«علوم و فناوری‌های پدافند نوین»

سال پانزدهم، شماره ۴، زمستان ۱۴۰۳؛ ص ۲۲۹- ۲۱۷

علمی- پژوهشی

ارزیابی و بهینه‌سازی مدل‌های یادگیری ماشین در سیستم‌های تشخیص نفوذ با استفاده از روش‌های کاهش ابعاد PCA و ICA

علی عظیمی[۱]، مهنا فاتح [۲]، سینا صمدی قره ورن[۳]*

۱ و ۲- کارشناسی دانشگاه صنعتی سهند- تبریز- ایران، ۳- استادیار دانشگاه تبریز- تبریز- ایران

(دریافت: ۱۴۰۳/۱۰/۱۰، بازنگری: ۱۴۰۳/۱۱/۰۶، پذیرش: ۱۴۰۳/۱۱/۲۴، انتشار: ۱۴۰۳/۱۲/۰۳)

چکیده

با گسترش روزافزون تهدیدات سایبری، توسعه سیستم‌های تشخیص نفوذ کارا و دقیق به یکی از چالش‌های اساسی در امنیت شبکه تبدیل‌شده است. در این مقاله، عملکرد شش مدل یادگیری ماشین شامل KNN، SVM، Random Forest، Decision Tree، Logistic Regression و XGBoost در تشخیص نفوذ شبکه ارزیابی و مقایسه شده است. به‌منظور بهبود کارایی محاسباتی و کاهش اثر ویژگی‌های غیرضروری، از دو روش کاهش ابعاد تحلیل مؤلفه‌های اصلی (PCA) و تحلیل مؤلفه‌های مستقل (ICA) استفاده گردید. مجموعه داده UNSW-NB۱۵ به‌عنوان داده مرجع انتخاب شد و ارزیابی مدل‌ها با معیارهای Accuracy، Precision، Recall، F۱-Score و زمان آموزش/پیش‌بینی انجام گرفت. نتایج نشان داد Random Forest و XGBoost بهترین عملکرد کلی را ارائه داده و حتی با کاهش ابعاد توسط ICA دقت بالای خود را حفظ کردند. ICA در اغلب مدل‌ها نسبت به PCA عملکرد بهتری داشت و توانست تعادل مطلوبی بین دقت و کارایی ایجاد کند. یافته‌های این مقاله می‌تواند به انتخاب مدل مناسب IDS بر اساس نیازهای عملیاتی، ازجمله اولویت کاهش هشدارهای کاذب یا افزایش نرخ تشخیص، کمک نماید.

کلیدواژه‌ها: سیستم تشخیص نفوذ، یادگیری ماشین، کاهش ابعاد، UNSW-NB۱۵.

Evaluation and Optimization of Machine Learning Models in Intrusion Detection Systems Using PCA and ICA Dimensionality Reduction Methods

A. Azimi, M. Fateh, S. Samadi Gharehveran*

Tabriz University, Tabriz, Iran

(Received: ۲۰۲٤/۱۲/۳۰, Revised: ۲۰۲٥/۰۱/۲٥, Accepted: ۲۰۲٥/۰۲/۱۲, Published: ۲۰۲٥/۰۲/۲۱)

Abstract

Intrusion Detection Systems (IDS) play a crucial role in protecting modern computer networks from diverse cyberattacks. However, the high dimensionality and complexity of network traffic data often degrade the accuracy and efficiency of machine learning–based IDS models. This paper proposes a comprehensive comparative framework that adaptively integrates two dimensionality reduction methods—Principal Component Analysis (PCA) and Independent Component Analysis (ICA)—to enhance IDS performance. Six widely adopted machine learning algorithms—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree, Logistic Regression, and XGBoost—are evaluated using the modern UNSW-NB۱۵ dataset. The performance of each model is assessed based on classical evaluation metrics (Accuracy, Precision, Recall, and F۱-Score) as well as operational efficiency indicators (training and prediction time). Experimental results demonstrate that ICA generally outperforms PCA, achieving a better balance between detection accuracy and computational cost. The findings provide valuable insights for designing practical, efficient, and high-performance IDS solutions for real-world applications.

Keywords: Intrusion Detection System, Machine Learning, Dimensionality Reduction, UNSW-NB۱۵.

*Corresponding Author: s.samadi@tabrizu.ac.ir

۱. مقدمه

آمارهای سال ۲۰۲۳ نشان می‌دهد حدود ۶۳ درصد از جمعیت جهان به اینترنت دسترسی دارند و این رقم به‌طور روزانه در حال افزایش است [۱]. امروزه زندگی انسان‌ها به‌شدت با این شبکه گسترده و پیچیده گره‌خورده و بسیاری از رویدادهای مهم و حیاتی را تسریع می‌کند. اینترنت به بخشی جدایی‌ناپذیر از محیط زندگی ما تبدیل‌شده و سهم قابل‌توجهی از فعالیت‌های روزمره، از دوران کودکی تا کهن‌سالی، در بستر آن شکل می‌گیرد[۲].

اینترنت، همانند هر محیط دیگری، نیازمند رعایت اصول بنیادینی همچون امنیت است. در محیط‌های گوناگون، معمولاً افراد آموزش می‌بینند تا این امنیت را برقرار کنند، اما امروزه ماشین‌ها نیز با سرعت و دقت چشمگیری آموزش می‌بینند تا بتوانند فعالیت‌ها را به‌طور مداوم پایش کرده و رفتارهای پرخطر را شناسایی نمایند. این رویکرد به‌ویژه در نسل جدید سیستم‌های تشخیص نفوذ[۱] نمود یافته است. در این مقاله، به بررسی نقش مدل‌های یادگیری ماشین در این حوزه پرداخته‌شده و عملکرد شش الگوریتم شامل نزدیک‌ترین همسایه[۲]، ماشین بردار پشتیبان[۳]، جنگل تصادفی[۴]، درخت تصمیم[۵]، رگرسیون لجستیک[۶] و تقویت گرادیانی شدید[۷] مورد ارزیابی قرارگرفته است. مقایسه مدل‌ها بر اساس معیارهای دقت[۸]، دقت مثبت[۹]، فراخوانی[۱۰] و امتیاز F1 [۱۱] انجام می‌شود [۳].

دقت بیانگر نسبت پیش‌بینی‌های صحیح به‌کل پیش‌بینی‌هاست. دقت مثبت نشان می‌دهد از میان تمام نمونه‌هایی که به‌عنوان حمله پیش‌بینی‌شده‌اند، چه درصدی واقعاً حمله بوده‌اند. فراخوانی بیانگر آن است که از میان تمام حملات واقعی، چه درصدی به‌درستی شناسایی‌شده‌اند. امتیاز F1 نیز میانگین هماهنگ دقت مثبت و فراخوانی است که معیار متعادلی برای ارزیابی عملکرد مدل در شرایط عدم توازن داده‌ها فراهم می‌کند. [۴].

در این مقاله از مجموعه داده UNSW-NB۱۵ استفاده‌شده است که به‌عنوان یک دیتاست مدرن و متنوع در حوزه تشخیص نفوذ شناخته می‌شود. این مجموعه شامل انواع حملات سایبری ازجمله ابزارهای تولید ورودی خراب، شناسایی، کد پوسته‌ای، تحلیل، درِ پشتی، حمله انکار سرویس، بهره‌برداری‌ها، حملات عمومی و کرم‌ها است [۵]. UNSW-NB۱۵ دارای ۴۹ ویژگی مرتبط با ترافیک شبکه است که به‌طور خلاصه شامل اطلاعات‌پایه مانند آدرس و پورت مبدأ و مقصد، پروتکل‌های مورداستفاده، وضعیت اتصال (مانند FIN [۱۲] و RST [۱۳])، شاخص‌های عملکردی نظیر مدت‌زمان اتصال، حجم داده مبادله شده، زمان حیات بسته‌ها و نرخ انتقال داده، و نیز ویژگی‌های آماری مانند تغییرات تأخیر در مبدأ و مقصد، زمان تأخیر در برقراری ارتباط TCP، وضعیت ورود پروتکل FTP، و تعداد درخواست‌های HTTP (مانند GET/POST) است [۶]. تمامی این ویژگی‌ها در مدل اصلی لحاظ شده‌اند، بااین‌حال به دلیل ابعاد بالای داده، دو روش کاهش ابعاد PCA و ICA نیز به‌کاررفته تا ضمن بهبود کارایی محاسباتی، امکان مقایسه عملکرد مدل‌ها در شرایط مختلف فراهم شود. نوآوری اصلی این مقاله در چند محور قابل‌توجه است. نخست، برخلاف اغلب مطالعات پیشین که تمامی ویژگی‌های دیتاست را بدون کاهش بعد به کار گرفته‌اند، در این تحقیق برای اولین بار در حوزه تشخیص نفوذ، دو رویکرد PCA و ICA به‌صورت تطبیقی استفاده‌شده تا با حذف ویژگی‌های زائد، ضمن حفظ دقت مدل‌ها، سرعت آموزش و پیش‌بینی به‌طور چشمگیری افزایش یابد. دوم، این مقاله یک چارچوب مقایسه‌ای جامع میان شش الگوریتم پرکاربرد یادگیری ماشین (KNN، SVM، Random Forest، Decision Tree، Logistic Regression و XGBoost) ارائه کرده است که نه‌تنها دقت و معیارهای کلاسیک را پوشش می‌دهد، بلکه ملاحظات زمانی را نیز در ارزیابی لحاظ می‌کند؛ موضوعی که برای کاربردهای عملی IDSها در محیط‌های واقعی بسیار حیاتی است. نهایتاً، نتایج این تحقیق نشان می‌دهد که استفاده از کاهش بعد، به‌ویژه روش ICA، می‌تواند تعادل مطلوبی بین دقت و کارایی ایجاد کرده و یک رویکرد نوآورانه برای بهبود امنیت شبکه‌های مدرن ارائه دهد.

