KNO-1202-4805

ارائه روشی جهت افزایش دقت سیستم تشخیص نفوذ در برابر حملات در محاسبات ابری با رویکرد هوش مصنوعی، یادگیری ماشین و بهبود انتخاب ویژگی

محمدحسن فرهمند[۱]، Hasanfr277@gmail.com
موسی مجرد[۲]، musa.mojarad@iau.ac.ir
[۱،۲]گروه کامپیوتر، دانشگاه آزاد اسلامی، فیروزآباد، ایران

چکیده

امنیت رایانش ابری از اهمیت ویژه‌ای برخوردار است، به‌ویژه با توجه به گسترش استفاده از رایانش ابری در بیشتر سازمان‌های دولتی و خصوصی. سیستم تشخیص نفوذ پیشنهادی بر طراحی مدلی تمرکز داشته است که می‌تواند دقت تشخیص نفوذ را با استفاده از پیشرفت‌هایی در بخش انتخاب ویژگی‌ها افزایش دهد. روش پیشنهادی با ترکیب انتخاب ویژگی فیلتر و انتخاب خودکار ویژگی‌ها و بهینه‌سازی وزن‌ها با استفاده از الگوریتم COA در مدل تجمعی، نتایج بهتری نسبت به مدل‌های پیشرفته موجود به دست آورده است. رویکرد پیشنهادی قادر است انواع مختلف حملات را با نرخ تشخیص بالاتر و نرخ هشدار کاذب کمتری شناسایی کند. در این مقاله، عملکرد سه روش مختلف در مقایسه با یکدیگر با استفاده از معیارهای دقت کلی (ACC)، دقت (P)، فراخوانی (R)، و معیار F مورد بررسی قرار گرفت. روش مبتنی بر فیلتر با دقت کلی متوسط ٪۹۶،۵۴ تا ٪۹۸،۹۱ و دقت بالا در شناسایی نمونه‌های مثبت، عملکرد قابل قبولی داشت، اما مقدار فراخوانی آن به نسبت دیگر روش‌ها پایین‌تر بود. این روش تعادلی متوسط بین دقت و فراخوانی را از خود نشان داد. در مقابل، روش رمزگذار خودکار پشته‌ای با دقت کلی بین ٪۹۷،۶۹ تا ٪۹۹،۰۶ و دقت بالا در شناسایی نمونه‌های مثبت، بهبود قابل توجهی در معیار فراخوانی نشان داد و توانست درصد بیشتری از نمونه‌های مثبت را شناسایی کند. این روش تعادل بهتری بین دقت و فراخوانی داشت و از نظر معیار F نیز عملکرد بهتری داشت. روش ترکیب ویژگی‌ها در مقایسه با سایر روش‌ها بهترین عملکرد را از خود نشان داد. این روش با دقت کلی بین ٪۹۸،۹۸ تا ٪۹۹،۹۸ و دقت بسیار بالا در شناسایی نمونه‌های مثبت، تقریباً تمام نمونه‌های مثبت را شناسایی کرد و از نظر معیار F نیز برتری قابل توجهی نسبت به سایر روش‌ها داشت.

واژه‌های کلیدی: امنیت، رایانش ابری، تشخیص نفوذ، یادگیری ماشین و انتخاب ویژگی.

۱.مقدمه

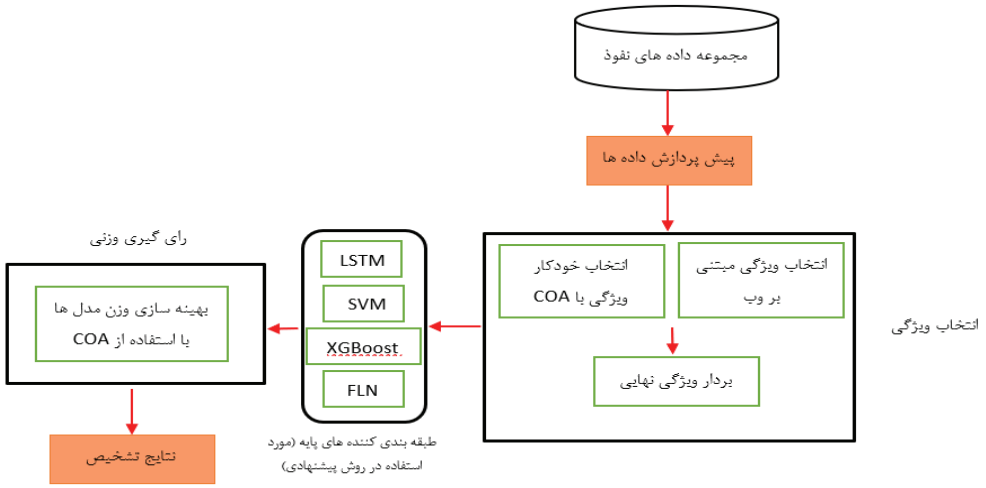
رایانش ابری به عنوان یک فناوری تحول‌آفرین، خدمات اینترنتی را با مدل پرداخت به ازای استفاده فراهم می‌کند و در سال‌های اخیر به دلیل مقرون‌به‌صرفه بودن و مقیاس‌پذیری بالا مورد توجه گسترده قرار گرفته است[۱-۲]. با وجود مزایای متعدد آن، چالش‌های امنیتی همچون نبود ابزارهای پزشکی قانونی مناسب برای بررسی رفتارهای مجرمانه در محیط ابری همچنان پابرجاست[۳]. از سوی دیگر، افزایش کاربران و داده‌های حساس در ابر، نیاز به سیستم‌های تشخیص نفوذ (IDS) دقیق و کارآمد را دوچندان کرده است، که این امر باعث شده استفاده از هوش مصنوعی و یادگیری ماشین برای تحلیل داده‌ها و شناسایی تهدیدات، به یک ضرورت تبدیل شود[۴]. یکی از راهکارهای کلیدی در افزایش دقت سیستم‌های IDS، بهبود انتخاب ویژگی است که با شناسایی ویژگی‌های مؤثر، باعث کاهش پیچیدگی داده‌ها و افزایش سرعت و دقت مدل‌های یادگیری ماشین می‌شود. این ترکیب از الگوریتم‌های هوش مصنوعی و انتخاب ویژگی‌های بهینه، می‌تواند عملکرد سیستم‌های امنیتی را در برابر تهدیدات سایبری به طور چشمگیری ارتقا دهد. همچنین با بهره‌گیری از مدل‌های پیشرفته یادگیری ماشین و یادگیری عمیق، تشخیص بلادرنگ حملات و کاهش نرخ هشدارهای اشتباه ممکن می‌شود. این سیستم‌ها به‌طور مداوم می‌توانند خود را با داده‌های جدید تطبیق دهند و عملکردشان را بهبود دهند[۵-۱۰]. مدل‌های متکی بر یادگیری ماشین، به‌ویژه مدل‌های گروهی مبتنی بر ترکیب طبقه‌بندها، در مقایسه با مدل‌های منفرد دقت بالاتری در تشخیص نفوذ دارند[12-11]. یادگیری عمیق نیز با شبکه‌های عصبی چندلایه توانایی شناسایی الگوهای پیچیده را دارد و نقش مهمی در امنیت شبکه‌های ابری ایفا می‌کند. ترکیب انتخاب ویژگی مبتنی بر فیلتر و رمزگذار خودکار پشته‌ای (Stacked Autoencoder) موجب حذف ویژگی‌های زائد و کاهش ابعاد داده می‌شود که در نهایت به افزایش دقت مدل منجر خواهد شد. در این رویکرد، مدل نهایی شامل مجموعه‌ای از طبقه‌بندها نظیر SVM، LSTM، XGBoost و FLN است که با الگوریتم بهینه‌سازی Crayfish (COA) تنظیم می‌شوند تا بهترین عملکرد طبقه‌بندی حاصل شود[13].

