Karafan

Quarterly Scientific Journal of National University of Skills

spring 2025, Vol. 22, No. 1, p. 127-149

Journal Homepage: https://karafan.nus.ac.ir/?lang=en

10.48301/KSSA.2025.476687.2997

Improving the accuracy of intrusion detection using the combined method of PCA-GWO and a deep neural network

Zahra Vakilzadeh [1], Zahra Heydaran Daroogheh Amnyieh [2], Iman Zabbah*[3], Zeinab Binaie [4]

[1] Department of Computer Engineering, Torbat Heydariyeh University, Torbat Heydariyeh, Iran.
[2] Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.
[3] Department of Computer, Torbat Heydariyeh Branch, Islamic Azad University, Torbat Heydariyeh, Iran.
[4] School of Mathemetics and Computer Science, Damghan University, Damghan, Iran.

ARTICLE INFO

Article Type:
Original Research

Received: 29.02.2024
Revised: 10.11.2024
Accepted: 13.01.2025

Keyword:
Intrusion Detection
Deep Learning
LSTM Network
PCA
Gray Wolf Algorithm

***Corresponding Author:**
Iman Zabbah
Email: *imanzabbah@gmail.com*

ABSTRACT

Computer networks play a vital role in communication and data exchange. However, with the expansion of these networks, the potential for cyber attacks and unauthorised access has also increased. In the real world, constant changes in traffic patterns and the emergence of new threats make the need for rapid and up-to-date training of intrusion detection models essential. Intrusions encompass illegal activities that compromise the integrity, confidentiality, and availability of organisational resources. As a critical component of network security, Intrusion Detection Systems (IDS) monitor for attacks that may go undetected by traditional firewalls. However, different types of attacks exhibit unique behaviours, and enhancing the detection of these attack types remains a significant challenge for intrusion detection models. In this research, a deep learning method is proposed that incorporates dimensionality reduction and optimal feature selection. Initially, Principal Component Analysis was applied for dimensionality reduction. Subsequently, the Gray Wolf Optimisation (GWO) algorithm was used to select superior features. Finally, key features were extracted to determine the presence of an attack and apply them to a deep Long Short-Term Memory (LSTM) network. The learning process was conducted using the NSL_KDD dataset. One of the key aspects of this research was the integration of PCA and GWO to extract the most relevant features while reducing dimensionality within the dataset. The results indicated that it is unnecessary to include all features in the learning model to detect attacks. By minimizing computational load and reducing the model's learning time, accuracy for attack detection was also enhanced.

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

EXTENDED ABSTRACT

Introduction

Computer networks are crucial for communication and data exchange. As these networks grow, so does the risk of cyberattacks. To effectively detect and mitigate these threats, intrusion detection systems (IDS) have become indispensable. However, the diversity of attacks presents a challenge for traditional IDS.

This research proposes a deep learning approach that combines dimensionality reduction and feature selection. Principal Component Analysis (PCA) was used to reduce the dimensionality of the data. Then, Grey Wolf Optimization (GWO) was employed to select the most informative features. These selected features were fed into a deep LSTM network for intrusion detection. The present research experiments on the NSL-KDD dataset demonstrated that this combined approach improves detection accuracy while reducing computational overhead. By focusing on the most relevant features, better results were achieved than when using all features.

Methodology

In the current research, initially, the PCA algorithm, an unsupervised linear algorithm, was used to reduce feature dimensionality. To this end, the initial 41 features were reduced to a smaller number (e.g., n=15 features) using PCA. Subsequently, a Long Short-Term Memory (LSTM) neural network was trained with the objective of classifying attacks using these 15 features. In the subsequent step, the Grey Wolf Optimization (GWO) algorithm, a nonlinear optimization algorithm that is not capable of varying the feature size, was introduced. In this case, GWO began its search among the 15 features and ultimately succeeded in extracting the top 6 features. The LSTM network from the previous stage was used as an evaluation function for the GWO algorithm. This process was repeated with specific step sizes of n=15, 20, 30. Ultimately, by selecting the best features, the convergence speed of the network in the training phase was improved, and consequently, the accuracy of intrusion detection was enhanced. The algorithm used in this research is illustrated in Figure 1. The workflow is described in detail as below.