در این مقاله، نوآوری‌های اصلی به چند وجه برجسته شده‌اند: اول، کاهش بعد تطبیقی با استفاده از PCA و ICA به‌طور هم‌زمان برای نخستین بار بررسی‌شده است تا تأثیر آن‌ها بر عملکرد طیف وسیعی از الگوریتم‌های یادگیری ماشین (KNN، SVM، Random Forest، Decision Tree، Logistic Regression و XGBoost) تحلیل شود. دوم، یک چارچوب مقایسه‌ای جامع طراحی‌شده است که علاوه بر شاخص‌های کلاسیک دقت (Accuracy، Precision، Recall، F۱-Score)، معیارهای عملیاتی شامل زمان آموزش و پیش‌بینی نیز لحاظ شده‌اند؛ موضوعی که در کاربردهای واقعی سیستم‌های تشخیص نفوذ بسیار حیاتی است. سوم، این مطالعه بر روی دیتاست مدرن و پیچیده UNSW-NB۱۵ انجام‌شده است که شامل حملات جدید و متنوع است، درحالی‌که بسیاری از مطالعات پیشین بر روی دیتا

[۱] Intrusion Detection System (IDS)
[۲] K-Nearest Neighbors (KNN)
[۳] Support Vector Machine (SVM)
[۴] Random Forest (RF)
[۵] Decision Tree
[۶] Logistic Regression
[۷] Extreme Gradient Boosting (XGBoost)
[۸] Accuracy
[۹] Precision
[۱۰] Recall
[۱۱] F۱-Score
[۱۲] Finish
[۱۳] Reset

ست‌های قدیمی مانند KDDCup۹۹ و NSL-KDD محدود مانده‌اند. به‌این‌ترتیب، مقاله حاضر فراتر از یک مقایسه ساده، یک چارچوب نوآورانه و کاربردی برای طراحی IDS مدرن ارائه می‌دهد.

در ادامه، ساختار مقاله به شرح زیر سازمان‌دهی شده است. در بخش دوم، روش تحقیق تشریح می‌شود که شامل معرفی مجموعه داده، فرایند پیش‌پردازش، به‌کارگیری روش‌های کاهش بُعد PCA و ICA، و مدل‌های یادگیری ماشین مورداستفاده است. در بخش سوم، نتایج شبیه‌سازی و ارزیابی عملکرد مدل‌ها ارائه و تحلیل می‌شود. در بخش چهارم، جمع‌بندی و نتیجه‌گیری کلی بیان‌شده و پیشنهادهایی برای تحقیقات آتی ارائه می‌گردد. در پایان نیز، فهرست منابع مورداستفاده آورده شده است.

۲. پیشینه تحقیق

یکی از بخش‌های کلیدی معماری‌های امنیتی مدرن، سیستم‌های تشخیص نفوذ هستند که نقشی اساسی در شناسایی و مقابله با تهدیدات سایبری ایفا می‌کنند. در سال‌های اخیر، با افزایش تنوع و پیچیدگی روش‌های نفوذ، استفاده از یادگیری ماشین در این حوزه به‌طور چشمگیری گسترش‌یافته است؛ زیرا الگوریتم‌های یادگیری ماشین قادرند الگوهای پیچیده حملات را فراگیرند و شناسایی کنند، درحالی‌که روش‌های سنتی مبتنی بر امضا در مواجهه با تهدیدات نوظهور کارایی کمتری دارند. در این بخش، مروری بر مقالات مبتنی بر یادگیری ماشین کلاسیک در حوزه سیستم‌های تشخیص نفوذ ارائه می‌شود.

پیشرفت‌های اولیه در سیستم‌های تشخیص نفوذ مبتنی بر یادگیری ماشین عمدتاً بر استفاده از مجموعه داده‌های قدیمی مانند KDD Cup۹۹ و NSL-KDD متمرکز بود. بااین‌حال، مطالعه [۷] نشان داد که این مجموعه داده‌ها با محدودیت‌هایی همچون تکرار داده‌ها و فقدان نمونه‌های مربوط به حملات مدرن مواجه هستند. این چالش‌ها پژوهشگران را بر آن داشت تا مجموعه داده‌های واقع‌گرایانه‌تری را معرفی کنند. در همین راستا، مصطفی و همکاران [۸] مجموعه داده UNSW-NB۱۵ را ارائه کردند که شامل ترافیک واقعی شبکه و ۹ نوع حمله مدرن شامل Fuzzers، Reconnaissance، Shellcode، Analysis، Backdoors، DoS، Exploits، Generic و Worm است و ۴۹ ویژگی متنوع را در برمی‌گیرد. این مجموعه داده به‌عنوان یک استاندارد معتبر در ارزیابی IDSها شناخته می‌شود و تحقیقات متعددی بر پایه آن انجام‌شده است. به‌عنوان نمونه، کورعا و همکاران [۹] در یک مطالعه جامع، عملکرد الگوریتم‌هایی همچون رگرسیون خطی، رگرسیون لجستیک، ماشین بردار پشتیبان خطی، KNN، جنگل تصادفی، درخت تصمیم و MLP را با استفاده از فن «حساسیت به انسداد» ارزیابی کردند و دریافتند که جنگل تصادفی با دقت ۹۰٪ و سرعت‌بالا بهترین عملکرد را دارد، به‌طوری‌که اکثر مدل‌ها تنها با کمتر از سه ویژگی کلیدی به این دقت دست‌یافته‌اند. همچنین، مالل و همکاران [۱۰] با رویکردی مشابه و تمرکز بر طبقه‌بندی چند کلاسه گزارش دادند که جنگل تصادفی با دقت ۹۹/۷ ٪ در رتبه نخست، XGBoost با ۹۹/۱٪ و KNN با ۹۷/۶٪ در رتبه‌های بعدی قرار دارند و KNN سریع‌ترین زمان آموزش (۱۷ ثانیه) را ارائه کرده است.

ذوقی و همکاران [۱۱] با تمرکز بر چالش‌های ذاتی مجموعه داده UNSW-NB۱۵ مانند عدم توازن کلاس‌ها و همپوشانی ویژگی‌ها، مدلی ترکیبی شامل بگینگ متعادل، XGBoost و جنگل تصادفی با درخت تصمیم مبتنی بر فاصله هلینگر ارائه کردند که در طبقه‌بندی باینری و چند کلاسه عملکرد برتری داشت. همچنین، پالانیسوانی و همکاران [۱۲] با ترکیب الگوریتم‌های متنوع و استفاده از روش‌های انتخاب ویژگی ترکیبی و موازنه کلاس‌ها (SMOTE)، روی سه مجموعه داده ازجمله UNSW-NB۱۵ به‌دقت بالای ۹۶٪ دست یافتند و نیز نشان دادند که انتخاب ویژگی مؤثر می‌تواند حتی در مدل‌های ساده‌ای مانند درخت تصمیم، عملکردی هم‌سطح مدل‌های پیچیده ایجاد کند. بااین‌حال، یکی از چالش‌های مهم UNSW-NB۱۵ توزیع نابرابر داده‌هاست (۸۷٪ ترافیک عادی در مقابل ۱۳٪ حمله).

عجابه و همکاران [۱۳] نشان دادند که استفاده از فن‌هایی مانند روش بیش نمونه‌گیری مصنوعی اقلیت می‌تواند دقت مدل‌ها را تا ۱۵٪ افزایش دهد. همچنین، باوجود ۴۹ ویژگی در UNSW-NB۱۵، کاهش ابعاد برای بهبود کارایی ضروری است. ساحید و همکاران [۱۴] نشان دادند که روش‌هایی مانند PCA و ICA می‌توانند ابعاد را تا ۶۰٪ کاهش دهند، اما تأثیر آن‌ها بر مدل‌های مختلف متفاوت است و هشدار دادند که مدل‌های آموزش‌دیده روی یک دیتاست ممکن است در برابر حملات نوظهور (Zero-Day) عملکرد ضعیفی داشته باشند و پیشنهاد کردند استفاده از یادگیری ترکیبی می‌تواند این مشکل را مرتفع کند. با وجود پیشرفت‌های اخیر، چند کاستی اساسی در تحقیقات پیشین حوزه سیستم‌های تشخیص نفوذ مشهود است.

۱. تمرکز محدود بر مدل‌ها: بسیاری از مقالات عمدتاً بر مدل‌های خاصی مانند جنگل تصادفی و XGBoost متمرکز بوده‌اند و مقایسه جامعی با مدل‌های پایه‌ای همچون KNN، ماشین بردار پشتیبان و رگرسیون لجستیک ارائه نکرده‌اند.

۲. نادیده گرفتن نقش کاهش ابعاد: آثاری مانند [۱۵] به بررسی تأثیر روش‌های کاهش ابعاد نظیر PCA و ICA بر عملکرد مدل‌ها نپرداخته‌اند، درحالی‌که این فن‌ها می‌توانند کار آیی محاسباتی را به‌طور چشمگیری افزایش دهند.

۳. محدودیت در معیارهای ارزیابی: بسیاری از مطالعات صرفاً از دقت به‌عنوان شاخص ارزیابی استفاده کرده‌اند، درحالی‌که معیارهایی چون دقت مثبت، فراخوانی و F1-Score برای کاهش هشدارهای کاذب و بهبود نرخ تشخیص اهمیت ویژه‌ای دارند[۱۶].

۴. بی‌توجهی به زمان محاسباتی: زمان آموزش و پیش‌بینی به‌ندرت در ارزیابی‌ها لحاظ شده است، درحالی‌که این عامل برای کاربردهای عملیاتی حیاتی محسوب می‌شود[۱۷].

در راستای پر کردن این شکاف‌ها، این مقاله مقایسه‌ای جامع و منصفانه میان شش مدل پایه‌ای یادگیری ماشین شامل KNN، ماشین بردار پشتیبان، جنگل تصادفی، درخت تصمیم، رگرسیون لجستیک و XGBoost بر روی مجموعه داده UNSW-NB۱۵ ارائه می‌دهد. نوآوری‌های کلیدی این تحقیق عبارت‌اند از:

۱. ارزیابی جامع مدل‌ها که کمتر به‌طور هم‌زمان در تحقیقات اخیر بررسی‌شده‌اند.

۲. تحلیل اثر کاهش ابعاد با استفاده از PCA و ICA و شناسایی مدل‌های مقاوم یا حساس به این فرایند.

۳. به‌کارگیری مجموعه‌ای کامل از معیارهای ارزیابی شامل Accuracy، دقت مثبت، فراخوانی، F1-Score و نیز زمان آموزش و پیش‌بینی.

۴. تمرکز بر کارایی عملیاتی و ایجاد تعادل بین دقت و سرعت در کاربردهای واقعی IDS.

این مقاله با ارائه بینش‌های کاربردی، می‌تواند راهنمای انتخاب مدل مناسب IDS بر اساس اولویت‌های امنیتی خاص، مانند کاهش هشدارهای کاذب یا افزایش نرخ تشخیص و درنتیجه گامی مؤثر در بهبود امنیت شبکه‌های مدرن باشد.

۳. روش تحقیق

در این مقاله، یک چارچوب جامع برای ارزیابی عملکرد مدل‌های یادگیری ماشین در تشخیص نفوذ شبکه با استفاده از مجموعه داده UNSW-NB۱۵ طراحی و پیاده‌سازی شد و به دلیل زیاد بودن ویژگی‌ها از روش‌های کاهش ابعاد استفاده شد. روش‌شناسی تحقیق شامل مراحل زیر است که به‌طور سامانمند توصیف می‌شوند.