۲.کارهای گذشته

در سال‌های اخیر، پژوهش‌های متعددی با تمرکز بر بهبود سیستم‌های تشخیص نفوذ (IDS) با استفاده از روش‌های یادگیری ماشین و انتخاب ویژگی انجام شده‌اند. در این راستا، تحقیقات [14] تا [16] به توسعه مدل‌هایی پرداخته‌اند که با بهره‌گیری از الگوریتم‌هایی مانند MultiSVM، الگوریتم‌های فراابتکاری و شبکه‌های باور عمیق، بهینه‌سازی انتخاب ویژگی‌ها را دنبال می‌کنند تا دقت و سرعت تشخیص حملات سایبری افزایش یابد. این مطالعات نشان می‌دهند که با حذف ویژگی‌های غیرضروری و تمرکز بر ویژگی‌های کلیدی، می‌توان زمان پردازش را کاهش داد و نرخ هشدارهای اشتباه را به حداقل رساند. مطالعات [17] تا [22] نیز روش‌هایی را معرفی کرده‌اند که با استفاده از شبکه‌های عصبی عمیق، الگوریتم‌های ژنتیک و الگوریتم‌های بهینه‌سازی نوین مانند Honey Badger، عملکرد سیستم‌های IDS را بهبود می‌بخشند. این رویکردها اغلب در بسترهایی مانند رایانش ابری، شهرهای هوشمند و محیط‌های صنعتی به کار گرفته شده‌اند و هدف آن‌ها ارتقاء امنیت سایبری با تحلیل دقیق‌تر داده‌های شبکه و شناسایی الگوهای غیرعادی است. انتخاب ویژگی در این مطالعات نقش کلیدی در افزایش دقت و کاهش پیچیدگی مدل‌ها ایفا کرده است. در ادامه، تحقیقات [23] تا [26] تمرکز بیشتری بر کاربرد این روش‌ها در محیط‌های خاص مانند اینترنت اشیاء و رایانش ابری بلادرنگ داشته‌اند. استفاده از یادگیری عمیق در کنار انتخاب ویژگی‌های بهینه موجب شده است تا این سیستم‌ها قابلیت واکنش سریع‌تر و مؤثرتری در برابر حملات جدید و ناشناخته داشته باشند. این مطالعات به وضوح نشان می‌دهند که ترکیب الگوریتم‌های فراابتکاری و یادگیری ماشین می‌تواند راهکارهایی مؤثر برای مقابله با تهدیدات پیچیده در محیط‌های مدرن و داده‌محور ارائه دهد.

۳.روش پیشنهادی

سیستم تشخیص نفوذ ابر پیشنهادی شامل پیش پردازش برای رسیدگی به مشکل عدم تعادل کلاس است. مجموعه ویژگی های نهایی با ترکیب روش های فیلتر و انتخاب خودکار ویژگی ایجاد می شود. شکل۱ تکنیک پیشنهادی را نشان می دهد. مدل‌های ML و DL با استفاده از مجموعه ویژگی‌ها آموزش داده می‌شوند. سپس وزن‌های مدل‌های ML و DL با استفاده از رویکرد COA بهینه‌سازی می‌شوند و نتایج طبقه‌بندی نهایی از مدل مجموعه از طریق رأی‌گیری وزنی به‌دست می‌آیند.



شکل ۱- بلوک دیاگرام پیشنهادی.

این نمودار یک سیستم چندمرحله‌ای برای تشخیص نفوذ در شبکه را نمایش می‌دهد. ابتدا با استفاده از مجموعه داده نفوذ (Intrusion Dataset)، داده‌های خام مربوط به حملات یا رفتارهای مشکوک جمع‌آوری می‌شود. سپس مرحله پیش‌پردازش آغاز می‌شود که شامل تمیزسازی، عادی‌سازی و آماده‌سازی داده‌ها برای مراحل بعدی است. در گام انتخاب ویژگی، از دو روش استفاده می‌شود: یکی مبتنی بر فیلتر که از روش‌های آماری برای انتخاب ویژگی‌های مهم بهره می‌برد، و دیگری مبتنی بر رمزگذار خودکار پشته‌ای (SAE) که به صورت خودکار ویژگی‌های آموزنده را استخراج می‌کند. خروجی این دو روش ترکیب شده و بردار ویژگی نهایی شکل می‌گیرد که برای طبقه‌بندی استفاده می‌شود. در بخش طبقه‌بندی، سه مدل مختلف به‌کار گرفته می‌شود: شبکه عصبی LSTM برای تحلیل داده‌های متوالی، الگوریتم SVM برای طبقه‌بندی دقیق، و مدل XGBoost برای بهبود عملکرد و دقت تشخیص. نتایج این مدل‌ها در مرحله رأی‌گیری وزنی ترکیب می شوند، جایی که هر مدل وزنی متناسب با اهمیت خود دارد. این وزن‌ها به صورت بهینه با استفاده از الگوریتم بهینه سازی خرچنگ (COA) تعیین می شوند تا عملکرد کلی سیستم بهبود یابد. در نهایت، خروجی نهایی سیستم شامل تشخیص دقیق حملات و نفوذهای شناسایی شده خواهد بود. این فرآیند ترکیبی از تکنیک‌های هوش مصنوعی و بهینه سازی است که هدف آن افزایش دقت، اطمینان و بهره‌وری در شناسایی تهدیدات امنیتی شبکه می‌باشد.

```
% Pseudocode for improving feature selection and weighting using COA
% Step 1: Load the dataset
load('cloud_computing_dataset.mat'); % Load the cloud computing attack dataset
% Step 2: Preprocess the data
% Normalize and handle missing values
preprocessed_data = preprocessData(dataset);
% Step 3: Apply feature selection using filter and automatic models
% Filter method: Information Gain, Chi-square, etc.
selected_features_filter = featureSelectionFilter(preprocessed_data);
% Automatic method: Using a feature selection model like AE (autoencoder) or others
selected_features_automatic = featureSelectionAutomatic(preprocessed_data);
```

```
% Combine selected features from filter and automatic methods
final_selected_features = combineSelectedFeatures(selected_features_filter, selected_features_automatic);
% Step 4: Apply Crayfish Optimization Algorithm (COA) to optimize feature weights
% Initialize COA parameters
COA_params = initializeCOA();
optimized_weights = crayfishOptimizationAlgorithm(final_selected_features, COA_params);
% Step 5: Split the dataset into training and testing sets
[train_data, test_data] = splitData(preprocessed_data);
% Step 6: Train classifiers (LSTM, SVM, XGBoost, FLN) using the optimized feature set
% Train LSTM model
LSTM_model = trainLSTM(train_data, optimized_weights);
% Train SVM model
SVM_model = trainSVM(train_data, optimized_weights);
% Train XGBoost model
XGBoost_model = trainXGBoost(train_data, optimized_weights);
% Train FLN model
FLN_model = trainFLN(train_data, optimized_weights);
% Step 7: Evaluate the models on the test data
LSTM_accuracy = evaluateModel(LSTM_model, test_data);
SVM_accuracy = evaluateModel(SVM_model, test_data);
XGBoost_accuracy = evaluateModel(XGBoost_model, test_data);
FLN_accuracy = evaluateModel(FLN_model, test_data);
% Step 8: Combine predictions from all classifiers using a voting or weighted average scheme
combined_predictions = combineClassifications(LSTM_model, SVM_model, XGBoost_model, FLN_model);
% Step 9: Calculate the accuracy of the final prediction model
final_accuracy = calculateAccuracy(combined_predictions, test_data);
% Output the final accuracy
disp(['Final Accuracy of IDS using COA: ', num2str(final_accuracy)]);
% Supporting functions
function preprocessed_data = preprocessData(dataset)
  % Preprocess the dataset (normalization, handling missing values, etc.)
  % Return the preprocessed data
end
function selected_features_filter = featureSelectionFilter(data)
  % Apply filter-based feature selection methods (e.g., Information Gain, Chi-Square)
  % Return the selected features
end
function selected_features_automatic = featureSelectionAutomatic(data)
  % Apply automatic feature selection method (e.g., AE, Wrapper)
  % Return the selected features
end
function final_features = combineSelectedFeatures(filter_features, automatic_features)
  % Combine selected features from filter and automatic methods
  % Return the combined feature set
end
```

