هدف اصلی این مقاله، تمرکز بر به‌کارگیری مدل پیشرفته CTGAN است که یک رویکرد شبکه‌های متخاصم مولد در مدل‌سازی و تولید داده‌ها است. برای دستیابی به این هدف، محققان با یک مجموعه داده نامتعادل به نام Bot-IOT کار کردند و روش‌های عملی برای رسیدگی به این مشکل ابداع کردند. این یافته‌ها نتایج امیدوارکننده‌ای را نشان می‌دهد که پس از استفاده از CTGAN برای تقویت داده‌ها، پرسپترون چند لایه به دقت چشمگیر ۹۸.۹۳٪ در تشخیص موفقیت‌آمیز حملات بات‌نت اینترنت اشیا دست یافت. نتایج عملکرد برای تشخیص حمله چشمگیر است اما با این حال، پیشرفت‌های بیشتر نیز مورد نیاز است.

در مقاله [۱۲]، یک مدل جدید شناسایی حمله IoT- BOTNET هوشمند معرفی شد که از یک روش طبقه‌بندی ترکیبی بهینه استفاده می‌کند. این مقاله از مجموعه داده‌های بات‌نت اینترنت اشیا برای آموزش و آزمایش مدل استفاده می‌کند. یک مدل افزایش یافته اطلاعات برای شناسایی مؤثرترین ویژگی‌ها در داده‌ها استفاده می‌شود. مدل تشخیص یک طبقه‌بندی ترکیبی است که Bi-GRU بهینه‌سازی شده را با شبکه عصبی بازگشتی[3] (RNN) ادغام می‌کند. مدل ارائه شده به دقت تشخیص ۹۷٪ دست می‌یابد، اما هنوز یک خطای سه درصدی در دقت وجود دارد که باید برطرف شود.

[1] Peer-to-Peer
[2] Convolutional Neural Network - Long Short-Term Memory
[3] Recurrent Neural Network

راهکار پیشنهادی

در این بخش، توضیحات لازم در مورد روش تحقیق و مراحل انجام کار ارائه می‌کنیم. علاوه بر این، مجموعه داده‌ای را که برای ارزیابی روش پیشنهادی استفاده می‌شود را معرفی می‌کنیم. مراحل انجام کار شامل سه بخش پیش‌پردازش اولیه داده‌ها، کاهش داده‌ها و در نهایت دسته‌بندی داده‌ها است که در ادامه آنها را شرح می‌دهیم.

پیش‌پردازش داده‌ها

قبل از اینکه داده‌ها وارد فاز دسته‌بندی شوند ابتدا اقدام به حذف نویزها و داده‌های پرت می‌کنیم. نرمال‌سازی روی داده‌ها با تغییر بازه مقادیر به بازه صفر تا یک صورت می‌گیرد که از رابطه ۱ به‌دست می‌آید. در این رابطه، از روی مقادیر بازه قبلی، مقدار جدید که عددی بین صفر و یک است به‌دست می‌آید. همچنین با اعمال نمونه‌برداری جهت‌گیری شده (stratified sampling) [۱۳]، متناسب با هر کلاس اقدام به انتخاب تصادفی نمونه داده‌ها با نرخ ۵۰٪ می‌کنیم. در این رابطه، x_{min}، x_{max} و x_i به‌ترتیب نشان دهنده کمترین مقدار در بازه قبلی، بیشترین مقدار در بازه قبلی و مقدار جدید پارامتر می‌باشند.

$$x_i = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (۱)$$

دسته‌بندی k- نزدیکترین همسایگی برای کاهش داده‌ها

با توجه به حجم بالای داده‌ها برای دسته‌بندی نیاز است تا آن دسته از ویژگی‌هایی که نقش مؤثری در تعیین کلاس و دسته داده ندارند از مجموعه داده کنار گذاشته شده و تنها ویژگی‌ها با تأثیر بالاتر برای پردازش‌های بعدی مورد استفاده قرار گیرند. دسته‌بندی k- نزدیکترین همسایگی[1] (KNN) [۱۴]، یک الگوریتم یادگیری نظارت شده پرطرفدار است که برای کارهای طبقه‌بندی در یادگیری ماشین استفاده می‌شود. این الگوریتم با قرار دادن k- نزدیکترین همسایه به یک نقطه داده جدید و دسته‌بندی آن بر اساس اکثر کلاس‌های همسایه‌های آن عمل می‌کند. الگوریتم مذکور، یک الگوریتم ساده و با کارکرد مؤثر است، اما در برخورد با داده‌های با ابعاد زیاد مجموعه داده دچار مشکلاتی می‌شود که در صورت حجیم بودن مجموعه داده می‌تواند از نظر محاسباتی نیز گران تمام شود. این الگوریتم اغلب به‌عنوان یک مدل پایه برای مقایسه با الگوریتم‌های پیچیده‌تر استفاده می‌شود.