۳-۱. مجموعه داده

مطالعه حاضر از مجموعه داده UNSW-NB۱۵ استفاده کرد که توسط [۱۸] معرفی‌شده است. این مجموعه داده شامل ترافیک شبکه واقعی با ۹ نوع حمله مدرن شامل Fuzzers، Reconnaissance، Shellcode، Analysis، Backdoors، DoS، Exploits، Generic، Worms و ترافیک عادی است. مجموعه داده شامل ۱۷۵,۳۴۱ نمونه آموزشی و ۸۲,۳۳۲ نمونه آزمایشی با ۴۹ ویژگی خام برای هر نمونه است. وظیفه طبقه‌بندی به‌صورت باینری تعریف شد تا ترافیک عادی (برچسب ۰) را از فعالیت‌های مخرب (برچسب ۱) تشخیص دهد.

۳-۲. انتخاب ویژگی

ویژگی‌های مجموعه داده UNSW-NB۱۵ شامل ۴۲ ویژگی است که برای تحلیل جامع ترافیک شبکه طراحی‌شده‌اند. ویژگی‌های پایه شامل مدت‌زمان ارتباط، پروتکل شبکه، سرویس شبکه و وضعیت ارتباط هستند که اساس تحلیل ترافیک را تشکیل می‌دهند. ویژگی‌های حجمی مانند تعداد پکت‌های مبدأ و مقصد، حجم داده‌های مبدأ و مقصد و نرخ پکت‌ها، میزان و حجم ترافیک شبکه را اندازه‌گیری می‌کنند.

ویژگی‌های زمانی و کیفی شامل مقدار TTL، نرخ انتقال داده، تعداد پکت‌های گمشده، فاصله زمانی بین پکت‌ها و نوسانات زمانی هستند که کیفیت و الگوهای زمانی ارتباطات را تحلیل می‌کنند. ویژگی‌های مخصوص TCP مانند اندازه پنجره TCP، زمان رفت‌وبرگشت، زمان‌های برقراری ارتباط و ویژگی‌های آماری شامل میانگین اندازه پکت‌ها، عمق تراکنش HTTP و طول پاسخ HTTP جزئیات فنی ارتباطات را پوشش می‌دهند [۱۹-۲۱].

ویژگی‌های شمارشی، مانند مبدأ و مقصد در سیستم رهگیری اتصال، زمان انقضای وضعیت اتصال، پورت مبدأ و مقصد در بخش مدیریت ترافیک محلی، تعداد ارتباطات با الگوهای خاص در بازه‌های زمانی را نشان می‌دهند. ویژگی‌های دودویی شامل وضعیت ورود FTP، تعداد دستورات FTP، تعداد درخواست‌های HTTP و تشابه IP/پورت، با ردیابی الگوهای تکراری و رفتارهای غیرعادی، امکان شناسایی حملات شبکه مانند DoS، اسکن، نفوذ و سوءاستفاده از سرویس‌ها را فراهم می‌کنند.

این مجموعه جامع با ترکیب داده‌های حجمی، زمانی، پروتکلی و آماری، ابزاری قدرتمند برای تحلیل ترافیک شبکه و تشخیص تهدیدات امنیتی محسوب می‌شود.

۳-۳. پیش‌پردازش داده‌ها

یک خط لوله پیش‌پردازش سامانمند پیاده‌سازی شد تا از کیفیت داده‌ها و سازگاری آن‌ها با الگوریتم‌های یادگیری ماشین اطمینان حاصل شود.

- کدگذاری ویژگی‌های دسته‌ای: ویژگی‌های دسته‌ای ('proto', 'service', 'state') با استفاده از کدگذاری برچسب به مقادیر عددی تبدیل شدند. این فرایند برای هر ستون به‌طور جداگانه انجام شد تا مقادیر متنی به برچسب‌های عددی منحصربه‌فرد نگاشت شوند[۲۲].

• استانداردسازی ویژگی‌های عددی: ویژگی‌های عددی با استفاده از نرمال‌سازی امتیاز z استاندارد شدند. این فرایند شامل محاسبه میانگین و انحراف معیار برای هر ویژگی و سپس تبدیل مقادیر به مقیاس استاندارد با میانگین صفر و انحراف معیار یک بود.

۳-۴. پیش‌پردازش داده‌ها

به دلیل زیاد بودن حجم ویژگی‌ها از دو فن کاهش ابعاد برای ارزیابی تأثیر آن‌ها بر عملکرد مدل‌ها و کارایی محاسباتی به کار گرفته‌شده است.

• **تحلیل مؤلفه‌های اصلی (PCA)**

به‌منظور کاهش ابعاد داده‌های دیتاست UNSW-NB۱۵ و مدیریت هم خطی بین ویژگی‌ها، از روش تحلیل مؤلفه‌های اصلی استفاده شد. ابتدا داده‌های ورودی ($X \in R^{n\times c}$ با n نمونه و d ویژگی) استانداردسازی شدند تا میانگین هر ویژگی صفر ($\mu = ۰$)و واریانس واحد ($\sigma^۲$=۱) گردد[۲۳].

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad i = 1,\dots,n;\ j = 1,\dots,d \qquad (۱)$$

سپس ماتریس کوواریانس $C \in R^{d\times d}$ توسط رابطه (۲) محاسبه می‌گردد [۲۳].

$$c = \frac{1}{n-1} Z^T Z \qquad (۲)$$

که در آن Z ماتریس داده‌های استانداردشده است. در گام بعد، مقادیر ویژه (λ_i) و بردارهای ویژه (v_i) ماتریس C از طریق حل معادله مشخصه $Cv_i = \lambda_i v_i$ استخراج شدند. مؤلفه‌های اصلی بر اساس بزرگی مقادیر ویژه $(\lambda_۱\lambda_۱ \geq \lambda_۲ \geq \cdots \geq \lambda_d)$ مرتب‌سازی و تعداد بهینه مؤلفه‌های نگهداری شده (k) با حفظ حداقل ۹۵٪ واریانس تجمعی توسط رابطه (۳) تعیین گردید.

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{k} \lambda_i} \geq 0.95 \qquad (۳)$$

درنهایت، داده‌های کاهش‌یافته $T \in R^{n\times k}$ با پرو جکت کردن داده‌های استانداردشده Z روی فضای جدید مؤلفه‌های اصلی $(W = [v_۱, v_۲, \dots, v_k])$ حاصل‌شده است: $T = ZW$

این فرایند ضمن کاهش ابعاد از d به (k)، بیشترین واریانس داده‌ها را حفظ کرده و پیچیدگی محاسباتی مدل‌های یادگیری ماشین را کاهش می‌دهد.

• **تحلیل مؤلفه‌های مستقل (ICA)**

به‌منظور استخراج ویژگی‌های مستقل و کاهش ابعاد داده‌های دیتاست UNSW-NB۱۵، از روش تحلیل مؤلفه‌های مستقل استفاده شد. ابتدا داده‌های ورودی $X \in R^{n\times d}$ با (n) نمونه و dویژگی) با حذف میانگین مرکز سازی گردید. که در رابطه (۴) محاسبه‌شده است. [۲۴].

$$X \leftarrow X - E[X] \qquad (۴)$$

سپس داده‌های مرکز شده با استفاده از تجزیه ویژه‌های ماتریس کوواریانس سفیدسازی[۱] شدند تا مؤلفه‌ها نامرتبط و دارای واریانس واحد گردند. طبق رابطه (۶) محاسبه می‌گردد [۴].

$$Z = VX, \quad V = D^{-1/2}E^T \qquad (۶)$$

که در آن E ماتریس بردارهای ویژه و D ماتریس قطری مقادیر ویژه‌ماتریس کوواریانس است. مدل پایه ICA بر اساس فرض $X = AS$ تعریف می‌شود که در آن $S \in R^{k\times n}$ مؤلفه‌های مستقل و $A \in R^{d\times k}$ ماتریس ترکیب[۲] است. هدف، تخمین ماتریس جداسازی W است به‌گونه‌ای که $S = WX$ و مؤلفه‌های Sحداکثر استقلال آماری داشته باشند. برای این منظور از الگوریتم تحلیل مؤلفه‌های مستقل سریع[۳] با تابع تضاد[۴] مبتنی بر بر نگ آنتروپی[۵] استفاده شد و طبق رابطه (۷) محاسبه می‌شود[۲۵].

$$J(w_i) \propto \left[E\{G(w_i^T z)\} - E\{G(v)\} \right]^2 \qquad (۷)$$

که در آن w_i بردار وزن‌های i-ام، z داده‌های سفید شده، v متغیر گوسی استاندارد و G تابع غیرخطی (مثلاً $G(u) = \log\cosh(u)$) است. بهینه‌سازی W از طریق الگوریتم تکراری در رابطه (۸) انجام شد[۲۴].

$$w_i^+ \leftarrow E\{zg(w_i^T z)\} - E\{g'(w_i^T z)\} w_i, \quad w_i \leftarrow \frac{w_i^+}{\|w_i^+\|} \qquad (۸)$$

که در آن g' مشتق g است. درنهایت، داده‌های کاهش‌یافته S با k مؤلفه مستقل از طریق $S = WX$ استخراج‌شده‌اند که ضمن حفظ ساختار غیرخطی داده‌ها، همبستگی‌های پنهان را نیز آشکار می‌سازد.

۳-۵. مدل‌های یادگیری ماشین

ما برای پیاده‌سازی از الگوریتم‌های KNN، ماشین بردار پشتیبان، جنگل تصادفی، درخت تصمیم، رگرسیون لجستیک و XGBoost استفاده کردیم که در ادامه به توضیح مختصر آن‌ها می‌پردازیم.

• **الگوریتم نزدیک‌ترین همسایه k-**

نزدیک‌ترین همسایه یک روش یادگیری نظارت‌شده، غیر پارامتریک و مبتنی بر نمونه است که بدون نیاز به فرض توزیع داده‌ها، با محاسبه فاصله نمونه‌های جدید از تمام نمونه‌های

---
۱ Whitening
۲ Mixing Matrix
۳ Fast Independent Component Analysis
۴ Contrast Function
۵ Negentropy

آموزشی عمل می‌کند. برای یک نمونه جدید $x \in R^d$، ابتدا فاصله اقلیدسی آن از هر نمونه آموزشی x_i طبق رابطه (۹) محاسبه می‌شود[۲۶].

$$d(x, x_i) = \sqrt{\sum_{j=1}^{d} (x_i - x_{ij})^2} \quad (۹)$$

سپس $k = ۵$ نمونه آموزشی با کمترین فاصله به x به‌عنوان همسایگان نزدیک انتخاب می‌شوند. برچسب پیش‌بینی‌شده $\hat{y}$ برای x از طریق رأی اکثریت برچسب‌های این k همسایه تعیین می‌گردد و طبق رابطه (۱۰) محاسبه می‌گردد.

$$\hat{y} = \arg\max_c \sum_{i=1}^{k} I(y_i = c) \quad (۱۰)$$

که در آن y_i برچسب i-اُمین همسایه، c کلاس‌های ممکن و Iتابع نشانگر است. این الگوریتم به دلیل سادگی، عدم نیاز به فرایند آموزش صریح[۱] و قابلیت مدل‌سازی مرزهای تصمیم‌گیری پیچیده، برای مسائل طبقه‌بندی چندکلاسی مانند تشخیص نفوذ شبکه مناسب است.