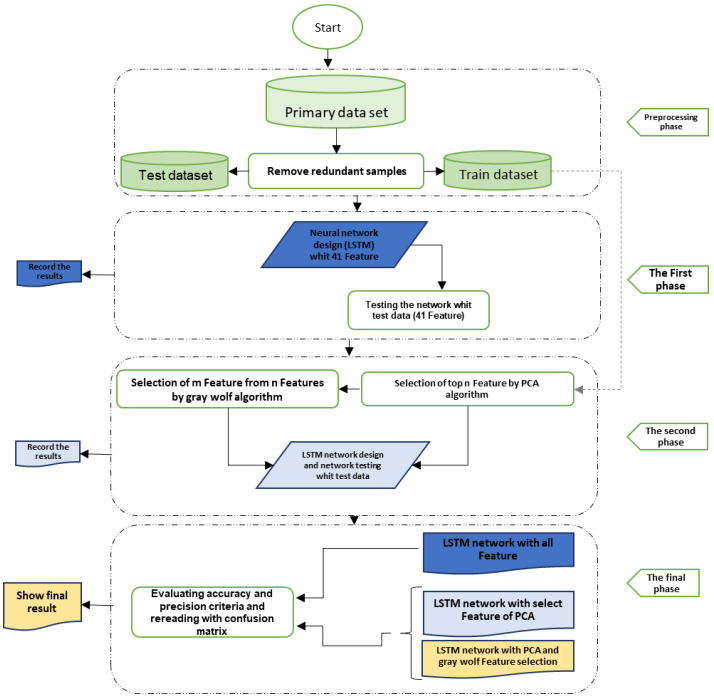


Figure 1. Proposed algorithm.

Preprocessing phase

The data used in this research was from the NSL-KDD dataset, which was collected by the IST group at the MIT Lincoln Laboratory as the first standard dataset for intrusion detection.

Designing an LSTM Network with 41 Features

In the initial phase of this research, a Long Short-Term Memory (LSTM) neural network was trained using 41 features. Subsequently, this network was employed to test the output data of the PCA algorithm and calculate the fitness function in the Grey Wolf Optimization algorithm.

Phase II: Dimensionality Reduction using PCA Grey Wolf Optimization

The less important or unimportant features were removed from the NSL-KDD dataset, and out of the existing 41 features, 15 features were identified as the most important. The Grey Wolf Optimizer (GWO) is a widely used metaheuristic algorithm inspired by the hunting behavior of grey wolves. After feature selection using PCA, an initial population of random numbers between 0 and 1 was generated to create the GWO algorithm. In the next step, to eliminate unimportant features and extract useful ones, the random numbers from the previous step were converted into binary form. If the threshold for feature selection is considered to be 0.5, it means that a feature with a weight of 0.5 is considered a selected feature. After calculating the fitness function, each row of the initial population (each wolf) was multiplied by all samples to activate relevant features

(1) and deactivate irrelevant features (0). The new dataset was applied to the LSTM1 network to calculate the MP value. TF is constant and equal to N, while SF varies for each wolf. The final alpha obtained from the iterations is considered the best wolf.

Results and discussion

Table 3 displays the accuracies resulting from each phase of the proposed method. It is noteworthy that the Grey Wolf Algorithm was iterated 15 times with weights of 0.99 and 0.01, respectively. To ensure the reliability of the calculations, each experiment was repeated 5 times, and the average accuracy was recorded in the table.