KNN یک روش ناپارامتریک است که با مکان‌یابی k- نزدیک‌ترین همسایگان یک نمونه داده جدید و پیش‌بینی متغیر خروجی آن بر اساس متغیر خروجی که توسط اکثر همسایگان k مشترک است عمل می‌کند. KNN یک روش آسان قابل درک است که ممکن است برای برنامه‌های رگرسیون و طبقه‌بندی اعمال شود. عملکرد روش روی داده‌های اعتبارسنجی ممکن است با تنظیم فراپارامترهای KNN، مانند تعداد همسایگان k و اندازه‌گیری فاصله، بهینه شود. با این حال، از آنجایی که KNN باید فواصل بین هر جفت نمونه را محاسبه کند می‌تواند برای مجموعه داده‌های عظیم و فضاهای ورودی با ابعاد بالا از نظر محاسباتی پرهزینه باشد.

در زیر نحوه بیان الگوریتم k- نزدیکترین همسایگی به‌صورت ریاضی آمده است [۱۵]:

فرض کنید $X = x_1, x_2, \dots, x_n$ مجموعه داده آموزشی باشد که در آن هر نمونه x_i با یک متغیر خروجی $y_i \in 1,2, \dots, k$ و یک مجموعه از متغیرهای ورودی $x_{i1}, x_{i2}, \dots, x_{ip}$ مرتبط است. هدف KNN پیش‌بینی متغیر خروجی y برای یک نمونه جدید x بر اساس k- نزدیک‌ترین همسایه‌اش است.

[1] K- Nearest Neighbors

فاصله بین دو نمونه x_i و y_i را می‌توان با استفاده از معیارهای مختلفی مانند فاصله اقلیدسی، فاصله منهتن و فاصله مینکوفسکی [۱۶] اندازه‌گیری کرد. k- نزدیکترین همسایه‌های x را می‌توان با مرتب کردن نمونه‌های آموزشی به‌ترتیب افزایش فاصله آنها به x و انتخاب k نمونه با کوچکترین فاصله شناسایی کرد.

زمانی‌که k همسایه شناسایی شدند، یک قانون اکثریت رأی می‌تواند متغیر خروجی پیش‌بینی شده را برای x محاسبه کند. به‌طور خاص، متغیر خروجی پیش‌بینی شده برای x، برچسب کلاس k است که بیشتر در میان همسایگان k ظاهر می‌شود که در رابطه ۲ نیز مشاهده می‌شود:

$$\hat{y} = \arg \max_{k \epsilon 1,2,\dots,K} \sum_{i \in N_k(x)} [y_i = k] \quad (۲)$$

که در این رابطه $N_k(x)$، مجموعه شاخص‌های k- نزدیکترین شاخص‌های همسایگی x و $[y_i = k]$ یک تابع نشانگر است که اگر $y_i = k$ باشد مقدار ۱ و در غیر این صورت مقدار صفر می‌گیرد. اگر بین متداول‌ترین متغیرهای خروجی یک نوع برابری وجود داشته باشد، یک مقدار تصادفی انتخاب می‌شود. با استفاده از این الگوریتم، ویژگی‌های کم‌اقرتر در تعیین کلاس کنار گذاشته می‌شوند.

گرادیان کاهش تصادفی برای دسته‌بندی داده‌ها

دسته‌بندی کننده گرادیان کاهش تصادفی[1] (SGD) [۱۷]، یک الگوریتم طبقه‌بندی خطی است که به‌طور گسترده در برنامه‌های یادگیری ماشین استفاده می‌شود. این الگوریتم بر اساس بهینه‌سازی SGD، یک راهکار بهینه-سازی پرکاربرد برای آموزش مدل‌های یادگیری ماشین در مقیاس بزرگ است. این الگوریتم با تنظیم مکرر وزن‌های یک مدل خطی در طول گرادیان منفی تابع زیان کار می‌کند. طبقه‌بندیSGD به‌دلیل کارایی، مقیاس‌پذیری و توانایی مدیریت مجموعه داده‌های بزرگ شناخته شده است که آن را به یک انتخاب محبوب در بسیاری از برنامه‌های کاربردی دنیای واقعی تبدیل می‌کند.

نمایش ریاضی دسته‌بندی کننده SGD را می‌توان به‌صورت زیر بیان کرد [۱۸]:

فرض کنید $X = x_1, x_2, \dots, x_n$ مجموعه داده آموزشی باشد که در آن هر نمونه x_i به یک متغیر خروجی دودویی $y_i \epsilon -1,1$ و یک مجموعه متغیر ورودی $x_{i1}, x_{i2}, \dots, x_{ip}$ انتساب داده شده باشد. فرض کنید $w = w_1, w_2, \dots, w_p$ مجموعه وزن‌هایی باشد که قرار است تخمین زده شوند. الگوریتم طبقه‌بندی گرادیان کاهش تصادفی به‌صورت زیر توصیف می‌شود:

۱- وزن‌ها را با مقادیر تصادفی کوچک مقداردهی اولیه کن، یا به‌عبارتی $w \sim N(0, \sigma^2)$.