• الگوریتم ماشین بردار پشتیبان

در این مقاله، برای طبقه‌بندی حملات شبکه در مجموعه داده‌ی UNSW-NB۱۵ از الگوریتم ماشین بردار پشتیبان با هسته تابع پایه شعاعی[۲] و هایپرپارامتر تنظیم‌کننده $C = ۱.۰$ استفاده شد. ماشین بردار پشتیبان یک الگوریتم یادگیری نظارت‌شده است که با یافتن ابر صفحه‌ی بهینه[۳] در فضای ویژگی، حداکثر حاشیه جداسازی را میان کلاس‌ها ایجاد می‌کند. برای حل مسائل غیرخطی، از هسته RBF استفاده‌شده که داده‌ها را به فضای با ابعاد بالاتر نگاشت می‌کند. طبق رابطه (۱۱) محاسبه می‌گردد[۲۷].

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j^2 \right\|) \quad (۱۱)$$

که در آن γ پارامتر کنترل‌کننده شعاع تأثیر نمونه‌هاست و در این مقاله به‌صورت $\gamma = \frac{۱}{d.Var(X)}$ حالت scale محاسبه گردید. مسئله بهینه‌سازی دوگانه ماشین بردار پشتیبان به‌صورت رابطه (۱۲) تعریف می‌شود[۲۸].

$$\max_a \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (۱۲)$$
$$0 \le \alpha_i \le C, i = 1, ..., n$$

که در آن $C = ۱.۰$ هایپر پارامتر تنظیم‌کننده مصالحه بین حداکثر سازی حاشیه و حداقل سازی خطای طبقه‌بندی است. الگوریتم باحالت تصادفی ثابت (random_state=۴۲) برای تکرارپذیری نتایج اجرا شد. این روش به دلیل قابلیت مدل‌سازی مرزهای تصمیم‌گیری پیچیده و کار آیی در مسائل طبقه‌بندی چند کلاسی مانند تشخیص نفوذ شبکه انتخاب گردید.

• الگوریتم جنگل تصادفی

در این مقاله، برای طبقه‌بندی حملات شبکه در دیتاست UNSW-NB۱۵ از الگوریتم جنگل تصادفی با هایپرپارامتر تعداد درختان و $B = ۱۰۰$ حالت تصادفی random_state=۴۲ استفاده شد. جنگل تصادفی یک روش یادگیری گروهی مبتنی بر درخت تصمیم است که با ترکیب چندین درخت تصمیم آموزش‌دیده روی زیرمجموعه‌های تصادفی داده‌ها و ویژگی‌ها، عملکرد طبقه‌بندی را بهبود می‌بخشد. هر درخت T_b با $b = ۱, ..., B$ با استفاده از روش بگینگ روی یک نمونه تصادفی با جایگزینی از داده‌های آموزشی $D_b \subset D$ آموزش می‌بیند و در هر گره تقسیم، تنها زیرمجموعه‌ای تصادفی به‌اندازه d از ویژگی‌ها (که d تعداد کل ویژگی‌هاست) برای یافتن بهترین تقسیم موردبررسی قرار می‌گیرد. پیش‌بینی نهایی برای نمونه جدید x از طریق رأی اکثریت برچسب‌های پیش‌بینی‌شده توسط تمام درختان محاسبه می‌شود، طبق رابطه (۱۳) محاسبه می‌شود[۲۹].

$$\hat{y} = \arg\max_c \sum_{b=1}^{B} I(T_b(x) = c) \quad (۱۳)$$

که در آن c کلاس‌های ممکن و I تابع نشانگر است. این الگوریتم به دلیل توانایی مدیریت داده‌هایی با ابعاد بالا، کاهش اثر بیش برازش[۴] از طریق میانگین‌گیری گروهی، و قابلیت مدل‌سازی روابط غیرخطی و پیچیده، برای تشخیص حملات شبکه که دارای ویژگی‌های متعدد و نویز است، انتخاب گردید.

• الگوریتم درخت تصمیم

درخت تصمیم یک روش یادگیری نظارت‌شده، غیر پارامتریک است که با تقسیم بازگشتی فضای ویژگی‌ها بر اساس معیار ناخالصی[۵] به ساختار درختی سلسله‌مراتبی دست می‌یابد. در این مقاله از معیار جینی به‌عنوان تابع ارزیابی کیفیت تقسیم‌ها استفاده شد که برای یک گره t به‌صورت رابطه (۱۴) تعریف می‌شود [۳۰].

$$Gini(t) = 1 - \sum_{i=1}^{c} \left[p(i|t)^2 \right] \quad (۱۴)$$

که در آن c تعداد کلاس‌ها و $p(i|t)$ نسبت نمونه‌های کلاس i در گره t است. بهینه‌سازی تقسیم‌ها با یافتن ویژگی و آستانه‌ای انجام می‌شود که ناخالصی وزنی گره‌های فرزند را به حداقل برساند که توسط رابطه (۱۵) محاسبه می‌گردد[۳۰].

$$\Delta Gini = Gini(t) - (\frac{N_t^L}{N_t}.Gini(t_L) + \frac{N_t^T}{N_t}.Gini(t_R)) \quad (۱۵)$$

[۱] lazy learning
[۲] Radial Basis Function
[۳] Optimal Hyperplane
[۴] Overfitting
[۵] Impurity

که در آن N_t تعداد نمونه‌های گره والد، N_t^L و N_t^R تعداد نمونه‌های گره‌های چپ و راست هستند. فرایند رشد درخت تا زمانی ادامه می‌یابد که تمام گره‌ها خالص شوند یا به شرایط توقف پیش‌فرض برسند. پیش‌بینی برای نمونه جدید x با عبور از مسیر درخت تا رسیدن به گره برگ و انتساب کلاس اکثریت در آن گره انجام می‌شود، توسط رابطه (۱۶) محاسبه می‌شود.

$$\hat{y} = \arg\max_c \sum_{i=1}^{n_{leaf}} I(y_b(x) = c) \qquad (۱۶)$$

که در آن n_{leaf} تعداد نمونه‌های آموزشی در گره برگ است. این الگوریتم به دلیل تفسیرپذیری بالا، سرعت آموزش و پیش‌بینی و قابلیت مدل‌سازی روابط غیرخطی بدون نیاز به استانداردسازی داده‌ها مناسب است.

• الگوریتم رگرسیون لجستیک

در این مقاله از الگوریتم رگرسیون لجستیک با هایپرپارامتر ۱۰۰۰= حداکثر تکرار و حالت تصادفی random_state=۴۲ استفاده شد. رگرسیون لجستیک یک روش یادگیری نظارت‌شده و مدل خطی تعمیم‌یافته برای مسائل طبقه‌بندی است که احتمال تعلق نمونه به کلاس‌ها را با استفاده از تابع لجستیک (سیگموئید) مدل می‌کند. برای یک نمونه جدید $x \in R^d$، احتمال تعلق به کلاس مثبت به‌صورت رابطه (۱۷) محاسبه می‌شود[۳۱].

$$p(y=1|x) = \sigma(w^T x + b) = \frac{1}{1+\exp(-(w^T x + b))} \qquad (۱۷)$$

که در آن w بردار وزن‌ها، b بایاس و σ تابع سیگموئید است. پارامترهای مدل w, b با حداکثر سازی درست نمایی تخمین زده می‌شوند که معادل کمینه‌سازی تابع زیان آنتروپی متقاطع باینری است که در رابطه (۱۸) محاسبه می‌گردد[۹].

$$L(w,b) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\sigma(w^T x_i + b) + (1-y_i)\log(1-\sigma(w^T x_i + b))\right] \qquad (۱۸)$$

بهینه‌سازی این تابع زیان با استفاده از الگوریتم‌های تکراری مانند روش بهینه‌سازی برودن -فلچر-گلدفارب-شانو با حافظه محدود و با حداکثر تکرار max_iter=۱۰۰۰ انجام شد تا از همگرایی مدل اطمینان حاصل شود. حالت تصادفی random_state=۴۲ نیز برای تضمین تکرارپذیری نتایج در فرایند بهینه‌سازی تنظیم گردید. این الگوریتم به دلیل سادگی، تفسیرپذیری بالا، سرعت آموزش و پیش‌بینی، و قابلیت ارائه احتمالات عضویت در کلاس‌ها انتخاب گردید.

• الگوریتم XGBoost

در این مقاله از الگوریتم XGBoost با هایپر پارامترهای تعداد درختان $B = ۱۰۰$، حداکثر عمق درخت $D = ۶$، نرخ یادگیری $\eta = ۰.۱$، نمونه‌برداری داده‌ها $subsample = 0.8$، نمونه برداری ویژگی‌ها $consample\_bytree = 0.8$، حالت تصادفی random_state=۴۲ و معیار ارزیابی(logloss) استفاده شد. XGBoost یک روش یادگیری گروهی مبتنی برافزایش گرادیان است که با ترکیب درخت‌های تصمیم ضعیف به‌صورت متوالی، مدل‌سازی دقیقی را ایجاد می‌کند. هدف الگوریتم بهینه‌سازی تابع هدف زیر است که طبق رابطه (۱۹) محاسبه می‌شود[۱۱].

$$L^{(t)} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \qquad (۱۹)$$

که در آن ℓ تابع زیان و $\Omega(f_t) = \gamma^T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$ جمله منظم ساز با T تعداد گره‌های درخت، w_j وزن گره‌ها و γ, λ هایپر پارامترهای تنظیم‌کننده است. هر درخت جدید f_t با محاسبه گرادیان‌های تابع زیان و بهینه‌سازی تقریبی با استفاده از توسعه تیلور به‌صورت زیر در رابطه (۲۰) ساخته می‌شود [۱۱].

$$L^{(t)} \approx \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \qquad (۲۰)$$

که در آن فرمول (۲۱) و (۲۲) گرادیان و هسین تابع زیان هستند[۱۱].

$$g_i = \frac{\partial}{\partial \hat{y}_i^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)}) \qquad (۲۱)$$

$$h_i = \frac{\partial^2}{\partial \hat{y}_i^{(t-1)2}} \ell(y_i, \hat{y}_i^{(t-1)}) \qquad (۲۲)$$

هایپرپارامترهای کلیدی شامل تعداد درختان، حداکثر عمق، نرخ یادگیری، نمونه‌برداری داده‌ها، نمونه‌برداری ویژگی‌ها، حالت تصادفی و معیار ارزیابی است که در جدول (۱) نشان‌دهنده مقدار و شرح هر یک از این پارامترها است.