Table1:Training Data Report

ROW	Number of Selected Features	Normals that have been correctly identified	Attacks that have been correctly identified	Normals that have been identified as attacks	Normals that have been identified as attacks	Accuracy	precision	recall	F۱ Score	Gray Wolf Evaluation Function	time	composite criterion
۱	۴۱	۶۷۰۶۴	۵۵۴۷۶	۳۱۵۴	۲۷۹	٪ ۹۷.۳	٪ ۹۹.۵	٪ ۹۴۶	٪ ۹۶.۹۸	--	۲۲۸	۴۸.۴۹۲
۲	۲-۴۱	۶۶۰۱۰	۵۱۹۲۶	۶۷۰۴	۱۳۳۳	٪ ۹۳.۶	٪ ۹۷.۵	٪ ۸۸.۶	٪ ۹۲.۸۳	۱.۱۳۳۲	۲۲۲	۴۶.۴۱۷
۳	۳۰	۶۶۶۱۴	۵۵۲۰۸	۳۴۲۲	۷۲۹	٪ ۹۶.۷	٪ ۹۸.۷	٪ ۹۴.۲	٪ ۹۶.۳۹	--	۲۵۴	۴۸.۱۹۶
۴	۸-۳۰	۵۶۳۷۸	۵۷۰۲۴	۱۶۰۶	۱۰۹۶۵	٪ ۹۰۶	٪ ۸۳.۹	٪ ۹۷.۳	٪ ۹۰.۱۵	۱.۰۰۲۸	۳۳۰	۴۵.۰۷۸
۵	۲۵	۶۶۷۷۴	۵۶۱۴۴	۲۴۸۶	۵۶۹	٪ ۹۷۶	٪ ۹۹	٪ ۹۵.۸	٪ ۹۷.۳۷	--	۲۹۹	۴۸.۶۸۶
۶	۶-۲۵	۶۶۹۱۵	۵۵۸۱۹	۲۸۱۱	۴۲۸	٪ ۹۷.۴	٪ ۹۹.۲	٪ ۹۵.۲	٪ ۹۷.۲۵	۱.۰۰۲۰	۴۵۴	۴۸.۶۲۶
۷	۱۵	۶۶۷۶۷	۵۵۳۴۲	۳۵۸۸	۵۷۶	٪ ۹۶.۹	٪ ۹۹	٪ ۹۴.۴	٪ ۹۶.۶۴	--	۲۹۴	۴۸.۳۲۱
۸	۶-۱۵	۶۶۶۶۱	۵۵۶۶۹	۲۹۶۱	۶۸۲	٪ ۹۷.۱	٪ ۹۸.۸	٪ ۹۴.۹	٪ ۹۶.۸۱	۰.۹۸۵۳	۲۶۵	۴۸.۴۰۶
					Test Data Report							
۱	۴۱	۹۱۸۵	۷۴۱۱	۵۴۲۲	۵۲۵	٪ ۷۳.۶	٪ ۹۳.۴	٪ ۵۷.۷	٪ ۷۱.۳۳	--	۲۲۸	۳۵.۶۶۷
۲	۲-۴۱	۹۴۱۷	۸۰۳۰	۴۸۰۳	۲۹۳	٪ ۷۷.۴	٪ ۹۶.۵	٪ ۶۲.۶	٪ ۷۵.۹	۱.۱۳۳۲	۲۲۲	۳۷.۹۵۲
۳	۳۰	۹۰۷۵	۷۴۱۵	۵۴۱۸	۶۳۵	٪ ۷۳.۱	٪ ۹۲.۱	٪ ۵۷.۸	٪ ۷۱.۰۲	--	۲۵۴	۳۵.۵۱۱
۴	۸-۳۰	۸۹۸۱	۸۴۶۰	۴۳۷۳	۷۲۹	٪ ۷۷.۴	٪ ۹۲.۱	٪ ۶۵.۹	٪ ۷۶.۸۲	۱.۰۰۲۸	۳۳۰	۳۸.۴۱۱
۵	۲۵	۹۰۴۵	۷۸۵۸	۴۹۷۵	۶۶۵	٪ ۷۵	٪ ۹۲.۲	٪ ۶۱.۲	٪ ۷۳.۵۶	--	۲۹۹	۳۶.۷۸۱
۶	۶-۲۵	۹۲۸۶	۸۵۶۵	۴۲۶۸	۴۲۴	٪ ۷۹.۲	٪ ۹۵.۳	٪ ۶۶.۷	٪ ۷۸.۴	۱.۰۰۲۰	۴۵۴	۳۹.۲۰۱
۷	۱۵	۸۹۷۱	۷۸۸۲	۴۹۵۱	۷۳۹	٪ ۷۴.۸	٪ ۹۱.۴	٪ ۶۱.۴	٪ ۷۳.۴	--	۲۹۴	۳۶.۷۰۱
۸	۶-۱۵	۸۹۳۲	۸۵۵۳	۴۲۸۰	۷۷۸	٪ ۷۷.۶	٪ ۹۱.۷	٪ ۶۶.۶	٪ ۷۷.۱۶	۰.۹۸۵۳	۲۶۵	۳۸.۵۸۱

As Table 1 illustrates, the odd rows (1, 3, 5, 7) in both the training and testing data represent the case where only PCA was used for intrusion detection. However, in the even rows (2, 4, 6, 8), the Grey Wolf Algorithm was used to find the best features

Conclusion

Conformal mapping can calculate the capacitance of a parallel-plate capacitor including fringing effects. However, certain assumptions limit the accuracy of these calculations. Therefore, when applying the capacitive method to power transmission in medical implants, it is crucial to have precise information about the capacitance formed between two plates with the human muscle as a dielectric. This article presents a detailed formula with accurately adjusted parameters for capacitive power transfer applications in the field of medical implants. The correction parameters for the formula were derived through electromagnetic simulation using Ansys Maxwell software in the present study. Using the new formula reduced the computation error from over 40% to less than 16%.