۲- برای هر تکرار t، به‌طور تصادفی یک زیرمجموعه از مجموعه داده آموزشی $S_t \subseteq X$ را انتخاب کن.

۳- مجموع وزن‌های متغیرهای ورودی و ضرایب را برای نمونه‌ها در S_t محاسبه کن یا به‌عبارتی $z_t = \sum_{i \epsilon S_t} w_i x_i$.

۴- گرادیان تابع زیان را با توجه به وزن‌ها با استفاده از نمونه‌ها در S_t محاسبه کن یا به‌عبارتی $\nabla_m \mathcal{L}(w) = \frac{1}{|S_t|} \sum_{i \epsilon S_t} y_i x_i \left(1 - \sigma\left(y_i z_i\right)\right) + 2\alpha w$.

۵- وزن‌ها را با استفاده از قانون گرادیان کاهشی به‌روزرسانی کن یا به‌عبارتی $w_{t+1} = w_t - \eta \nabla_m \mathcal{L}(w)$ به‌گونه‌ای که η نرخ یادگیری باشد.

۶- مراحل ۲ الی ۵ را تا رسیدن به همگرایی یا حداکثر تکرار، تکرار کن.

[1] Stochastic Gradient Descent

۷- از وزن‌های تخمینی برای انجام پیش‌بینی نمونه‌های جدید استفاده کن یا به‌عبارتی $\hat{y} = sign(\sum_{j=1}^{p} \hat{w}_j x_j)$.

مجموع وزنی متغیرها و ضرایب ورودی با استفاده از تابع سیگموئید که برای وظایف دسته‌بندی دودویی مناسب است به محدوده [۱و۱-] نگاشت می‌شود. تابع زیان، اندازه وزن‌ها را با استفاده از نرم L2 جریمه می‌کند و مجموع مربعات خطاهای بین متغیر خروجی پیش‌بینی شده و متغیر خروجی واقعی را محاسبه می‌کند. قدرت منظم‌سازی L2 و نرخ یادگیری به‌ترتیب توسط فراپارامترهای $alpha$ و eta کنترل می‌شوند و ممکن است برای بهبود عملکرد الگوریتم در داده‌های اعتبارسنجی تنظیم شوند. بنابراین، طبقه‌بندی کننده خطی مؤثر که بتواند مجموعه داده‌های با ابعاد بالا را مدیریت کند طبقه‌بندی SGD است که نحوه کار آن شرح داده شد.

نتایج آزمایش

برای به‌دست آوردن میزان عملکرد روش پیشنهادی نیاز است تا معیارهای مشخصی برای اندازه‌گیری عملکرد سیستم داشته باشیم. همچنین باید مشخص شود چگونه عمل آموزش و آزمایش بر ر وی داده‌ها صورت گرفته است. در ادامه به معرفی مجموعه داده‌های استفاده شده، معیارهایی که در آزمایش‌ها اندازه‌گیری شده و نوع اعتبارسنجی داده‌ها برای آموزش و آزمایش مدل پیشنهادی می‌پردازیم.

مجموعه داده‌های استفاده شده

دو نوع مجموعه داده در این مقاله برای مدل‌سازی‌ها و ارزیابی‌ها استفاده شده است که در ادامه به تشریح آنها می‌پردازیم.

مجموعه داده CTU-۱۳

مجموعه داده CTU-۱۳ [۱۹] که به‌صورت عمومی در دسترس قرار دارد برای انجام آزمایش‌ها در این مقاله استفاده می‌شود. این مجموعه داده در سال ۲۰۱۱، از دانشگاهی واقع در جمهوری چک به نام دانشگاه CTU به‌دست آمد. هدف اصلی این مجموعه داده، ضبط ترافیک واقعی حملات باتنت و نمونه‌های ترافیک عادی در شبکه است. سیزده سناریوی مربوط به بدافزار برای ایجاد مجموعه داده CTU-۱۳ اجرا شد. انواع حملات شبکه موجود در مجموعه داده CTU-۱۳ عبارتند از IRC، تقلب کلیک[1]، منع سرویس[2]، پویش پورت[3]، ترافیک هرزنامه و شار سریع[4] [۲۰]. مجموعه داده استفاده شده بر اساس ۵۸ ویژگی مربوط به ترافیک عادی و حمله باتنت است. این مجموعه داده شامل برچسب‌های ترافیک عادی با مقدار صفر و ترافیک حمله باتنت با مقدار یک است. تجزیه و تحلیل توزیع داده نشان می‌دهد که مجموعه داده نامتعادل است. ترافیک عادی شامل ۵۳۳۱۴ نمونه و ترافیک حمله باتنت شامل ۳۸۸۹۸ نمونه است.