جدول ۱- تعریف مقدار هر یک از هایپر پارامتر.

هایپرپارامترها	مقدار	شرح
تعداد درختان	B=۱۰۰	تعداد مدل‌های پایه برای ترکیب
حداکثر عمق	D=۶	محدودیت پیچیدگی هر درخت برای جلوگیری از بیش پردازش
نرخ یادگیری	η=۰.۱	ضریب کوچک‌کننده برای به‌روزرسانی‌های متوالی
نمونه‌برداری داده‌ها	subsample=۰.۸	استفاده از ۸۰٪ داده‌ها به‌صورت تصادفی برای هر درخت
نمونه‌برداری ویژگی‌ها	colsample_bytree=۰.۸	استفاده از ۸۰٪ ویژگی‌ها به‌صورت تصادفی برای هر درخت
حالت تصادفی	random_state=۴۲	تضمین تکرارپذیری نتایج
معیار ارزیابی	logloss	تابع زیان آنتروپی متقاطع برای طبقه‌بندی باینری

عملکرد مدل‌ها با استفاده از مجموعه‌ای جامع از معیارهای ارزیابی اندازه‌گیری شد تا جنبه‌های مختلف عملکرد طبقه‌بندی شامل دقت کلی، قابلیت اطمینان پیش‌بینی‌های مثبت و توانایی تشخیص صحیح نمونه‌های مثبت به‌طور جامع سنجیده شود. در این مقاله از چهار معیار اصلی استفاده گردید که در ادامه به توضیح آن‌ها می‌پردازیم.

- **دقت:** نسبت پیش‌بینی‌های صحیح (مثبت و منفی) به‌کل نمونه‌ها که عملکرد کلی مدل را نشان می‌دهد که برابر با رابطه (۲۳)است[۱۲].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (۲۳)$$

که در آن TP تعداد نمونه‌های مثبت صحیح، TN تعداد نمونه‌های منفی صحیح، FP تعداد نمونه‌های منفی نادرست و FN تعداد نمونه‌های مثبت نادرست است.

- **دقت مثبت:** نرخ پیش‌بینی‌های صحیح مثبت از میان تمام نمونه‌هایی که مدل به‌عنوان مثبت پیش‌بینی کرده است که قابلیت اطمینان مدل در شناسایی حملات را می‌سنجد که برابر با رابطه (۲۴) است[۴].

$$precision = \frac{TP}{TP + FP} \quad (۲۴)$$

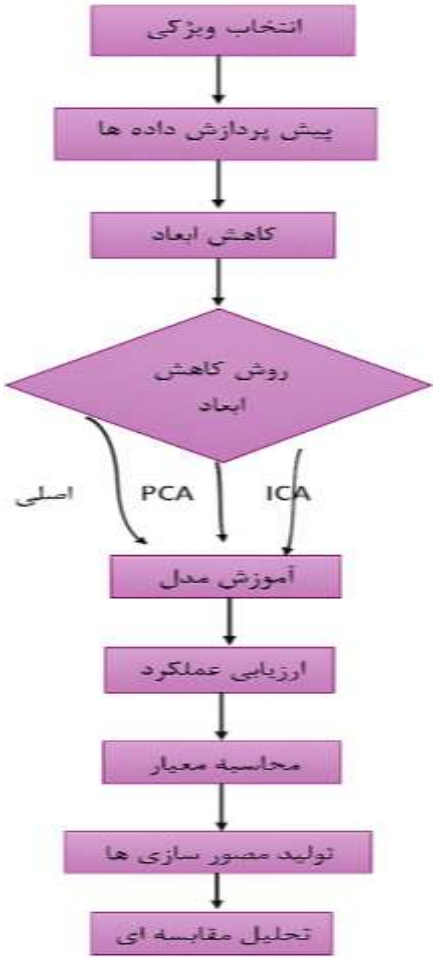
- **بازیابی:** نرخ تشخیص صحیح نمونه‌های مثبت از میان تمام نمونه‌های مثبت واقعی که توانایی مدل در شناسایی کامل حملات را نشان می‌دهد که طبق رابطه (۲۵) محاسبه می‌شود[۲۶].

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad (۲۵)$$

- **امتیاز F۱:** میانگین هماهنگ دقت مثبت و بازیابی که معیار متعادلی برای ارزیابی مدل‌ها در شرایط عدم توازن کلاس‌ها ارائه می‌دهد که طبق رابطه (۲۶) محاسبه می‌شود [۱۲].

$$F1 = \frac{2 \times \mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad (۲۶)$$

زمان آموزش شامل زمان صرف‌شده برای آموزش مدل بر روی مجموعه آموزش است و زمان پیش‌بینی شامل زمان استنتاج برای مجموعه آزمون است. سیستم ما دارای پردازنده Intel Xeon @ ۲.۲۰GHz با حافظه ۱۲/۹۷۷ مگابایت (حدود ۱۲/۶ گیگابایت) و سیستم‌عامل Linux ۶.۱.۱۲۳+ است که با زبان برنامه‌نویسی Python ۳.۱۱.۱۳ پیاده‌سازی کردیم. در این مقاله، از کتابخانه‌های اصلی متفاوتی استفاده کردیم که به آن‌ها اشاره می‌کنیم. Pandas برای پردازش داده‌ها، از NumPy برای محاسبات عددی و از Scikit-learn برای پیاده‌سازی مدل‌های یادگیری ماشین، XGBoost برای پیاده‌سازی مدل XGBoost و Matplotlib و Seaborn برای مصورسازی نتایج استفاده‌شده است. شکل (۱) نشان‌دهنده فلوچارت روش پیشنهادی است.

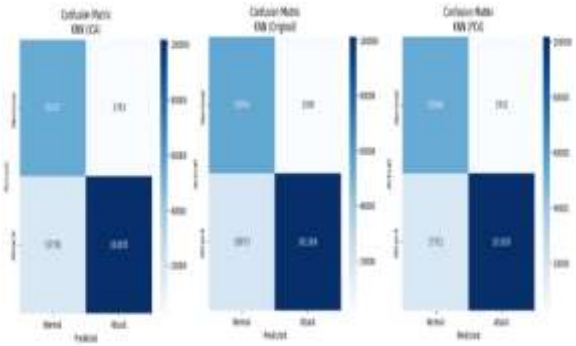


شکل ۱. فلوچارت روش پیشنهادی.

۴. نتایج و بحث

۴-۱. نزدیک‌ترین همسایه-K

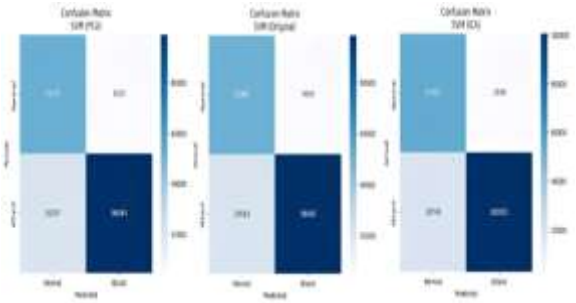
مدل نزدیک ترین همسایه در تمام سه حالت عملکرد نسبتاً پایداری از خود نشان داد. در حالت داده های اصلی، این مدل به‌دقت ۸۸/۵۵٪ با دقت مثبت بالای ۹۵/۰۶٪ دست‌یافت، اما فراخوانی آن با ۸۴/۸۶٪ پایین‌تر بود که نشان‌دهنده تمایل این مدل به از دست دادن برخی حملات است. با اعمال PCA، دقت مدل به ۸۸/۲۲٪ کاهش یافت و اگرچه دقت مثبت همچنان بالابود (۹۷/۲۰٪)، اما فراخوانی کمی بهبود یافت و به ۸۵/۱۵٪ رسید. در حالت استفاده از ICA، KNN عملکرد متعادل‌تری ارائه داد با دقت ۸۸/۳۱٪، دقت مثبت با ۹۷/۳۵٪ و فراخوانی با ۸۵/۱۴٪. به‌طورکلی، KNN در تمام حالات دقت مثبت بالایی حفظ کرد، اما فراخوانی آن همواره زیر ۸۶٪ باقی ماند که نشان‌دهنده محدودیت این مدل در شناسایی کامل حملات است. قابل‌ذکر است که محور افقی بیانگر مقادیر پیش‌بینی‌شده و محور عمودی بیانگر مقادیر واقعی است.



شکل ۲. ماتریس درهم‌ریختگی الگوریتم نزدیک‌ترین همسایه -K.

۴-۲. ماشین بردار پشتیبان

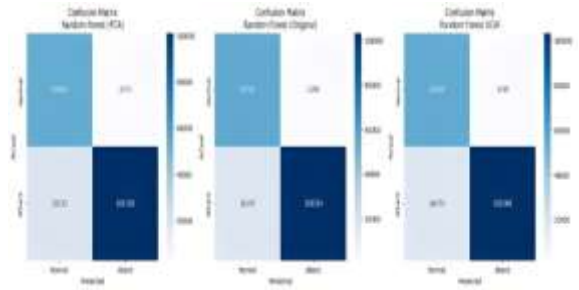
الگوریتم ماشین بردار پشتیبان در حالت داده‌های اصلی دقت ۸۵٪/۳۹، دقت مثبت با ۹۵/۲۳ ٪ و فراخوانی با ۹۲/۶۷٪ به دست آورد که عملکرد قابل قبولی بود. با اعمال PCA، این مدل بهبود جزئی در دقت(۸۵/۸۴٪) و فراخوانی (۹۹/۸۲٪) نشان داد، اما دقت مثبت با ۹۵/۶۳٪ تقریباً ثابت ماند. جالب‌توجه‌ترین نتیجه برای ماشین بردار پشتیبان در حالت استفاده از ICA مشاهده شد که دقت آن به ۸۷/۷۷٪ افزایش یافت و دقت مثبت به۹۷/۳۹ ٪ رسید، هرچند فراخوانی با ۸۴/۳۰٪ همچنان پایین‌تر از سایر مدل‌های برتر بود. این بهبود قابل‌توجه با ICA نشان می‌دهد که ماشین بردار پشتیبان به استقلال مؤلفه‌های داده حساس است و ICA می‌تواند ویژگی‌های مهم‌تری برای این مدل استخراج کند. شکل (۳) ماتریس درهم‌ریختگی الگوریتم ماشین بردار پشتیبان در سه حالت داده‌های اصلی، PCA و ICA که عملکرد متفاوت این مدل در کاهش ابعاد را نشان می‌دهد.