شکل ۲- شبه کد روش پیشنهادی

۴.پیش پردازش

این بخش به فرآیند پیش‌پردازش داده‌ها برای آماده‌سازی آن‌ها جهت انتخاب ویژگی و آموزش مدل می‌پردازد. از آن‌جا که مجموعه داده‌ها شامل داده‌های نمادین، باینری و عددی هستند و برخی ممکن است دارای مقادیر ناقص یا متناقض باشند، تمیزسازی داده‌ها ضروری است تا سیستم پیشنهادی فقط ورودی‌های عددی و پاک‌سازی‌شده را دریافت کند [27]. برای این منظور، در مجموعه داده NSL-KDD ویژگی‌هایی مانند "difficulty_level"، در کیوتو ویژگی‌هایی مانند «آدرس IP منبع» و «آدرس IP مقصد» و در مجموعه داده ۲۰۱۸CSE-CIC-IDS ویژگی‌هایی نظیر «مهر زمان»، «آدرس منبع»، «آدرس مقصد» و «درگاه منبع» حذف شدند تا تنها داده‌های مفید باقی بمانند [28].

در این مقاله، برای نرمال‌سازی مقادیر ویژگی‌ها، از مقیاس‌بندی Min-Max استفاده کرده‌ایم که در معادله زیر نشان داده شده است:

$$X_j^* = \left(X_j - X_{\min}\right) / \left(X_{\max} - X_{\min}\right) \quad (۱)$$

که در آن X_j مقدار اولیه است، X_j داده‌های پس از فرایند هستند، X_{min} کمترین مقدار از داده‌های تأثیرگذار در دنباله است و X_{max} بالاترین مقدار آن دنباله است [29].

در این بخش، داده‌ها ابتدا بر اساس نوع حمله برچسب‌گذاری شده و به دو گروه "نرمال" و "غیرنرمال" تقسیم می‌شوند؛ سپس بسته به مجموعه داده، به دسته‌های دقیق‌تری طبقه‌بندی می‌گردند: چهار نوع حمله در NSL-KDD [30]، دو نوع حمله شناخته‌شده و ناشناخته در Kyoto، و چهارده نوع حمله در ۲۰۱۸CSE-CIC-IDS. برای کاهش ابعاد و شناسایی ویژگی‌های مؤثر، از تکنیک آماری همبستگی ویژگی‌ها استفاده می‌شود [31] و داده‌های پیش‌پردازش‌شده به مجموعه‌های آموزشی و آزمایشی تقسیم می‌شوند. همچنین، به‌منظور مقابله با عدم تعادل کلاس‌ها از روش SMOTE استفاده می‌شود [32]. در مرحله انتخاب ویژگی (FS)، ویژگی‌های بی‌ربط حذف شده و ویژگی‌های مهم شناسایی می‌شوند تا دقت و کارایی مدل افزایش یابد [33]. در روش پیشنهادی، از تکنیک فیلتر برای انتخاب ویژگی استفاده می‌شود که با کمک روش‌های رتبه‌بندی، ویژگی‌های کلیدی را بدون وابستگی به الگوریتم طبقه‌بندی تعیین می‌کند [34]. از جمله روش‌های مورد استفاده در این بخش، می‌توان به بهره اطلاعاتی (IG)، نسبت افزایش، کای-مربع و عدم قطعیت متقارن اشاره کرد که IG با محاسبه آنتروپی، اهمیت هر ویژگی را سنجیده و ویژگی‌هایی با بیشترین اطلاعات را برای پیش‌بینی برچسب کلاس انتخاب می‌کند [35]. آنتروپی یک نمونه S به‌صورت زیر محاسبه می‌شود:

$$Entropy\left(S\right) = -\sum_{i}^{n} P_i \log_2 P_i \quad (۲)$$

در جایی که n به تعداد مقادیر طبقه بندی و P به تعداد نمونه های کلاس i اشاره می کند، به دست آوردن اطلاعات به صورت زیر داده می شود:

$$IG\left(S,A\right) = Entropy\left(S\right) - \sum Values\left(A\right)\left(\frac{|S_n|}{|S|}\right) EntropyS_n \quad (۳)$$

که در آن (A) ویژگی است، Values(A) مقادیر ممکن ویژگی است، |Sn| تعداد نمونه‌ها برای مقدار n، و |S|تعداد کل نمونه‌ها است. EntropySn آنتروپی برای نمونه‌ای است که مقدار n را دارد. طبق، افزایش اطلاعات (IG) ویژگی A به‌صورت زیر محاسبه می‌شود [36]:

$$IG\left(A\right) = Info\left(M\right) - Info_A\left(M\right) \quad (۴)$$

آنتروپی کل مجموعه داده به صورت Info(M) و آنتروپی ویژگی A به صورت InfoA(M) نمایش داده می شود.

انتخاب ویژگی مبتنی بر نسبت افزایش (GR) روش افزایش اطلاعات (IG) را بهبود می‌بخشد. نسبت افزایش زمانی بالا است که داده‌ها به طور یکنواخت توزیع شده باشند و مقدار کوچکتری می‌گیرد زمانی که داده‌ها تنها از یک شاخه ویژگی به دست می‌آیند [37]. نسبت افزایش با کمک اطلاعات تقسیم (split information) تعیین می‌شود [38].

$$Split\ info_{A}(M) = -\sum \frac{|M_n|}{|M|} * \log_2 \frac{M_n}{M} \quad (۵)$$

نسبت بهره، نسبت تعداد مقادیر احتمالی است که ویژگی x می تواند انتخاب کند به تعداد مقادیر واقعی ویژگی x.

$$Gain\ ratio(A) = IG(A) / Split\ Info(A) \quad (۶)$$

کای-مربع

روش کای-مربع یکی از رویکردهای انتخاب ویژگی است که برای یافتن رابطه بین دو متغیر استفاده می‌شود. استقلال بین دو رویداد با استفاده از این روش قابل تعیین است و می‌تواند به تعیین استقلال یک ویژگی از کلاس آن کمک کند. کای-مربع به‌صورت زیر محاسبه می‌شود:

$$\gamma^2 = \sum_{iz} (O_{iz} - E_{iz})^2 / E_{iz} \quad (۷)$$

متغیرهای i و z نشان دهنده دو متغیر هستند، O و E به ترتیب مقادیر مشاهده شده و مورد انتظار هستند و γ^2 نشان دهنده مقدار Chi-squared است.