مجموعه داده BoT-IoT

مجموعه داده BoT-IoT با طراحی یک محیط شبکه واقعی در آزمایشگاه Cyber Range در UNSW Canberra ایجاد شد. محیط شبکه ترکیبی از ترافیک عادی و باتنت را در خود جای داده است. فایل‌های منبع مجموعه داده در

[1] Click Fraud
[2] DoS
[3] Port Scan
[4] Fast Flux

قالب‌های مختلف از جمله فایل‌های pcap اصلی، فایل‌های argus تولید شده و فایل‌های csv ارائه شده‌اند. فایل‌ها برای برچسب‌گذاری بهتر بر اساس دسته‌ی حمله به‌طور جداگانه طبقه‌بندی شده‌اند.

حجم فایل‌های pcap گرفته شده ۶۹.۳ گیگابایت و بیش از ۷۲.۰۰۰.۰۰۰ رکورد است. حجم ترافیک استخراج شده با فرمت csv ۱۶.۷ گیگابایت است. مجموعه داده شامل حملات DDoS، DoS، OS، Service Scan، Keylogging و Exfiltration Data است که حملات DDoS و DoS بر اساس پروتکل مورد استفاده بیشتر سازمان‌دهی شده است. برای سهولت مدیریت مجموعه داده، ۵٪ از مجموعه داده اصلی، با استفاده از جستجوهای MySQL استخراج می‌شوند. ۵٪ استخراج شده از ۴ فایل با حجم کلی ۱.۰۷ گیگابایت و حدود ۳ میلیون رکورد تشکیل شده است.

آموزش و آزمایش مدل

در این مقاله برای آموزش و آزمایش مدل از دو روش اعتبارسنجی k-fold و تقسیم درصد[1] [۲۱] استفاده می‌شود. اعتبارسنجی k-fold با مقادیر مختلف k، مجموعه داده را به k زیرمجموعه برابر تقسیم می‌کند به‌گونه‌ای که تعداد k-1 تا از زیرمجموعه‌های به‌دست آمده برای آموزش مدل و تنها زیرمجموعه باقی‌مانده برای آزمایش مدل و محاسبه معیارهای ارزیابی استفاده می‌شود. این عملیات k بار تکرار می‌شود و میانگین مقادیر به‌دست آمده از معیارها به‌عنوان مقدار نهایی گزارش می‌شود.

روش تقسیم درصد نیز این گونه عمل می‌کند که درصدی از داده‌ها برای آموزش و مابقی داده‌ها برای آزمایش استفاده می‌شود. در اینجا برای به‌دست آوردن بهترین نتایج، درصد نمونه‌های آموزش را از ۶۰٪ الی ۹۰٪ (متعاقباً درصد داده‌های آزمایش بین ۱۰٪ الی ۴۰٪) تغییر می‌دهیم و برای هر یک نتایج را بررسی می‌کنیم.

معیارهای ارزیابی

با توجه به اینکه دو نوع دسته‌بندی دودویی و چند کلاسه در آزمایش‌های انجام شده مد نظر قرار دارد و همچنین با توجه به اینکه تمرکز زیادی بر روی کاهش نرخ هشدارهای نادرست داریم، در اینجا پنج معیار با توضیحات زیر را برای مجموعه داده به‌دست می‌آوریم:

- **الف) نرخ مثبت کاذب (FPR):** این معیار به انتظار نسبت مثبت کاذب اشاره دارد و از رابطه ۳ به‌دست می‌آید.

$$FPR = \frac{FP}{FP + TN} \quad (۳)$$

که در این رابطه FP بیانگر تعداد داده‌های عادی است که به اشتباه حمله در نظر گرفته شده‌اند و TN بیانگر تعداد داده‌های عادی است که به‌درستی عادی پیش‌بینی شده‌اند.

- **ب) نرخ منفی کاذب (FNR):** این معیار به انتظار منفی کاذب اشاره دارد و از رابطه ۴ به‌دست می‌آید.

$$FNR = \frac{FN}{FN + TP} \quad (۴)$$

- **ج) دقت (Accuracy):** این معیار، تقسیم تعداد داده‌های درست دسته‌بندی شده به تعداد کل داده‌ها را نشان می‌دهد و از رابطه ۵ به‌دست می‌آید.

[1] Percentage split

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (۵)$$

- **د) امتیاز F1 (F1 score):** این پارامترکه نشان دهنده میانگین هارمونیک دو معیار دیگر به نام‌های Precision و Recall است از رابطه ۶ به‌دست می‌آید.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (۶)$$

- **ه) نرخ خطای کلی (Error rate):** این نرخ نشان دهنده خطای کلی مدل است و از رابطه ۷ به‌دست می‌آید.

$$Error\ rate = \frac{FP + FN}{TP + FP + TN + FN} \quad (۷)$$

نتایج

در این قسمت، نتایج به‌دست آمده از انجام حالت‌های مختلف مقادیر برای روش اعتبارسنجی و معیارهای ارزیابی در دسته‌بندی دودویی و دسته‌بندی چند کلاسه را نشان می‌دهیم.