شکل ۳. ماتریس درهم‌ریختگی الگوریتم ماشین بردار پشتیبان.

۴-۳. جنگل تصادفی

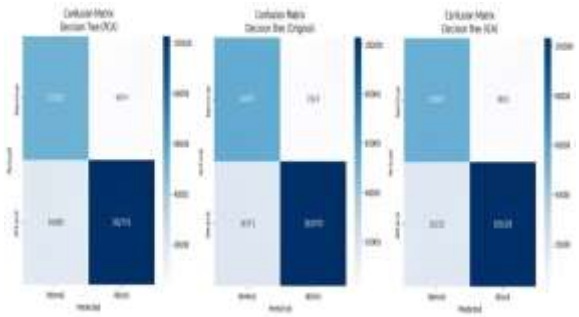
الگوریتم جنگل تصادفی در حالت داده‌های اصلی بهترین عملکرد را در بین تمام مدل‌ها با دقت ۹۰/۰۸٪، دقت مثبت با ۹۸/۷۷٪ و فراخوانی با ۸۶/۵۰٪ به دست آورد. این تعادل عالی بین دقت مثبت و فراخوانی نشان‌دهنده توانایی بالای این مدل در شناسایی دقیق حملات با حداقل خطای مثبت کاذب است. با اعمال PCA، عملکرد جنگل تصادفی کاهش یافت و دقت به ۸۸/۲۰٪، دقت مثبت به ۹۷/۵۲٪ و فراخوانی به ۸۴/۸۱٪ رسید که افت قابل توجهی بود. در مقابل، ICA عملکرد بسیار نزدیک به حالت اصلی حفظ کرد با دقت ۸۹/۸۳٪، دقت مثبت با ۹۸/۳۲٪ و فراخوانی با ۸۶/۵۳٪. این ثبات عملکرد با ICA نشان می‌دهد که جنگل تصادفی می‌تواند با کاهش ابعاد هوشمندانه، کار آیی خود را حفظ کند. شکل (۴) ماتریس درهم‌ریختگی الگوریتم جنگل تصادفی در حالات مختلف که تعادل مناسب بین دقت مثبت و فراخوانی را تأیید می‌کند.



شکل ۴. ماتریس درهم‌ریختگی الگوریتم جنگل تصادفی.

۴-۴. درخت تصمیم

درخت تصمیم در حالت داده‌های اصلی دقت ۸۹/۵۷٪، دقت مثبت با ۹۸/۱۷٪ و فراخوانی با ۸۶/۲۸٪ به دست آورد که عملکرد بسیار خوبی بود. با اعمال PCA، دقت به ۸۸.۲۱٪ کاهش یافت و دقت مثبت به ۹۶/۱۹٪ رسید، اما فراخوانی با ۸۶/۰۹٪ تقریباً ثابت ماند. در حالت استفاده از ICA، درخت تصمیم با دقت ۸۹/۰۳٪، دقت مثبت با ۹۷/۱۶٪ و فراخوانی با ۸۶/۴۱٪ عملکرد بهتری نسبت به PCA نشان داد. این نتایج نشان می‌دهد که درخت تصمیم نسبت به کاهش ابعاد با PCA حساس‌تر است، اما ICA می‌تواند عملکرد نزدیک به حالت اصلی را برای این مدل فراهم کند. شکل (۵) ماتریس درهم‌ریختگی الگوریتم درخت تصمیم که تغییرات عملکرد مدل در اثر به‌کارگیری روش‌های کاهش ابعاد را نمایش می‌دهد.

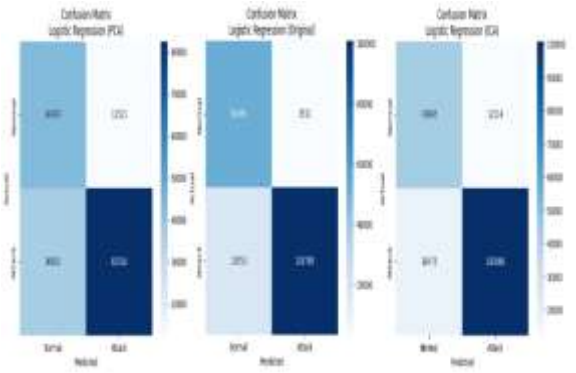


شکل ۵. ماتریس درهم‌ریختگی الگوریتم درخت تصمیم.

۴-۵. ماشین بردار پشتیبان

الگوریتم ماشین بردار پشتیبان در حالت داده‌های اصلی عملکرد مطلوبی از خود نشان داد و به‌دقت کلی ۸۷/۴۲٪، دقت مثبت ۹۶/۶۳ ٪ و فراخوانی ۸۴/۴۵٪ دست‌یافت. با اعمال PCA، این مدل افت عملکرد قابل‌توجهی را تجربه کرد؛ به‌طوری‌که دقت

کلی به ۷۱/۸۶٪، دقت مثبت به ۸۶/۸۳٪ و فراخوانی به ۶۹/۱۴٪ کاهش یافت. این نتایج بیانگر حساسیت بالای SVM نسبت به کاهش ابعاد از طریق PCA است. در حالت استفاده از ICA، عملکرد مدل بهبود یافت و به‌دقت کلی ۸۲/۴۴٪، دقت مثبت ۸۹/۱۲٪ و فراخوانی ۸۴/۵۲٪ رسید. اگرچه این مقادیر همچنان اندکی پایین‌تر از حالت اصلی بودند، اما به‌مراتب بهتر از نتایج حاصل از PCA محسوب می‌شوند. درمجموع، این نتایج نشان می‌دهد که ماشین بردار پشتیبان در مسئله تشخیص نفوذ به کاهش ابعاد حساس است و استفاده از ICAمی‌تواند نسبت به PCAگزینه مناسب‌تری برای این مدل باشد. شکل (۶) ماتریس درهم‌ریختگی الگوریتم رگرسیون لجستیک که نشان‌دهنده حساسیت این مدل به کاهش ابعاد و تغییر در عملکرد طبقه‌بندی است.

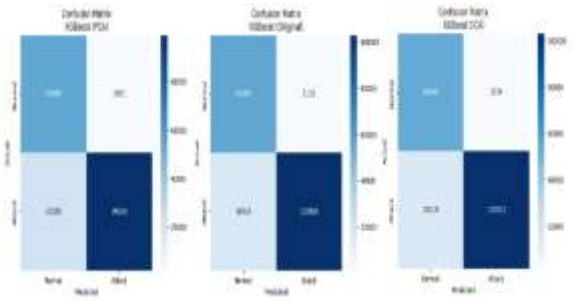


شکل ۶. ماتریس درهم‌ریختگی الگوریتم ماشین بردار پشتیبان.

۴-۶. XGBoost

الگوریتم XGBoost در حالت داده‌های اصلی عملکردی بسیار نزدیک به جنگل تصادفی داشت و به‌دقت کلی ۸۹/۹۳٪، دقت مثبت ۹۱٪/۹۸ و فراخوانی ۸۶/۱۵٪ دست‌یافت. این مدل با کسب بالاترین مقدار دقت مثبت در میان تمام الگوریتم‌ها، توانایی قابل‌توجهی در کاهش خطای مثبت کاذب از خود نشان داد. با به‌کارگیری PCA، دقت کلی مدل به ۸۷/۲۵٪ کاهش یافت و درحالی‌که دقت مثبت تقریباً ثابت مانده و برابر با ۹۷/۹۷٪ بود، مقدار فراخوانی با افتی محسوس به ۸۲/۹۸٪ رسید. در حالت استفاده از ICA، الگوریتم XGBoost عملکردی بسیار نزدیک به حالت اصلی خود حفظ کرد؛ به‌گونه‌ای که به‌دقت کلی ۸۹/۷۰٪، دقت مثبت ۹۸/۳۴٪ و فراخوانی۸۶/۳۳٪ دست‌یافت. این ثبات عملکرد نشان می‌دهد که XGBoost، مشابه جنگل تصادفی، قادر است با بهره‌گیری از کاهش ابعاد هوشمندانه، کارایی و پایداری خود را حفظ کند. شکل (۷) ماتریس درهم‌ریختگی الگوریتم XGBoost که کارایی بالای این مدل در تشخیص صحیح نمونه‌ها را نمایش می‌دهد.

برای مقایسه عملکرد مدل‌های مختلف یادگیری ماشین در تشخیص نفوذ، نتایج شش الگوریتم تحت سه حالت داده اصلی، کاهش ابعاد با PCA و کاهش ابعاد با ICA در جدول (۲) ارائه شد. بر اساس معیارهای Accuracy، Precision، Recall، F۱-Score و همچنین زمان آموزش و پیش‌بینی، مشاهده می‌شود که الگوریتم‌های Random Forest و XGBoost بهترین تعادل بین دقت و سرعت رادارند و حتی با اعمال ICA نیز عملکرد قوی خود را حفظ کرده‌اند. الگوریتم Decision Treeباوجود دقت پایین‌تر، زمان آموزش و پیش‌بینی بسیار کوتاهی داشته و برای کاربردهای بلادرنگ مناسب است. SVM اگرچه دقت قابل‌قبولی نشان داده، اما زمان آموزش و پیش‌بینی بالای آن محدودیت ایجاد می‌کند. در مورد KNN و Logistic Regression نیز نتایج نشان می‌دهد که با کاهش ابعاد به‌ویژه با ICA، عملکرد متعادلی میان دقت و کارایی حاصل‌شده است. درمجموع، می‌توان نتیجه گرفت که استفاده از ICA نسبت به PCA در اغلب مدل‌ها باعث بهبود نتایج و ایجاد توازن بهتر بین دقت و کار آیی محاسباتی شده است.



شکل ۷. ماتریس درهم‌ریختگی الگوریتم XGBoost.