کارافن
فصلنامه علمی دانشگاه ملی مهارت
بهار۱٤۰٤، دوره ۲۲، شماره ۱، ۱٤۹-۱۲۷
آدرس نشریه: https://karafan.nus.ac.ir
doi 10.48301/KSSA.2025.476687.2997

بهبود دقت تشخیص نفوذ با استفاده از روش ترکیبی PCA-GWO و شبکه عصبی عمیق

زهرا وکیل زاده [۱]، زهرا حیدران داروقه امنیه [۲]، ایمان ذباح [۳]، زینب بینایی [۴]
۱-دانش آموخته گروه برق و کامپیوتر، دانشگاه تربت حیدریه، تربت حیدریه، ایران.
۲- گروه برق، واحد دولت آباد، دانشگاه آزاد اسلامی، اصفهان، ایران.
۳- گروه برق و کامپیوتر، واحد تربت حیدریه، دانشگاه آزاد اسلامی، تربت حیدریه، ایران.
۴- دانشجوی دکتری، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان، دامغان ایران.

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۳/۰۷/۰۱
بازنگری مقاله: ۱۴۰۳/۰۸/۲۰
پذیرش مقاله: ۱۴۰۳/۱۱/۱۰

کلید واژگان:
تشخیص نفوذ
یادگیری عمیق
شبکه **LSTM**
تحلیل مولفه اصلی
الگوریتم گرگ خاکستری

***نویسنده مسئول:** ایمان ذباح
پست الکترونیکی:
imanzabbah@gmail.com

چکیده

شبکه‌های کامپیوتری نقش حیاتی در ارتباطات و تبادل داده‌ها دارند. با گسترش این شبکه‌ها، شرایط برای حملات سایبری و نفوذ بیشتر فراهم شده است. در دنیای واقعی، تغییرات مداوم در الگوهای ترافیک و ظهور تهدیدات جدید، نیاز به آموزش سریع و به‌روز مدل‌های تشخیص نفوذ را اجتناب‌ناپذیر می‌کند. نفوذ شامل فعالیت‌های غیرقانونی است که سلامت اطلاعات، محرمانگی و دسترسی به منابع سازمان را به خطر می‌اندازد. سیستم‌های تشخیص نفوذ به عنوان یکی از عوامل اصلی و مهم در امنیت شبکه، حملاتی را که توسط فایروال‌های سنتی شناسایی نمی‌شوند، رصد می‌کنند. با این حال، حملات مختلف رفتارهای خاص خود را دارند و بهبود تشخیص نوع حمله همچنان یکی از چالش‌های مدل‌های تشخیص نفوذ است. در این پژوهش، یک روش عمیق مبتنی بر کاهش ابعاد و انتخاب برترین ویژگی‌ها پیشنهاد شده است. ابتدا کاهش ابعاد توسط الگوریتم تحلیل مؤلفه اصلی انجام می‌شود، سپس ویژگی‌های برتر توسط الگوریتم گرگ خاکستری انتخاب شده و در نهایت ویژگی های کلیدی در تشخیص حمله بودن یا نبودن استخراج شده، و به شبکه عمیق بازگشتی حافظه دار اعمال شده است. فرایند یادگیری بر روی داده‌های NSL_KDD پیاده‌سازی شده است. یکی از جنبه‌های کلیدی این تحقیق، در ترکیب تحلیل مولفه اصلی و گرگ خاکستری و تجمیع قابلیت های هر کدام از این دو الگوریتم به منظور استخراج بهترین ویژگی‌ها و کاهش ابعاد در مجموعه داده‌ها است. نتایج نشان می‌دهد که برای تشخیص حملات، اعمال تمام ویژگی‌ها به مدل یادگیر الزامی نیست و با کاهش حجم بار محاسباتی، ضمن کاهش مدت زمان یادگیری مدل، دقت نیز در تشخیص حملات بهبود می‌یابد.