نتایج دسته‌بندی دودویی

در این بخش، مقادیر به‌دست آمده از آزمایش‌های انجام شده برای حالت دسته‌بندی دودویی را نشان می‌دهیم. جدول ۱، مقادیر پارامترهای ارزیابی را برای درصدهای مختلف نمونه داده‌های آموزشی و مقادیر مختلف k در حالت اعتبارسنجی k-fold برای مجموعه داده CTU-۱۳ و جدول ۲ نیز نتایج را برای مجموعه داده BoT-IoT نشان می‌دهد (توضیح اینکه برای هر روش، تنها سه مقدار از بهترین مقادیر لیست شده‌اند). همان‌طور که در جدول ۱ مشاهده می‌شود بهترین نتیجه برای پارامترها در حالت اعتبارسنجی زمانی به‌دست می‌آید که از روش اعتبارسنجی k-fold با مقدار k=8 استفاده شده و در حالت تقسیم درصد، زمانی است که از کل مجموعه داده، ۶۳ ٪ نمونه داده برای آموزش و ۳۷٪ داده برای آزمایش استفاده می‌شود. همچنین بهترین حالت از بین این دو روش نیز استفاده از روش تقسیم درصد با ۶۳٪ داده آموزشی است که مقادیر مورد نظر هر یک از معیارها در جدول ۱ پررنگ نشان داده شده‌اند.

از روی مقادیر به‌دست آمده در جدول ۲ مشاهده می‌گردد که برای مجموعه داده BoT-IoT، بهترین نتیجه برای پارامترها در حالت اعتبارسنجی زمانی به‌دست می‌آید که از روش اعتبارسنجی k-fold با مقدار k=۸ استفاده شده و در حالت تقسیم درصد، زمانی است که از کل مجموعه داده، ۶۶ ٪ نمونه داده برای آموزش و ۳۴٪ داده برای آزمایش استفاده می‌شود. همچنین بهترین حالت از بین این دو روش نیز استفاده از روش تقسیم درصد با ۶۶٪ داده آموزشی است که مقادیر مورد نظر هر یک از معیارها در جدول ۲ پررنگ نشان داده شده‌اند.

جدول ۱. نتایج دسته‌بندی دودویی با اعتبارسنجی و تقسیم درصد مجموعه داده CTU-۱۳.

معیار	اعتبارسنجی fold -۵	اعتبارسنجی fold -۸	اعتبارسنجی fold -۱۰	آموزشی ۶۳٪	آموزشی ۷۴٪	آموزشی ۷۶٪
نرخ مثبت کاذب	۰.۱۵	**۰.۰۲**	۰.۰۷	**۰.۰۱**	۰.۰۹	۰.۰۸
نرخ منفی کاذب	۰.۱۲	**۰.۰۱**	۰.۰۹	**۰.۰۱**	۰.۰۷	۰.۱۰
دقت	۹۴.۸۷	**۹۹.۹۷**	۹۷.۳۸	**۹۹.۹۹**	۹۸.۳۵	۹۶.۵۸

معیار	اعتبارسنجی fold -۵	اعتبارسنجی fold -۸	اعتبارسنجی fold -۱۰	آموزشی ۶۳٪	آموزشی ۷۴٪	آموزشی ۷۶٪
امتیاز F1	۹۳.۰۵	**۹۹.۷۶**	۹۶.۴۴	**۹۹.۸۱**	۹۷.۵۵	۹۵.۸۱
نرخ خطای کلی	۵.۱۳	**۰.۰۳**	۲.۶۲	**۰.۰۱**	۱.۶۵	۳.۴۲

جدول ۲. نتایج دسته‌بندی دودویی با اعتبارسنجی و تقسیم درصد مجموعه داده BoT-IoT.

معیار	اعتبارسنجی fold -۶	اعتبارسنجی fold -۸	اعتبارسنجی fold -۱۰	آموزشی ۶۱٪	آموزشی ۶۶٪	آموزشی ۷۰٪
نرخ مثبت کاذب	۰.۱۷	**۰.۰۵**	۰.۰۹	۰.۰۳	**۰.۰۲**	۰.۰۹
نرخ منفی کاذب	۰.۱۶	**۰.۰۳**	۰.۱۴	۰.۰۴	**۰.۰۳**	۰.۱۳
دقت	۹۴.۲۱	**۹۹.۸۳**	۹۷.۱۰	۹۹.۷۶	**۹۹.۸۱**	۹۵.۴۲
امتیاز F1	۹۲.۱۶	**۹۹.۵۵**	۹۵.۲۲	۹۹.۰۵	**۹۹.۲۴**	۹۷.۷۴
نرخ خطای کلی	۵.۷۹	**۰.۱۷**	۲.۹۰	۰.۲۴	**۰.۱۹**	۴.۵۸