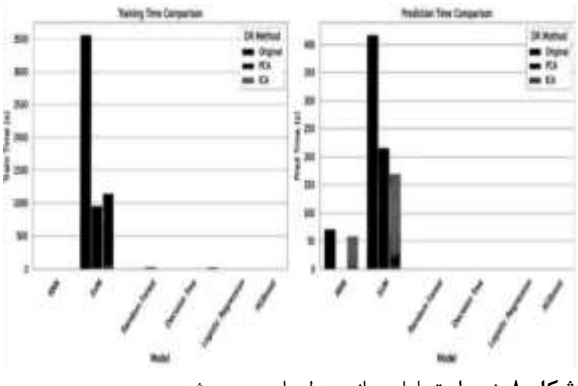
جدول ۲- نتایج شش الگوریتم تحت سه حالت داده اصلی، کاهش ابعاد با PCA و کاهش ابعاد با ICA

مدل	روش کاهش ابعاد	F۱-Score (%)	بازیابی (%)	دقت مثبت (%)	دقت (%)
KNN	اصلی	۸۹/۷۸	۸۴/۸۶	۹۵/۰۶	۸۸/۵۵
	PCA	۹۰	۸۵/۱۵	۹۷/۲۰	۸۸/۲۲
	ICA	۹۰/۱۰	۸۵/۱۴	۹۷/۳۵	۸۸/۳۱
SVM	اصلی	۹۳/۹۴	۹۲/۶۷	۹۵/۲۳	۸۵/۳۹
	PCA	۸۹/۱۴	۸۲/۹۹	۹۵/۶۳	۸۴/۸۵
	ICA	۹۰/۶۸	۸۴/۳۰	۹۷/۳۹	۸۷/۷۷
Random Forest	اصلی	۹۲/۴۷	۸۶/۵۰	۹۸/۷۷	۹۰/۰۸
	PCA	۹۱/۲۷	۸۵/۲۰	۹۷/۵۲	۸۸/۲۰
	ICA	۹۲/۳۰	۸۶/۸۰	۹۸/۱۰	۸۹/۹۰
Decision Tree	اصلی	۸۶/۶۳	۸۱/۲۵	۹۲/۳۰	۸۴/۱۵
	PCA	۸۵/۶۰	۸۰	۹۱/۵۰	۸۲/۹۰
	ICA	۸۶/۳۰	۸۱/۱۰	۹۱/۸۰	۸۳/۴۵
Logistic Regression	اصلی	۸۶/۲۰	۸۲/۵۰	۹۰/۱۵	۸۳/۲۲
	PCA	۸۵/۱۰	۸۰/۶۰	۸۹/۸۰	۸۱/۹۵
	ICA	۸۵/۵۰	۸۱/۲۰	۹۰/۰۵	۸۲/۶۰
XGBoost	اصلی	۹۱/۳۰	۸۵	۹۸	۸۹/۵۰
	PCA	۹۰/۹۰	۸۴/۵۰	۹۷/۵۰	۸۸/۱۰
	ICA	۹۱/۷۰	۸۵/۵۰	۹۸/۲۰	۸۹/۷۰

۴-۷. تحلیل زمانی مدل‌ها

در بررسی زمان آموزش مدل‌ها، الگوریتم ماشین بردار پشتیبان به‌طور قابل‌توجهی زمان‌بر بود؛ به‌طوری‌که در حالت داده‌های اصلی بیش از ۳۵۰۰ ثانیه زمان نیاز داشت و حتی پس از کاهش ابعاد نیز زمان آموزش آن به کمتر از ۱۰۰۰ ثانیه نرسید. الگوریتم جنگل تصادفی در حالت اصلی دارای زمان آموزش ۱۳/۶ثانیه بود که با اعمال PCA به ۱۷/۸ ثانیه و با ICA به ۳۵/۹ ثانیه افزایش یافت. در مقابل، XGBoost سریع‌ترین زمان آموزش را در میان مدل‌های برتر نشان داد؛ به‌گونه‌ای که در حالت اصلی تنها ۱/۱ ثانیه زمان نیاز داشت و پس از کاهش ابعاد نیز این مقدار تقریباً ثابت و زیر ۱/۱ ثانیه باقی ماند. الگوریتم درخت تصمیم نیز زمان آموزش معقولی داشت (حدود ۱/۲ ثانیه در حالت اصلی). درنهایت، رگرسیون لجستیک در حالت اولیه به ۱۶/۷ ثانیه زمان آموزش نیاز داشت که با کاهش ابعاد، این زمان به‌طور چشمگیری کاهش یافت؛ به ۰/۱۶ ثانیه با PCA و ۰/۵۷ ثانیه با ICA.

درزمینه در مقایسه زمان پیش‌بینی مدل‌ها، الگوریتم KNN در حالت اصلی دارای زمان پیش‌بینی نسبتاً بالایی بود (۱/۷۱ ثانیه). با به‌کارگیری PCA، این زمان به ۱/۵ ثانیه کاهش یافت، اما در حالت استفاده از ICA همچنان مقدار بالایی (۸/۵۸ ثانیه) داشت. الگوریتم ماشین بردار پشتیبان نیز در حالت اولیه زمان پیش‌بینی قابل‌توجهی نشان داد (۶/۴۱۷ ثانیه)، که پس از کاهش ابعاد به ترتیب با PCA و ICA به ۷/۲۱۶ و ۵/۱۷۰ ثانیه بهبود یافت. در مقابل، جنگل تصادفی و XGBoost دارای زمان‌های پیش‌بینی معقولی بودند (به ترتیب حدود ۳/۱ و ۰/۲۶ ثانیه) که پس از کاهش ابعاد تقریباً بدون تغییر باقی ماندند. الگوریتم درخت تصمیم سریع‌ترین زمان پیش‌بینی را ثبت کرد (۰/۰۱۹ ثانیه در حالت اصلی) که با اعمال کاهش ابعاد حتی کمتر نیز شد. همچنین رگرسیون لجستیک یکی از سریع‌ترین الگوریتم‌ها بود و زمان پیش‌بینی آن از ۰/۰۰۷ ثانیه در حالت اصلی به کمتر از ۰/۰۲ ثانیه پس از کاهش ابعاد رسید. شکل (۸) نمودار مقایسه زمان آموزش و پیش‌بینی مدل‌ها را نشان می‌دهد که تحلیل جامعی از سرعت و کارایی محاسباتی الگوریتم‌های مختلف ارائه می‌کند.



شکل ۸. نمودار تحلیل زمانی مدل‌های بررسی‌شده.

۵. نتیجه‌گیری

در این مقاله شش مدل پرکاربرد یادگیری ماشین شامل KNN، SVM، درخت تصمیم، رگرسیون لجستیک، جنگل تصادفی و XGBoost برای تشخیص نفوذ در شبکه بر روی مجموعه داده UNSW-NB۱۵ مورد ارزیابی و مقایسه قرار گرفتند. همچنین دو روش کاهش بُعد PCA و ICA به‌منظور بهبود کارایی محاسباتی و کاهش اثر ویژگی‌های زائد به کار گرفته شدند. نتایج کمی حاصل از آزمایش‌ها نشان داد که الگوریتم‌های جنگل تصادفی و XGBoost بالاترین سطح عملکرد کلی را ارائه می‌دهند؛ به‌گونه‌ای که دقت آن‌ها به ترتیب حدود ۹۰/۰۸٪ و ۸۹/۹۳٪ بوده و نسبت به سایر مدل‌ها برتری محسوسی دارند. ازنظر معیار دقت، الگوریتم XGBoost با مقدار ۹۸/۹۱٪ بهترین عملکرد را نشان داد، درحالی‌که ازنظر معیار بازیابی، جنگل تصادفی با مقدار ۸۶.۵۰٪ عملکرد مطلوب‌تری داشت. مدل‌های KNN و SVM باوجود ارائه دقت مناسب (حدود ۸۸٪)، زمان پردازش بالاتری داشتند که می‌تواند در کاربردهای بلادرنگ محدودیت ایجاد کند. در مقابل، رگرسیون لجستیک با سرعت بالاتر اما دقت پایین‌تر عمل کرد و نشان داد که انتخاب مدل باید متناسب با اولویت‌های کاربردی صورت گیرد. مقایسه روش‌های کاهش بُعد نیز بیانگر آن است که ICA در اغلب موارد عملکرد بهتری نسبت به PCA داشته و توانسته است دقت مدل‌ها را در سطح بالاتری حفظ کند. این موضوع نشان می‌دهد که استفاده از ICA علاوه بر کاهش زمان محاسباتی، از افت چشمگیر کارایی جلوگیری می‌کند. به‌طورکلی می‌توان نتیجه گرفت که ترکیب روش‌های کاهش بُعد کارآمد مانند ICA با مدل‌های یادگیری ماشین قوی به‌ویژه الگوریتم‌های تجمیعی نظیر جنگل تصادفی و XGBoost می‌تواند منجر به طراحی سامانه‌های تشخیص نفوذ با دقت و کارایی بالا شود. درعین‌حال، بسته به نیاز عملیاتی، می‌توان میان سرعت، دقت، و هزینه محاسباتی توازن برقرار کرد. برای پژوهش‌های آینده پیشنهاد می‌شود بررسی‌ها به سمت به‌کارگیری مدل‌های عمیق ترکیب روش‌های هیبریدی، و همچنین آزمایش در محیط‌های بلادرنگ و داده‌های جریان یافته گسترش یابد تا قابلیت کاربردی سامانه‌های تشخیص نفوذ بیش‌ازپیش تقویت گردد.

۶. مرجع‌ها

[۱] Zhang, S.; Cheng, D.; Deng, Z.; Zong, M.; Deng, X. "A Novel kNN Algorithm with Data-Driven k Parameter Computation"; Pattern Recognit. Lett. ۲۰۱۸, ۱۰۹, ٤٤–٥٤. DOI: ۱۰.۱۰۱٦/j.patrec.۲۰۱۷.۰۹.۰۳٦.

[۲] Zahiri, M.; Shirini, K.; Samadi Gharehveran, S. "Network Traffic Analysis with Machine Learning for Faster Detection of Distributed Denial of Service Attack"; J. Adv. Def. Sci. Technol. ۲۰۲٤, ۱٤, ۲۷۳–۲۸۲. (In Persian) DOR: ۲۰.۱۰۰۱. ۱.۲٦۷٦۲۹۳٥.۱٤۰۲.۱٤.٤.٦.۲.

[۳] Kazemitabar, J.; Taheri, R.; Kheradmandian, G. "A Novel Technique for Improvement of Intrusion Detection *via*

Combining Random Forest and Genetic Algorithm"; J. Adv. Def. Sci. Technol. ۲۰۱۹, ۱۰, ۲۸۷–۲۹۶. (In Persian) DOR: ۲۰.۱۰۰۱.۱.۲۶۷۶۲۹۳۵.۱۳۹۸.۱۰.۳.۹.۵

[٤] Samadi Gharehveran, S.; Ghassemzadeh, S.; Rostami, N. "Two-Stage Resilience-Constrained Planning of Coupled Multi-Energy Microgrids in the Presence of Battery Energy Storages"; Sustain. Cities Soc. ۲۰۲۲, ۸۳, ۱۰۳۹۵۲. DOI: ۱۰.۱۰۱٦/j.scs.۲۰۲۲.۱۰۳۹۵۲

[٥] Asadi, M.; Zarei, B. "Detection of Denial of Service Attacks by Ensemble Learning Method"; J. Adv. Def. Sci. Technol. ۲۰۲۳, ۱٤, ٥۱–٦۸. (In Persian) DOR: ۲۰.۱۰۰۱.۱.۲٦۷٦۲۹۳٥.۱٤۰۲.۱٤.۱.٥.٥

[٦] Alharthi, A.; Alaryani, M.; Kaddoura, S. "A Comparative Study of Machine Learning and Deep Learning Models in Binary and Multiclass Classification for Intrusion Detection Systems"; Array ۲۰۲٥, ۱۰۰٤۰٦. DOI: ۱۰.۱۰۱٦/j.array.۲۰۲٥.۱۰۰٤۰٦.