عدم قطعیت متقارن

این نوع انتخاب ویژگی برای ارزیابی رتبه‌بندی نتیجه به‌کار می‌رود. عدم قطعیت متقارن به‌صورت زیر محاسبه می‌شود:

$$SU(i,j) = 2 * IG(i/j)/H(i) + H(j) \quad (۸)$$

که در آن آنتروپی ویژگی i و آنتروپی ویژگی j به ترتیب نشان داده می شود.

۶.مجموعه داده

مجموعه داده‌های NSL-KDD، کیوتو و ۲۰۱۸CSE-CIC-IDS به‌طور گسترده‌ای برای شناسایی نفوذ در شبکه‌ها استفاده می‌شوند. مجموعه NSL-KDD نسخه بهبودیافته‌ای از '۹۹KDD است که با حذف رکوردهای تکراری و غیرضروری به بهبود کیفیت داده‌ها پرداخته و حملات را به چهار دسته تقسیم می‌کند. مجموعه داده کیوتو، دارای ۲۴ ویژگی است و شامل حملات شناخته‌شده و ناشناخته می‌باشد. همچنین، مجموعه داده ۲۰۱۸CSE-CIC-IDS با ۱۴ نوع حمله و ۸۴ ویژگی، شامل شرایط واقعی شبکه‌های امروزی است و برای آزمون فرضیات از یک زیرمجموعه تصادفی از داده‌ها استفاده می‌شود. این مجموعه‌ها با ویژگی‌های منحصر به‌فرد خود، مبنای مناسبی برای ارزیابی الگوریتم‌های شناسایی نفوذ به شمار می‌روند.

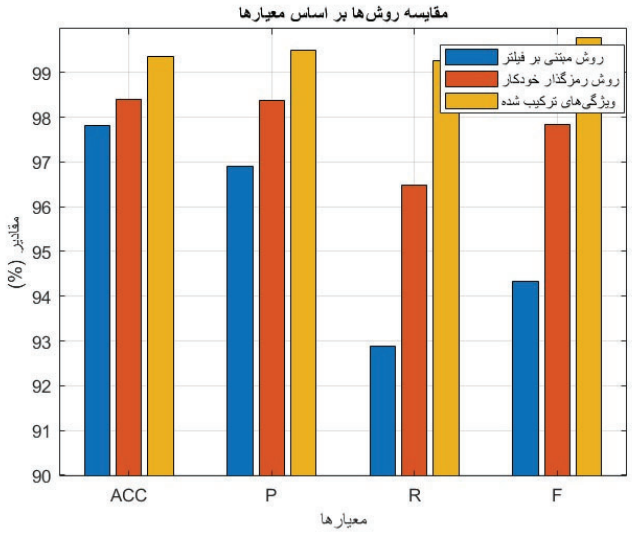
۵.نتایج شبیه سازی

در جدول ۱، عملکرد روش‌های مختلف انتخاب ویژگی مانند روش‌های فیلتر، خودرمزگذار انباشته و ویژگی‌های ترکیبی ارائه شده است. مشاهده می‌شود که روش ویژگی‌های ترکیبی توانسته است نتایجی بهتر از سایر روش‌ها نشان دهد.

جدول ۱. مقایسه عملکرد با انتخاب ویژگی مبتنی بر فیلتر، انتخاب ویژگی خودکار با رمزگذار خودکار پشته ای و ویژگی ها.

روش	مجموعه داده	تعداد ویژگی ها	ACC	P	R	F
روش مبتنی بر فیلتر	NSL-KDD	50	97.98	96.46	90.45	93.04
	Kyoto	10	96.54	94.64	95.43	94.82
	CSE-CIC-IDS-2018	6	98.91	99.64	92.79	95.11
خودکار با روش های رمزگذار خودکار پشته ای	NSL-KDD	93	98.48	98.53	94.55	97.34
	Kyoto	33	97.69	96.62	96.97	97.41
	CSE-CIC-IDS-2018	6	99.06	99.98	97.92	98.79

ویژگی های ترکیب شده	NSL-KDD	45	99.08	99.97	99.02	99.72
	Kyoto	15	98.98	98.55	98.94	99.65
	CSE-CIC-IDS-2018	6	99.98	99.95	99.86	99.96



شکل ۳- مقایسه عملکرد با انتخاب ویژگی مبتنی بر فیلتر، انتخاب ویژگی خودکار با رمزگذار خودکار پشته ای و ویژگی ها.

شکل فوق عملکرد سه روش مختلف را با استفاده از معیارهای دقت کلی (ACC)، دقت (P)، فراخوانی (R)، و معیار F مقایسه می‌کند. روش مبتنی بر فیلتر دقت کلی بین ۹۶/۵۴٪ تا ۹۸/۹۱٪ و دقت بین ۹۴/۶۴٪ تا ۹۹/۶۴٪ دارد، اما فراخوانی آن کمترین میزان (حدود ۹۲/۷۹٪) و معیار F در حدود ۹۳٪ تا ۹۵٪ است. روش رمزگذار خودکار پشته‌ای عملکرد بهتری دارد و دقت کلی آن بین ۹۷/۶۹٪ تا ۹۹/۰۶٪ و دقت آن بین ۹۶/۶۲٪ تا ۹۹/۹۸٪ متغیر است. این روش در فراخوانی نیز بهتر عمل کرده و مقدار F آن بالاتر از روش مبتنی بر فیلتر ا ست. روش ویژگی‌های ترکیب شده بهترین عملکرد را دارد، با دقت کلی بین ٪۹۸/۹۸ تا ۹۹/۹۸٪، دقت بالای ۹۹٪، فراخوانی بیشتر از ۹۹٪، و مقدار F بین ۹۹/۶۵٪ تا ۹۹/۹۶٪ که نشان‌دهنده برتری آن در ایجاد تعادل بین دقت و فراخوانی است.

جدول ۲. مقایسه عملکرد در شرایط استفاده و عدم استفاده از COA

روش	مجموعه داده	F	R	ACC	P
بدون الگوریتم فراابتکاری	NSL-KDD	92.60	90.21	96.01	96.45
	Kyoto	93.14	95.41	95.19	92.92
	CSE-CIC-IDS-2018	92.55	96.01	98.15	93.01
با الگوریتم فراابتکاریCOA	NSL-KDD	99.77	99.08	99.01	99.96
	Kyoto	99.53	98.92	99.90	98.17
	CSE-CIC-IDS-2018	99.96	99.91	99.99	99.95

جدول داده‌ها عملکرد دو روش را از نظر معیارهای اصلی مقایسه می‌کند: معیار F، فراخوانی (R)، دقت کلی (ACC) و دقت (P). در روش بدون الگوریتم فراابتکاری، مقدار F بین ۹۲,۵۵٪ تا ۹۳,۱۴٪ و فراخوانی بین ۹۰,۲۱٪ تا ۹۶,۰۱٪ متغیر است که نشان‌دهنده

توانایی محدودتر در شناسایی نمونه‌های مثبت است. دقت کلی این روش بین ۹۵,۱۹٪ و ۹۸,۱۵٪ و دقت آن بین ۹۲,۹۲٪ و ۹۶,۴۵٪ ا ست. در روش با الگوریتم فراابتکاری COA، مقدار F بین ۹۹,۵۳٪ تا ۹۹,۹۶٪، فراخوانی بین ۹۸,۹۲٪ تا ۹۹,۹۱٪ و دقت کلی نزدیک به ۹۹,۹۹٪ اسـت که عملکرد بسـیار بالاتری در شـناسـایی نمونه‌های مثبت و ایجاد تعادل بین دقت و فراخوانی نشـان می‌دهد. به‌طور کلی، اسـتفاده از الگوریتم فراابتکاری COA به‌طور چشـمگیری کارایی را در تمامی معیارها بهبود بخشـیده و نشـان می‌دهد که این الگوریتم می‌تواند به‌عنوان ابزاری مؤثر در افزایش دقت و شناسایی به کار رود.