نتایج دسته‌بندی چند کلاسه

در این بخش نیز مقادیر به‌دست آمده از آزمایش‌های انجام شده برای حالت دسته‌بندی چند کلاسه را نشان می‌دهیم. جدول ۳ و ۴ همانند جداول ۱ و ۲، مقادیر پارامترهای ارزیابی را برای درصدهای مختلف نمونه داده‌های آموزشی و مقادیر مختلف k در حالت اعتبارسنجی k-fold برای دسته‌بندی چند کلاسه نشان می‌دهد. همان‌طور که در این جدول ۳ مشاهده می‌شود بهترین نتیجه برای پارامترها برای مجموعه داده CTU-۱۳ زمانی به‌دست می‌آید که از روش اعتبارسنجی k-fold با مقدار k=۷ و یا از روش اعتبارسنجی با ۶۹ ٪ نمونه داده آموزشی و ۳۱٪ داده آزمایشی استفاده می‌شود. نتایج به‌دست آمده در این جدول نیز نشان می‌دهد که در کل، نتایج مربوط به روش تقسیم درصد از روش اعتبارسنجی بهتر است. در مجموع نیز نتایج به‌دست آمده از دسته‌بندی دودویی بهتر از حالت دسته-بندی چند کلاسه است.

از روی مقادیر جدول ۴ مشاده می‌گردد که برای مجموعه داده BoT-IoT بهترین نتیجه برای پارامترها زمانی به‌دست می‌آید که از روش اعتبارسنجی k-fold با مقدار k=۷ و یا از روش اعتبارسنجی با ۶۸ ٪ نمونه داده آموزشی و ۳۲٪ داده آزمایشی استفاده می‌شود. نتایج به‌دست آمده در این جدول نشان می‌دهد که در کل، نتایج مربوط به روش اعتبارسنجی از روش تقسیم درصد بهتر است. در مجموع نیز نتایج به‌دست آمده از دسته‌بندی دودویی بهتر از حالت دسته‌بندی چند کلاسه است.

جدول ۳. نتایج دسته‌بندی چند کلاسه با اعتبارسنجی و تقسیم درصد مجموعه داده CTU-۱۳.

معیار	اعتبارسنجی fold -۶	اعتبارسنجی fold -۷	اعتبارسنجی fold -۹	آموزشی ۶۷٪	آموزشی ۶۹٪	آموزشی ۷۲٪
نرخ مثبت کاذب	۰.۱۳	**۰.۰۵**	۰.۰۷	۰.۰۷	**۰.۰۴**	۰.۰۸
نرخ منفی کاذب	۰.۱۱	**۰.۰۵**	۰.۰۸	۰.۰۶	**۰.۰۵**	۰.۰۸
دقت	۹۶.۲۷	**۹۹.۰۳**	۹۷.۲۰	۹۸.۳۳	**۹۹.۳۴**	۹۶.۲۸
امتیاز F1	۹۵.۷۲	**۹۸.۷۶**	۹۷.۳۸	۹۷.۴۱	**۹۹.۲۸**	۹۶.۲۱
نرخ خطای کلی	۳.۷۳	**۰.۹۷**	۲.۸۰	۱.۶۷	**۰.۶۶**	۳.۷۲

جدول ۴. نتایج دسته‌بندی چند کلاسه با اعتبارسنجی و تقسیم درصد مجموعه داده BoT-IoT.

معیار	اعتبارسنجی fold -۶	اعتبارسنجی fold -۷	اعتبارسنجی fold -۱۰	آموزشی ۶۴٪	آموزشی ۶۸٪	آموزشی ۷۱٪
نرخ مثبت کاذب	۰.۱۶	**۰.۰۸**	۰.۱۵	۰.۱۱	**۰.۰۸**	۰.۲۱
نرخ منفی کاذب	۰.۱۸	**۰.۰۷**	۰.۲۱	۰.۱۴	**۰.۰۷**	۰.۱۸
دقت	۹۵.۴۴	**۹۸.۵۲**	۹۴.۳۳	۹۷.۴۴	**۹۷.۶۵**	۹۴.۹۳
امتیاز F1	۹۳.۸۱	**۹۷.۳۵**	۹۵.۸۸	۹۵.۸۴	**۹۸.۰۶**	۹۵.۲۶
نرخ خطای کلی	۴.۵۶	**۱.۴۸**	۵.۶۷	۲.۵۶	**۲.۳۵**	۵.۰۷

مقایسه

در این بخش، مقایسه‌ای بین نتایج به‌دست آمده از روش پیشنهادی با سایر روش‌های موجود مقایسه می‌کنیم. جدول ۵ و ۶، مقایسه بین مقدار بهترین نتیجه به‌دست آمده از مهم‌ترین معیارهای ارزیابی در روش پیشنهادی با برخی روش‌های دیگر یادگیری ماشین و هوش مصنوعی را به‌ترتیب برای حالت دسته‌بندی دودویی و چند کلاسه نشان می‌دهد. با مقایسه صورت گرفته می‌توان نتیجه گرفت که استفاده از روش‌های k- نزدیکترین همسایگی و گرادیان کاهشی می‌تواند منجر به بهبود دقت تشخیص نفوذ و به‌خصوص کاهش نرخ هشدارهای خطا در تشخیص باتنت‌ها در مجموعه داده شود.