[۷] Badrinarayanan, V.; Kendall, A.; Cipolla, R. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation"; IEEE Trans. Pattern Anal. ۲۰۱۷, ۳۹, ۲٤۸۱–۲٤۹٥. DOI: ۱۰.۱۱۰۹/TPAMI.۲۰۱٦.۲٦٤٤٦۱٥.

[۸] Samadi Gharehveran, S.; Nasiri, M. "Resilient Planning Against Disturbances and Optimal Location Determination for Mobile Energy Storage Systems in Smart Microgrids"; Passive Defense ۲۰۲٥, ۱٦, ٦۹–۸۰. (In Persian) DOR: ۲۰.۱۰۰۱.۱.۲۰۰۸٦۸٤۹.۱٤۰٤.۱٦.۲.٦.۲.

[۹] Moustafa, N.; Slay, J. "UNSW-NB۱٥: A Comprehensive Data Set for Network Intrusion Detection Systems"; Mil. Commun. Inf. Syst. Conf. ۲۰۱٥. DOI: ۱۰.۱۱۰۹/MilCIS. ۲۰۱٥.۷۳٤۸۹٤۲.

[۱۰] Corea, P. M.; Liu, Y.; Wang, J.; Niu, S.; Song, H. "Explainable AI for Comparative Analysis of Intrusion Detection Models"; IEEE Int. Mediterr. Conf. Commun. Netw. ۲۰۲٤, ٥۸٥–٥۹۰. IEEE. DOI: ۱۰.۱۱۰۹/MeditCom ٦۱۰٥۷.۲۰۲٤.۱۰٦۲۱۳۳۹.

[۱۱] Malele, V.; Mathonsi, T. E. "Testing the Performance of Multi-Class IDS Public Dataset Using Supervised Machine Learning Algorithms"; arXiv.۲۳۰۲.۱٤۳۷٤, ۲۰۲۳. DOI: ۱۰.٤۸٥٥۰/arXiv.۲۳۰۲.۱٤۳۷٤.

[۱۲] Zoghi, Z.; Serpen, G. "Ensemble Classifier Design Tuned to Dataset Characteristics for Network Intrusion Detection"; arXiv.۲۲۰٥.۰٦۱۷۷, ۲۰۲۲. DOI: ۱۰.٤۸٥٥۰/ arXiv.۲۲۰٥.۰٦۱۷۷

[۱۳] Samadi Gharehveran, S.; G. Zadeh, S.; Rostami, N. "Resilience-Oriented Planning and Pre-Positioning of Vehicle-Mounted Energy Storage Facilities in Community Microgrids"; J. Energy Storage ۲۰۲۳, ۷۲, ۱۰۸۲٦۳. DOI: ۱۰.۱۰۱٦/j.est.۲۰۲۳.۱۰۸۲٦۳.

[۱٤] Ajagbe, S. A.; Bamidele, A. J.; Florez, H. "Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset"; SN Comput. Sci. ۲۰۲٤, ٥, ۱۰۲۸. DOI: ۱۰.۱۰۰۷/ s٤۲۹۷۹-۰۲٤-۰۳۳٦۹-۰.

[۱٥] Rastgoo, M.; Jalali, M. "Detection of Cybercrimes in Online Connections by the Data Mining Approach"; Passive Defense ۲۰۲۰, ۱۱, ٦۳–۷۰. (In Persian) DOR: ۲۰.۱۰۰۱. ۱.۲۰۰۸٦۸٤۹.۱۳۹۹.۱۱.۱.٦.٥.

[۱٦] Saheed, Y. K.; Misra, S. "A Voting Gray Wolf Optimizer-Based Ensemble Learning Models for Intrusion Detection in the Internet of Things"; Int. J. Inf. Secur. ۲۰۲٤, ۲۳, ۱٥٥۷-۱٥۸۱. DOI: ۱۰.۱۰۰۷/s۱۰۲۰۷-۰۲۳-۰۰۸۰۳-x.

[۱۷] Abbasi, R.; Javadzade, M. A. "Predicting Public Unrest Using Social Networks, Based on Machine Learning in the Natural Language Processing"; Passive Defense ۲۰۲۲, ۱۳, ٤٥-٥٦. (In Persian) DOR: ۲۰.۱۰۰۱.۱.۲۰۰۸٦۸٤۹.۱٤۰۱.۱۳.۳.٥.۲.

[۱۸] Nemati, R.; Shirini, K.; Samadi Gharehveran, S. "FER-HA: A Hybrid Attention Model for Facial Emotion Recognition"; J. Supercomput. ۲۰۲٥, ۸۱, ۱٤۸٥. DOI: ۱۰.۱۰۰۷/s۱۱۲۲۷-۰۲٥-۰۷۹۸۳-٤.

[۱۹] Saeedi, N.; Baharvand, D.; Shirini, K.; Samadi Gharehveran, S. "Prediction of Electrical Energy Consumption Using Principal Component Analysis and Independent Components Analysis"; J. Supercomput. ۲۰۲٥, ۸۱, ۱۰۷۲. DOI: ۱۰.۱۰۰۷/s۱۱۲۲۷-۰۲٥-۰۷٥۰٥-۲

[۲۰] Samadi Gharehveran, S.; Shirini, K.; Khavar, S. C.; Abdollahi, A. "Optimizing Day-Ahead Power Scheduling: A Novel MIQCP Approach for Enhanced SCUC with Renewable Integration"; e-Prime-Adv. Electr. Eng., Electron. Energy ۲۰۲٥, ۱۰۱۰۲۲. DOI: ۱۰.۱۰۱٦/j.prime. ۲۰۲٥.۱۰۱۰۲۲

[۲۱] Samadi Gharehveran, S.; Shirini, K.; Abdolahi, A. "Optimizing Energy Storage Solutions for Grid Resilience: A Comprehensive Overview"; intechopen ۲۰۲٥. DOI: ۱۰.٥۷۷۲/ intechopen.۱۰۰٦٤۹۹

[۲۲] Taherihajivand, A.; Shirini, K.; Samadi Gharehveran, S. "An Overview of Product Performance Prediction Using Artificial Algorithms"; J. Agric. Mechanization ۲۰۲٤, ۹, ۱–۱٤. DOI: ۱۰.۲۲۰۳٤/jam.۲۰۲٤.٦۱۸۹۹.۱۲۷٦.

[۲۳] Zaki Dizaji, H.; Shirini, K.; Taherihajivand, A.; Monjezi, N. "Modelling Variables Affecting the Yield of Sugarcane Fields Using Deep Recurrent Neural Network"; Iran. J. Biosyst. Eng. ۲۰۲٤, ٥٥, ۹۳–۱۰۸. DOI: ۱۰.۲۲۰٥۹/ijbse.۲۰۲٥.۳۷۸۹٥۸.٦٦٥٥٥۷.

[۲٤] Sattari, M. T.; Bagheri, R.; Shirini, K.; Allahverdipour, P. "Modeling Daily and Monthly Rainfall in Tabriz Using Ensemble Learning Models and Decision Tree Regression"; Clim. Change Res. ۲۰۲٤, ٥, ۳۱–٤۸. DOI: ۱۰.۳۰٤۸۸/ccr.۲۰۲٤.٤۳۳۳۹٤.۱۱۹۲.

[۲٥] Sattari, M. T.; Shirini, K.; Javidan, S. "Evaluating the Efficiency of Dimensionality Reduction Methods in Improving the Accuracy of Water Quality Index Modeling in Qizil-Uzen River Using Machine Learning Algorithms"; Water Soil Manag. Model. ۲۰۲٤, ٤, ۸۹–۱۰٤. DOI: ۱۰.۲۲۰۹۸/ mmws.۲۰۲۳.۱۲٤۳٤.۱۲٤۱.

[۲٦] Taherihajivand, A.; Shirini, K.; Samadi Gharehveran, S. "Weed Detection in Fields Using Convolutional Neural Network Based on Deep Learning"; Agric. Eng. ۲۰۲٤, ٤۷, ۱۲۹–۱٤۲. DOI: ۱۰.۲۲۰٥٥/agen.۲۰۲٤.٤٥۳۲۷.۱٦۸۸.

[۲۷] Zahiri, M.; Shirini, K.; Samadi Gharehveran, S. "Network Traffic Analysis with Machine Learning for Faster Detection of Distributed Denial of Service Attack"; J. Adv. Def. Sci. Technol. ۲۰۲٤, ۱٤, ۲۷۳–۲۸۲. DOR: ۲۰.۱۰۰۱.۱.۲٦۷٦۲۹۳٥.۱٤۰۲.۱٤.٤.٦.۲.

[۲۸] Shirini, K.; Taherihajivand, A.; Samadi Gharehveran, S. "A Review of Algorithms for Solving the Project Scheduling Problem with Resource Constraints Considering Agricultural Problems"; J. Agric. Mechanization ۲۰۲۳, ۸, ۱–۱٤. DOI: ۱۰.۲۲۰۳٤/jam.۲۰۲۳.٥٥۷٥۱.۱۲۲۷.

[۲۹] Tan, J.; Radhi, R. M.; Shirini, K.; Samadi Gharehveran, S.; Parisooz, Z.; Khosravi M.; Azarinfar, H. "Innovative Framework for Fault Detection and System Resilience in Hydropower Operations Using Digital Twins and Deep Learning"; Sci. Rep. ۲۰۲٥, ۱٥, ۱٥٦٦۹. DOI: ۱۰.۱۰۳۸/s٤۱٥۹۸-۰۲٥-۹۸۲۳٥-۱

[۳۰] Jin, K.; Banizaman, H.; Samadi Gharehveran, S.; Jokar, M. R.; Mohamadi Amidi, A. R.; Yu, J.; Oleiwi Shami; H. "Robust Power Management Capabilities of Integrated

Energy Systems in the Smart Distribution Network Including Linear and Non-Linear Loads"; Sci. Rep. ۲۰۲۵, ۱۵, ٦٦۱۵. DOI: ۱۰.۱۰۳۸/s٤۱٥۹۸-۰۲٥-۸۹۸۱۷-۰.

[۳۱] Shirini, K.; Aghdasi, H. S.; Saeedvand, S. "Modified Imperialist Competitive Algorithm for Aircraft Landing Scheduling Problem"; J. Supercomput. ۲۰۲٤, ۸۰(۱۰). DOI: ۱۰.۱۰۰۷/s۱۱۲۲۷-۰۲٤-۰٥۹۹۹-w.