جدول ۳. مقایسه مدل پیشنهادی با روش‌های پیشرفته موجود بر روی مجموعه داده NSL-KDD

منبع	نوع روش پیاده سازی	ACC	F	P	R
[39]	AE-RL	80.15	79.41	79.73	80.18
[40]	CNN-1D	78.97	85.14	79.41	84.50
[41]	AFSA-GA-PSO-DBN	82.36	84.50	85.14	84.59
[42]	CNN-BiLSTM	83.59	85.14	85.82	84.50
[43]	DRNN	93.68	93.29	91.23	95.27
[44]	GA-KELM	83.59	93.29	83.63	94.52
[45]	OCNN-HMLSTM	90.71	92.46	86.88	95.90
[46]	Ensemble	99.01	99.51	99.95	99.08
روش پیشنهادی	COA	99.64	99.41	99.93	99.54

جدول فوق به مقایسه عملکرد مدل‌ها در معیارهای مختلف د سته‌بندی می‌پردازد. مدل‌های [46] و روش پی شنهادی در دقت کلی (ACC) بالاترین امتیاز را دارند (۹۹,۰۱ و ۹۹,۶۴ به ترتیب). در معیار F-Score، مدل [46] بالاترین مقدار (۹۹,۵۱) را دارد، در حالی که روش پیشـنهادی با ۹۹,۴۱ در جایگاه بعدی قرار دارد. از نظر دقت (P)، مدل [46] با ۹۹,۹۵ و روش پیشـنهادی با ۹۹,۹۳ درصـد پیش‌بینی‌های مثبت دقیقی ارائه داده‌اند. در معیار بازیابی (R)، مدل [46] و روش پیشنهادی به ترتیب ۹۹,۰۸ و ۹۹,۵۴ درصد از موارد مثبت را شناسایی کرده‌اند. در تحلیل کلی، مدل [46] عملکرد عالی در دقت و F-Score دارد، اما روش پیشنهادی در تمامی معیارها به ویژه در دقت کلی و بازیابی عملکرد بهتری ن سبت به مدل‌های دیگر دا شته و به عنوان گزینه‌ای منا سب‌تر برای م سائل م شابه معرفی می‌شود.

جدول ۴. مقایسه مدل ترکیبی پیشنهادی با روش‌های متداول بر روی مجموعه داده Kyoto

منبع	نوع روش پیاده سازی	F	P	ACC	R
[47]	VAE-Label	86.03	97.50	97.95	75.43
[48]	BA-ELM	56.83	75.43	96.97	98.83
[49]	HIDS(NBFS-OSVM-PKNN)	56.83	56.83	75.43	94.73
[46]	Ensemble	98.53	98.16	98.99	98.93
روش پیشنهادی	COA	99.01	98.69	99.04	99.03

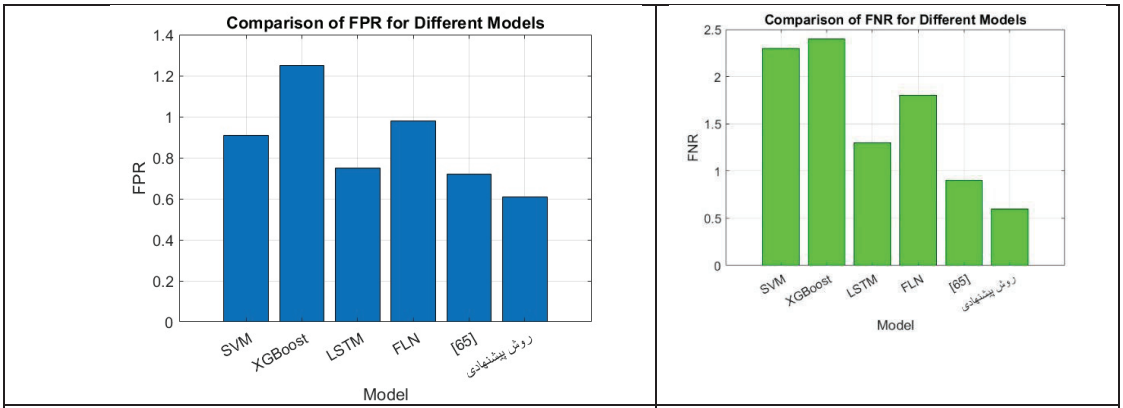
جدول مقایسـه عملکرد مدل ترکیبی پیشـنهادی با چهار مدل متداول ([47]، [48]، [49]، و [46]) بر روی مجموعه داده Kyoto نشـان می‌دهد که مدل پیشـنهادی در تمامی شـاخص‌های Precision (P)، F-Score (F)، Accuracy (ACC) و Recall (R) بهترین عملکرد را دارد. مدل پیشـنهادی با ۹۹,۰۱ در Precision، ۹۸,۶۹ در F-Score، ۹۹,۰۴ در Accuracy و ۹۹,۰۳ در Recall پیشـی می‌گیرد. مدل [46] نیز عملکرد بسـیار خوبی با ۹۸,۵۳ در Precision و ۹۸,۱۶ در F-Score دارد، اما مدل‌های [47] و [48] به ویژه در

Precision و F-Score ضعیف‌تر هستند. مدل [49] پایین‌ترین دقت و بازیابی را نشان می‌دهد، که ممکن است به مشکلات یادگیری یا پیش‌بینی آن اشاره داشته باشد. در نتیجه، مدل پیشنهادی به طور قابل توجهی بهتر از سایر روش‌ها عمل کرده و گزینه مناسبی برای مجموعه داده Kyoto به نظر می‌رسد.

جدول ۵. مقایسه مدل ترکیبی پیشنهادی با روش‌های دیگر بر روی مجموعه داده CSE-CIC-IDS۲۰۱۸

منبع	نوع روش پیاده سازی	P	F	ACC	R
[50]	IG+GR+CS-SVM	94.02	93.97	97.89	94.93
[51]	DSSTE+miniVGGNet	98.46	98.04	96.99	97.97
[52]	PTDAE+DNN	96.38	96.11	96.79	96.79
[46]	Ensemble	99.95	99.90	99.88	99.17
روش پیشنهادی	COA	99.97	99.91	99.03	99.23