جدول ۵. مقایسه نتایج دسته‌بندی دودویی روش پیشنهادی.

روش	نرخ منفی کاذب	نرخ مثبت کاذب	دقت
RF+LR [۲۲]	۶.۱۱	۸.۳۵	۹۸.۱۲
K-Means+RF [۲۳]	۰.۰۸	۰.۰۸	۹۴.۴۳
RF+AdaBoost [۲۴]	۰.۰۷	۰.۰۶	۹۴.۳۶
CNN+LSTM [۲۵]	۰.۰۵	۰.۰۷	۹۹.۲۸
SVM+DT [۲۶]	۰.۰۵	۰.۰۵	۹۹.۶۸
SGD+KNN (Proposed)	**۰.۰۱**	**۰.۰۱**	**۹۹.۹۹**

جدول ۶. مقایسه نتایج دسته‌بندی چند کلاسه روش پیشنهادی.

روش	نرخ منفی کاذب	نرخ مثبت کاذب	دقت
RF+LR [۲۲]	۰.۰۹	۰.۰۸	۹۶.۷۸
K-Means+RF [۲۳]	۰.۰۹	۰.۰۹	۹۲.۷۷
RF+AdaBoost [۲۴]	۰.۰۸	۰.۰۷	۹۰.۵۱
CNN+LSTM[۲۵]	۰.۰۹	۰.۰۹	۹۸.۴۸
SVM+DT [۲۶]	۰.۰۸	۰.۱۱	۹۹.۳۱
SGD+KNN (Proposed)	**۰.۰۴**	**۰.۰۵**	**۹۹.۳۴**

نتیجه‌گیری

با گسترش استفاده از فضای وب توسط کاربران، حمله مهاجمان به دستگاه‌های متصل به اینترنت نیز افزایش یافته است. تاکنون محققان زیادی برای مقابله با باتنت‌ها که منجر به آلودگی سیستم‌ها از راه دور می‌شوند راهکارهایی ارائه نمودند. اما از آنجایی که روش‌های پیشنهادی آنها از نرخ‌های هشدار نادرست بالا در تشخیص باتنت‌ها رنج می‌برند، بنابراین در این مقاله راهکار ترکیبی از دو الگوریتم یادگیری ماشین به نام‌های k- نزدیکترین همسایگی و گرادیان کاهش تصادفی، یکی برای کاهش ویژگی‌ها و دیگری را برای دسته‌بندی بهتر داده‌ها استفاده نمودیم. نتایج به‌دست آمده از آزمایش‌های صورت گرفته و مقایسه آنها با برخی روش‌های دیگر نشان می‌دهد که ترکیب این دو الگوریتم استفاده شده می‌تواند باعث بهبود دقت تشخیص و به‌خصوص کاهش نرخ مثبت کاذب و منفی کاذب در تشخیص باتنت‌ها شود.

به‌عنوان کارهای آتی برای بهبود دقت طبقه‌بندی داده‌ها می‌توان روش‌های استفاده شده در این مقاله را با سایر روش‌های مبتنی بر هوش مصنوعی مانند الگوریتم کرم شب‌تاب، الگوریتم پرندگان و یا غیره ترکیب نمود. استفاده از این الگوریتم‌ها با توجه با ماهیت بهینه‌سازی که دارند منجر به بهبود روش‌های استفاده شده برای طبقه‌بندی و کاهش نرخ هشدارهای نادرست در تشخیص باتنت‌ها خواهد شد.

References

[1] Debicha, I., Cochez, B., Kenaza, T., Debatty, T., Dricot, J-M., & Mees, W. (2023). Adv-Bot: Realistic adversarial botnet attacks against network intrusion detection systems. *Computers & Security*, *129*(4), 103176. https://doi.org/10.1016/j.cose.2023.103176

[2] He, K., Kim, D. D., & Asghar, M. R. (2023). Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey. *Institute of Electrical and Electronics Engineers Communications Surveys & Tutorials*, *25*(1), 538-566. https://doi.org/10.1109/COMST.2022.3233793

[3] Raza, A., Siddiqui, H. U. R., Munir, K., Almutairi, M., Rustam, F., & Ashraf, I. (2022). Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction. *Plos one*, *17*(11), e0276525. https://doi.org/10.1371/jour nal.pone.0276525

[4] Noori, A. (2022). A New Method for Detecting Influential Nodes in Social Network Graphs Using Deep Learning Techniques. *Karafan Quarterly Scientific Journal*, *19*(1), 607-628. https://doi.org/10.48301/kssa.2022.310565.1786

[5] Ibrahim, W. N. H., Anuar, S., Selamat, A., Krejcar, O., Crespo, R. G., Herrera-Viedma, E., & Fujita, H. (2021). Multilayer Framework for Botnet Detection Using Machine Learning Algorithms. *Institute of Electrical and Electronics Engineers Access*, *9*, 48753-48768. https://doi.org/10.1109/ACCESS.2021.3060778