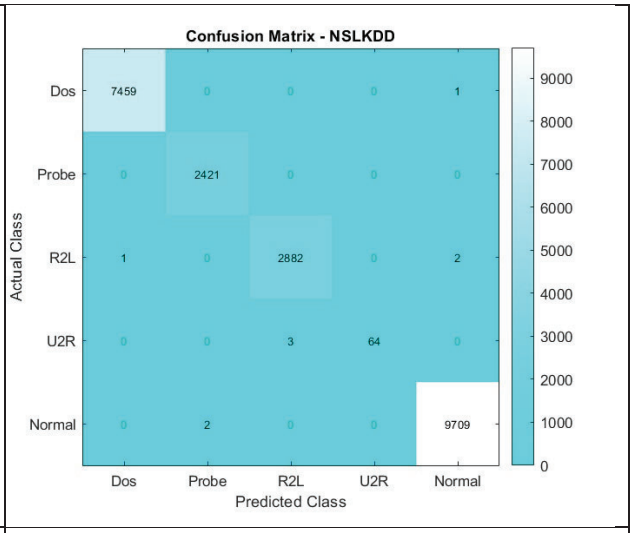
جدول مقایسه عملکرد مدل ترکیبی پیشنهادی با چهار روش دیگر بر روی مجموعه داده CSE-CIC-IDS۲۰۱۸ نشان می‌دهد که مدل پیشـــنهادی در تمامی معیارهای Precision (P)، F-Score (F)، Accuracy (ACC) و Recall (R) بهترین عملکرد را دارد. مدل پیشـــنهادی با ۹۹,۹۷ در Precision، ۹۹,۹۱ در F-Score، ۹۹,۰۳ در Accuracy و ۹۹,۲۳ در Recall پیشـــی می‌گیرد. مدل [46] نیز عملکرد بســیار خوبی با ۹۹,۹۰ در Precision و ۹۹,۸۸ در Accuracy دارد، ولی مدل پیشـنهادی در دیگر معیارها از آن پیشـی گرفته اســت. مدل‌های [50] و [52] به ویژه در Precision و Recall عملکرد ضـعیف‌تری دارند، و مدل [51] نیز در برخی معیارها نســبت به دیگر مدل‌ها کمی پایین‌تر قرار دارد. در نتیجه، مدل پی شنهادی به طور قابل توجهی بهتر از سایر روش‌ها عمل کرده و گزینه منا سبی برای مجموعه داده CSE-CIC-IDS۲۰۱۸ به نظر می‌رسد.



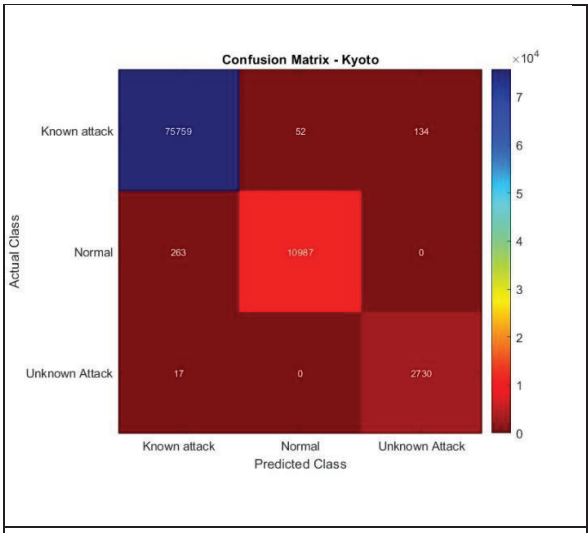
(الف) FPR (ب) FNR

شکل ۴- مقایسه FPR و FNR بین مدل‌های جداگانه و مدل ترکیبی برای مجموعه داده NSL-KDD. (الف) FPR و (ب) FNR.

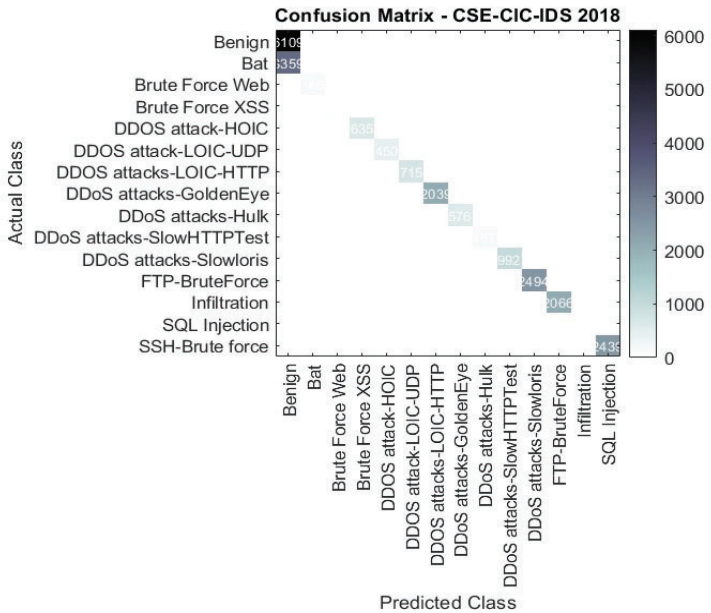
به طور کلی، شکل ۴ مقایسه‌ای از FPR و FNR بین هر یک از مدل‌های جداگانه (SVM، XGBoost، LSTM و FLN) استفاده‌شده در مدل ترکیبی و پیشنهاد ما که بر روی مجموعه داده NSL-KDD آزمایش شده است، ارائه می‌دهد. مقایسه عملکرد انجام‌شده نشان می‌دهد که مدل ترکیبی در مقایسه با مدل‌های فردی در کاهش FPR و FNR موفق‌تر است.



شکل ۵- ماتریس سردرگمی مجموعه داده NSL-KDD.



شکل ۶- ماتریس سردرگمی مجموعه داده کیوتو.



شکل ۷- ماتریس سردرگمی مجموعه داده CSE-CIC-IDS۲۰۱۸.

شکل‌های ۵ الی ۷ به ترتیب ماتریس‌های سردرگمی مجموعه داده‌های NSL-KDD، Kyoto و CSE-CIC-IDS۲۰۱۸ را نشان می‌دهند که در آن‌ها مدل ترکیبی پیشنهادی عملکرد بسیار دقیقی داشته است؛ به‌طوری که در NSL-KDD و CSE-CIC-IDS۲۰۱۸،

مدل بدون خطا نمونه‌های مربوط به حملات را پیش‌بینی کرده و در Kyoto نیز تنها ۲۶۳ نمونه از حملات شناخته‌شده را به‌اشتباه شناسایی کرده است. این نتایج نشان‌دهنده توان بالای مدل پیشنهادی در تفکیک دقیق حملات شبکه و ترافیک عادی هستند. با این حال، برخی محدودیت‌ها در این تحقیق وجود دارد، از جمله کمبود مجموعه‌داده‌های جامع و به‌روز، تأثیر منفی مجموعه‌داده‌های نامتوازن که با استفاده از تکنیک SMOTE مدیریت شده است، عدم ارزیابی در محیط‌های واقعی و نیاز بالای محاسباتی مدل‌های یادگیری عمیق؛ که برای آینده، استفاده از الگوریتم‌های الهام‌گرفته از طبیعت جهت بهینه‌سازی و کاهش مصرف منابع توصیه می‌شود.

۶.نتیجه‌گیری

امنیت رایانش ابری به‌دلیل استفاده گسترده از آن در سازمان‌های دولتی و خصوصی، از اهمیت بالایی برخوردار است. مدل تشخیص نفوذ پیشنهادی با تمرکز بر بهبود فرآیند انتخاب ویژگی و بهره‌گیری از الگوریتم COA در یک مدل ترکیبی، عملکردی بهتر نسبت به روش‌های موجود ارائه داده است. این مدل با ترکیب انتخاب ویژگی فیلتر، انتخاب خودکار ویژگی‌ها و بهینه‌سازی وزن‌ها، توانسته است نرخ تشخیص حملات را افزایش داده و نرخ هشدارهای کاذب را کاهش دهد. در مقایسه میان سه روش مختلف، روش فیلتر دقت کلی [۹۶,۵۴٪ تا ۹۸,۹۱٪] را به همراه دقت بالا در شناسایی نمونه‌های مثبت نشان داد، اما مقدار Recall پایین‌تری داشت. در مقابل، روش رمزگذار خودکار پشته‌ای با دقت کلی [۹۷,۶۹٪ تا ۹۹,۰۶٪] عملکرد بهتری در Recall و F-Score داشت. در نهایت، روش ترکیب ویژگی‌ها بهترین نتایج را با دقت کلی [۹۸,۹۸٪ تا ۹۹,۹۸٪] و دقت بسیار بالا در شناسایی نمونه‌های مثبت به‌دست آورد و از نظر F-Score نیز نسبت به سایر روش‌ها برتری داشت. مدل ترکیبی پیشنهادی در تمام معیارها از جمله دقت (P)، فراخوانی (R)، F-Score و دقت کلی (ACC) عملکرد برتری نشان داد و توانایی بالایی در شناسایی نمونه‌های مثبت دارد. برای کارهای آینده، هدف بررسی و مقایسه مصرف منابع محاسباتی مدل پیشنهادی با سایر رویکردها خواهد بود. همچنین، پیشنهاد می‌شود از الگوریتم‌های الهام‌گرفته از طبیعت برای بهینه‌سازی پارامترها و کاهش منابع مصرفی استفاده شود. از طرفی، برای اعتبارسنجی بیشتر، استفاده از جدیدترین مجموعه داده‌هایی که تنوع بالایی در نوع حملات دارند و نمایانگر رفتار واقعی شبکه هستند، پیشنهاد می‌شود. این مجموعه داده‌ها می‌توانند به بهبود دقت مدل و متعادل‌سازی داده‌ها کمک کرده و باعث ارتقاء بیشتر عملکرد سیستم پیشنهادی گردند.

منابع و مآخذ

[۱]Kumar, R.R.; Tomar, A.; Shameem, M.; Alam, M.D. Optcloud: An optimal cloud service selection framework using QoS correlation lens. Comput. Intell. Neurosci. ۲۰۲۲, ۲۰۲۲, ۲۰۱۹۴۸۵. [Google Scholar] [CrossRef] [PubMed]

[۲]Kumar, R.R.; Shameem, M.; Khanam, R.; Kumar, C. A hybrid evaluation framework for QoS based service selection and ranking in cloud environment. In Proceedings of the ۱۵ ۲۰۱۸th IEEE India Council International Conference (INDICON), Coimbatore, India, ۱۶– ۱۸December ۲۰۱۸; pp. ۱–۶. [Google Scholar] [CrossRef]

[۳]Cinar, B., & Bharadiya, J. P. (۲۰۲۳). Cloud computing forensics; challenges and future perspectives: A review. Asian Journal of Research in Computer Science, ۱۶(۱), ۱۴-۱.

[۴]Sanagana, D. P. R., & Tummalachervu, C. K. (۲۰۲۴, May). Securing Cloud Computing Environment via Optimal Deep Learning-based Intrusion Detection Systems. In ۲۰۲۴Second International Conference on Data Science and Information System (ICDSIS) (pp. ۶-۱). IEEE.

[۵]Kumar, R.R.; Shameem, M.; Kumar, C. A computational framework for ranking prediction of cloud services under fuzzy environment. Enterp. Inf. Syst. ۲۰۲۲, ۱۶, ۱۶۷–۱۸۷. [Google Scholar] [CrossRef]

[۶]Akbar, M.A.; Shameem, M.; Mahmood, S.; Alsanad, A.; Gumaei, A. Prioritization based taxonomy of cloud-based outsource software development challenges: Fuzzy AHP analysis. Appl. Soft Comput. ۲۰۲۰, ۹۵, ۱۰۶۵۵۷. [Google Scholar] [CrossRef]

[۷]Bakro, M.; Bisoy, S.K.; Patel, A.K.; Naal, M.A. Performance Analysis of Cloud Computing Encryption Algorithms. In Advances in Intelligent Computing and Communication; Springer: Singapore, ۲۰۲۱; pp. ۳۵۷– ۳۶۷.

[۸]Aryachandra, A. A., Arif, Y. F., & Anggis, S. N. (۲۰۱۶, May). Intrusion Detection System (IDS) server placement analysis in cloud computing. In ۴ ۲۰۱۶th International Conference on Information and Communication Technology (ICoICT) (pp. ۵-۱). IEEE.

[۹]Bhushan, K.; Gupta, B.B. Security challenges in cloud computing: State-of-art. Int. J. Big Data Intell. ۲۰۱۷, ۴, ۸۱–۱۰۷. [Google Scholar] [CrossRef]

[۱۰]Bakro, M.; Bisoy, S.K.; Patel, A.K.; Naal, M.A. Hybrid Blockchain-Enabled Security in Cloud Storage Infrastructure Using ECC and AES Algorithms. In Blockchain Based Internet of Things; Springer: Singapore, ۲۰۲۲; pp. ۱۳۹– ۱۷۰.

[۱۱]Mighan, S.N.; Kahani, M. A novel scalable intrusion detection system based on deep learning. Int. J. Inf. Secur. ۲۰۲۱, ۲۰, ۳۸۷–۴۰۳. [Google Scholar] [CrossRef]

[۱۲]Mayuranathan, M.; Murugan, M.; Dhanakoti, V. Best features based intrusion detection system by RBM [] model for detecting DDoS in cloud environment. J. Ambient. Intell. Humaniz. Comput. ۲۰۲۱, ۱۲, ۳۶۰۹– ۳۶۱۹.

[۱۳]Arora, N.; Kaur, P.D. A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Appl. Soft Comput. ۲۰۲۰, ۸۶, ۱۰۵۹۳۶.

[۱۴]Turukmane, A. V., & Devendiran, R. (۲۰۲۴). M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. Computers & Security, ۱۳۷, ۱۰۳۵۸۷.

[۱۵]Ali, A. H., Ammar, B., Charfeddine, M., & Hamed, B. B. (۲۰۲۴, September). Enhanced intrusion detection based hybrid meta-heuristic feature selection. In International Conference on Computational Collective Intelligence (pp. ۱۵-۳). Cham: Springer Nature Switzerland.

[۱۶]Sarkar, N., Keserwani, P. K., & Govil, M. C. (۲۰۲۴). A better and fast cloud intrusion detection system using improved squirrel search algorithm and modified deep belief network. Cluster Computing, ۲۷(۲), .۱۷۱۸-۱۶۹۹

[۱۷]Sharma, H. S., & Singh, K. J. (۲۰۲۴). A feed forward deep neural network model using feature selection for cloud intrusion detection system. Concurrency and Computation: Practice and Experience, ۳۶(۹), e.۸۰۰۱

[۱۸]Tomar, A., Umrao, S., & Kumar, D. (۲۰۲۴, March). To Decrease the Rate of Cyber Anomalies Using Intrusion Detection System with Feature Selection Approach. In ۲ ۲۰۲۴nd International Conference on Disruptive Technologies (ICDT) (pp. ۱۴۴۲-۱۴۳۹). IEEE.