[6] Dollah, R. F. M., Faizal, M. A., Arif, F., Mas'ud, M. Z., & Xin, L. K. (2018). Machine Learning for HTTP Botnet Detection Using Classifier Algorithms. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1-7), 27-30. https://jtec.utem.edu.my/jtec/article/view/3591

[7] Lee, S., Abdullah, A., Jhanjhi, N., & Kok, S. (2021). Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning. *Peer Journal Computer Science*, *7*(6), e350. https://doi.org/10.7717/peerj-cs.350

[8] Khan, R. U., Zhang, X., Kumar, R., Sharif, A., Golilarz, N. A., & Alazab, M. (2019). An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers. *Applied Sciences*, *9*(11), 2375. https://doi.org/10.3390/app9112375

[9] Alkahtani, H., & Aldhyani, T. H. (2021). Botnet attack detection by using CNN-LSTM model for Internet of Things applications. *Security and Communication Networks*, *2021*, 1-23. https://doi.org/10.1155/2021/3806459

[10] Alissa, K., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N., & Sakib, S. (2022). Botnet attack detection in iot using machine learning. *Computational Intelligence and Neuroscience*, *2022*, 1-14. https://doi.org/10.1155/2022/4515642

[11] Rustam, F., Raza, A., Ashraf, I., & Jurcut, A. D. (2023, June 13-15). *Deep Ensemble-based Efficient Framework for Network Attack Detection.* 2023 21st Mediterranean Communication and Computer Networking Conference, Island of Ponza, Italy. https://doi.org/10.1109/MedComNet58619.2023.10168864

[12] Bojarajulu, B., Tanwar, S., & Singh, T. P. (2023). Intelligent IoT-BOTNET attack detection model with optimized hybrid classification model. *Computers & Security*, *126*(2), 103064. https://doi.org/10.1016/j.cose.2022.103064

[13] Karthik, M. G., & Krishnan, M. B. M. (2021). Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks. *Journal of Ambient Intelligence and Humanized Computing*, 1-11. https://doi.org/10.1007/s12652-021-03082-3

[14] Moorthy, R. S., & Pabitha, P. (2020). Optimal Detection of Phising Attack using SCA based K-NN. *Procedia Computer Science*, *171*, 1716-1725. https://doi.org/10.1016/j.procs.2020.04.184

[15] Liao, Y., & Vemuri, V. R. (2002). Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security*, *21*(5), 439-448. https://doi.org/10.1016/S0167-4048(02)00514-X

[16] Jahromi, A. H., & Taheri, M. (2017, October 25-27). *A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features.* 2017 Artificial Intelligence and Signal Processing Conference, Shiraz, Iran. https://doi.org/10.1109/AISP.2017.8324083

[17] Peppes, N., Daskalakis, E., Alexakis, T., Adamopoulou, E., & Demestichas, K. (2021). Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *Sensors*, *21*(22), 7475. https://doi.org/10.3390/s21227475

[18] Bottou, L. (2012). Stochastic Gradient Descent Tricks. In G. Montavon, G. B. Orr, & K-R. Müller (Eds.), *Neural Networks: Tricks of the Trade: Second Edition* (2 ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_25

[19] García, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, *45*, 100-123. https://doi.org/10.1016/j.cose.2014.05.011

[20] Ahmed, S., Khan, Z. A., Mohsin, S. M., Latif, S., Aslam, S., Mujlid, H., Adil, M., & Najam, Z. (2023). Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron. *Future Internet*, *15*(2), 76. https://doi.org/10.3390/fi15020076

[21] Gong, D., & Liu, Y. (2022, May 20-22). *A Machine Learning Approach for Botnet Detection Using LightGBM.* 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications, Changchun, China. https://doi.org/10.1109/CVIDLICCEA56201.2022.9824033

[22] Waskle, S., Parashar, L., & Singh, U. (2020, July 2-4). *Intrusion Detection System Using PCA with Random Forest Approach.* 2020 International Conference on Electronics

and Sustainable Communication Systems, Coimbatore, India. https://doi.org/10.1109/ICESC48915.2020.9155656

[23] Samunnisa, K., Kumar, G. S. V., & Madhavi, K. (2023). Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. *Measurement: Sensors*, *25*, 100612. https://doi.org/10.1016/j.measen.2022.100612

[24] Dietterich, T. G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, *40*(2), 139-157. https://doi.org/10.1023/A:1007607513941

[25] Sivamohan, S., Sridhar, S. S., & Krishnaveni, S. (2021, June 25-27). *An Effective Recurrent Neural Network (RNN) based Intrusion Detection via Bi-directional Long Short-Term Memory*. 2021 International Conference on Intelligent Technologies, Hubli, India. https://doi.org/10.1109/CONIT51480.2021.9498552

[26] Wang, H., Gu, J., & Wang, S. (2017). An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems*, *136*, 130-139. https://doi.org/10.1016/j.knosys.2017.09.014