[۱۹]Pandithurai, O., Venkataiah, C., Tiwari, S., & Ramanjaneyulu, N. (۲۰۲۴). DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment. Expert Systems with Applications, ۲۴۱, .۱۲۲۵۴۴

[۲۰]Gill, K. S., & Dhillon, A. (۲۰۲۴). A hybrid machine learning framework for intrusion detection system in smart cities. Evolving Systems, .۱۵-۱

[۲۱]Rathod, G., Sabnis, V., & Jain, J. K. (۲۰۲۴). Intrusion Detection System (IDS) in Cloud Computing using Machine Learning Algorithms: A Comparative Study. Grenze International Journal of Engineering & Technology (GIJET), .(۱)۱۰

[۲۲]Fang, Y., Yao, Y., Lin, X., Wang, J., & Zhai, H. (۲۰۲۴). A feature selection based on genetic algorithm for intrusion detection of industrial control systems. Computers & Security, ۱۳۹, .۱۰۳۶۷۵

[۲۳]Sangaiah, A. K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., & Balasubramanian, S. (۲۰۲۳). A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Cluster Computing, ۲۶(۱), .۶۱۲-۵۹۹

[۲۴]Sah, G., Banerjee, S., & Singh, S. (۲۰۲۳). Intrusion detection system over real-time data traffic using machine learning methods with feature selection approaches. International Journal of Information Security, ۲۲(۱), .۲۷-۱

[۲۵]Kavitha, S., Gadde, S., Thatikonda, R., Vaddadi, S. A., Naresh, E., & Pareek, P. K. (۲۰۲۳). Enhancing Data Security in Cloud Computing with Optimized Feature Selection and Machine Learning for Intrusion Detection.

[۲۶]Zhang, L., Liu, K., Xie, X., Bai, W., Wu, B., & Dong, P. (۲۰۲۳). A data-driven network intrusion detection system using feature selection and deep learning. Journal of Information Security and Applications, ۷۸, .۱۰۳۶۰۶

[۲۷] Wang, D.; Zhang, Z.; Bai, R.; Mao, Y. A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring. J. Comput. Appl. Math. 2018, 329, 307–321. [Google Scholar] [CrossRef]

[۲۸] Liu, L.; Wang, P.; Lin, J.; Liu, L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. IEEE Access 2020, 9, 7550–7563. [Google Scholar] [CrossRef]

[۲۹] Ren, F.; Long, D. Carbon emission forecasting and scenario analysis in Guangdong Province based on optimized Fast Learning Network. J. Clean. Prod. 2021, 317, 128408. [Google Scholar] [CrossRef]

[۳۰] Wang, Z. Deep learning-based intrusion detection with adversaries. IEEE Access 2018, 6, 38367–38384. [Google Scholar] [CrossRef]

[۳۱] Xiao, P.; Qu, W.; Qi, H.; Li, Z. Detecting DDoS attacks against data center with correlation analysis. Computer Communications 2015, 67, 66–74.

[۳۲] Ma, X.; Shi, W. Aesmote: Adversarial reinforcement learning with smote for anomaly detection. IEEE Trans. Netw. Sci. Eng. 2020, 8, 943–956. [Google Scholar] [CrossRef]

[۳۳] Wang, W.; Liu, J.; Pitsilis, G.; Zhang, X. Abstracting massive data for lightweight intrusion detection in computer networks. Inf. Sci. 2018, 433, 417–430. [Google Scholar] [CrossRef]
[۳۴] Chandrashekar, G.; Sahin, F. A survey on feature selection methods. Comput. Electr. Eng. 2014, 40, 16–28. [Google Scholar] [CrossRef]
[۳۵] Suman, C.; Tripathy, S.; Saha, S. Building an effective intrusion detection system using unsupervised feature selection in multi-objective optimization framework. arXiv 2019, arXiv:1905.06562. [Google Scholar] [CrossRef]
[۳۶] Omuya, E.O.; Okeyo, G.O.; Kimwele, M.W. Feature selection for classification using principal component analysis and information gain. Expert Syst. Appl. 2021, 174, 114765. [Google Scholar] [CrossRef]
[۳۷] Arora, N.; Kaur, P.D. A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Appl. Soft Comput. 2020, 86, 105936. [Google Scholar] [CrossRef]
[۳۸] Krishnaveni, S.; Sivamohan, S.; Sridhar, S.; Prabakaran, S. Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Clust. Comput. 2021, 24, 1761–1779. [Google Scholar] [CrossRef]
[۳۹] Caminero, G.; Lopez-Martin, M.; Carro, B. Adversarial environment reinforcement learning algorithm for intrusion detection. Comput. Netw. 2019, 159, 96–109. [Google Scholar] [CrossRef]

[۴۰] Verma, A.K.; Kaushik, P.; Shrivastava, G. A network intrusion detection approach using variant of convolution neural network. In Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019; pp. 409–416. [Google Scholar] [CrossRef]
[۴۱] Wei, P.; Li, Y.; Zhang, Z.; Hu, T.; Li, Z.; Liu, D. An optimization method for intrusion detection classification model based on deep belief network. IEEE Access 2019, 7, 87593–87605. [Google Scholar] [CrossRef]
[۴۲] Jiang, K.; Wang, W.; Wang, A.; Wu, H. Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access 2020, 8, 32464–32476. [Google Scholar] [CrossRef]
[۴۳] Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory 2020, 101, 102031. [Google Scholar] [CrossRef]
[۴۴] Ghasemi, J.; Esmaily, J.; Moradinezhad, R. Intrusion detection system using an optimized kernel extreme learning machine and efficient features. Sādhanā 2020, 45, 1–9. [Google Scholar] [CrossRef]
[۴۵] Kanna, P.R.; Santhi, P. Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features. Knowl.-Based Syst. 2021, 226, 107132. [Google Scholar] [CrossRef]
[۴۶] Bakro, M., Kumar, R. R., Alabrah, A. A., Ashraf, Z., Bisoy, S. K., Parveen, N., ... & Abdelsalam, A. (2023). Efficient intrusion detection system in the cloud using fusion feature selection approaches and an ensemble classifier. Electronics, 12(11), 2427.
[۴۷] Malaiya, R.K.; Kwon, D.; Kim, J.; Suh, S.C.; Kim, H.; Kim, I. An empirical evaluation of deep learning for network anomaly detection. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 893–898. [Google Scholar] [CrossRef]
[۴۸] Shen, Y.; Zheng, K.; Wu, C.; Zhang, M.; Niu, X.; Yang, Y. An ensemble method based on selection using bat algorithm for intrusion detection. Comput. J. 2018, 61, 526–538. [Google Scholar] [CrossRef]
[۴۹] Saleh, A.I.; Talaat, F.M.; Labib, L.M. A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artif. Intell. Rev. 2019, 51, 403–443. [Google Scholar] [CrossRef]
[۵۰] Bakro, M.; Kumar, R.R.; Bisoy, S.K.; Addas, M.O.; Khamis, D. Developing a Cloud Intrusion Detection System with Filter-Based Features Selection Techniques and SVM Classifier. In Proceedings of the Computing, Communication and Learning: First International Conference (CoCoLe 2022), Warangal, India, 27–29 October 2022; Springer: Cham, Switzerland, 2023; pp. 15–26. [Google Scholar] [CrossRef]
[۵۱] Liu, L.; Wang, P.; Lin, J.; Liu, L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. IEEE Access 2020, 9, 7550–7563. [Google Scholar] [CrossRef]
[۵۲] Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. J. Inf. Secur. Appl. 2021, 58, 102804. [Google Scholar] [CrossRef]