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

۱-مقدمه

افزایش استفاده از شبکه های کامپیوتری در حوزه های مختلف از یک سو و افزایش پیچیدگی و تعداد حملات سایبری در دنیای امروز، از سوی دیگر منجر شده است تا توسعه روش‌های مؤثر برای تشخیص و پیشگیری از نفوذ به شبکه‌ها و سیستم‌های اطلاعاتی مورد توجه محققین قرار گیرد. براساس نظر آماروسو[۱]تشخیص نفوذ عبارت است از نظارت بر وقایع رخ داده در یک شبکه، و یا سامانه جهت کشف موارد انحراف از سیاست های امنیتی و در نهایت پاسخ به فعالیت های مشکوک علیه منابع پردازشی و شبکه ای[1]. سامانه های تشخیص نفوذ (IDS[۲]) از حیث شیوه تحلیل و تشخیص غالبا به دو دسته ی سامانه های تشخیص سوءاستفاده و سامانه های تشخیص ناهنجاری تقسیم می شوند[2]. سامانه های تشخیص ناهنجاری ابتدا نمایه هایی از رفتارهای هنجار (یا نرمال) از سامانه ای که در آن مستقر است را تشکیل داده سپس هرگونه تخطی یا انحراف از نمایه هنجار بالاتر از یک حد آستانه را به عنوان رفتار ناهنجار و مهاجمانه تلقی می کنند[3]. پژوهش حاضر از نوع سامانه های تشخیص امضاء است. تشخیص نفوذ مبتنی بر امضاء ، با در اختیار داشتن مجموعه ای از الگوهای حمله در بین هشدارهای موجود به دنبال هشدار یا زنجیره ای از هشدارها می گردد که با یکی از الگوهای حمله مطابقت داشته باشد[2]. لذا این سامانه نیازمند مجموعه داده ای خواهد بود که در آن الگوهای حمله در دو بخش آموزش و آزمایش تعریف شده باشند و نیازمند بهنگام سازی مکرر پایگاه داده حملات در فواصل زمانی کوتاه است[4]. سیستم های تشخیص نفوذ (IDS) تحلیل الگو را با استفاده از فنون داده کاوی یا تحلیل آماری انجام می دهند. قبل از به کارگیری فنون داده کاوی لازم است آماده سازی داده ها انجام گیرد. یکی از مهم ترین مراحل آماده سازی داده های سامانه تشخیص نفوذ تعیین خصایص یا ویژگی هایی از داده ها است که براساس آن خصایص بتوان مراحل داده کاوی را به خوبی دنبال کرد[5]. در همین راستا فرایندی موسوم به انتخاب ویژگی یا کاهش ویژگی نیز وجود دارد که منجر به انتخاب مطلوب خصایص متناسب با هدف مورد نظر می گردد[4]. هرچه این انتخاب ، دقیق تر باشد تشخیص نفوذ موفق تر خواهد بود. ایده اصلی در پژوهش حاضر معرفی یک رویکرد نوین برای تشخیص نفوذ در شبکه، با استفاده از ترکیب الگوریتم گرگ خاکستری(GWO[۳]) و تحلیل مؤلفه‌های اصلی (PCA) است. از آنجا که PCA یک روش خطی و گرگ خاکستری یک الگوریتم غیرخطی است، ترکیب این دو الگوریتم امکانی را فراهم می کند که هر یک محدودیت های دیگری را پوشش دهد. در این روش ابتدا با استفاده از PCA به کاهش ابعاد داده‌ها پرداخته می‌شود تا ویژگی‌های کمتری که دارای اطلاعات کلیدی هستند شناسایی شود. سپس، الگوریتم گرگ خاکستری برای بهینه‌سازی و انتخاب بهترین ویژگی‌ها از مجموعه داده‌های کاهش‌یافته به کار گرفته می‌شود. این رویکرد نه تنها پارامتر تعداد ویژگی‌ها را تعیین می‌کند، بلکه به شناسایی و استخراج ویژگی‌های مؤثر برای بهبود دقت مدل‌های طبقه‌بندی عمیق کمک می‌کند. نتایج نشان می دهد که عملکرد مدل بعد از کاهش توسط PCA و گرگ خاکستری بهبود می یابد.در این پژوهش در بخش ۲ به مرور کارهای گذشته می پردازیم و پس از توضیح روش پیشنهادی و فاز بندی اجرای آن در بخش سوم ، به ارزیابی نتایج در بخش چهارم خواهیم پرداخت. در نهایت مقایسه روش پیشنهادی با پژوهش های دیگران انجام شده و پیشنهادات آتی مطرح می گردد.

---

[۱] Amarso

[۲] Intrusion Detection System

[۳] Grey Wolf Optimizer

۲- مرور کارهای گذشته

استفاده از تکنیک های یادگیری ماشین در حوزه تشخیص نفوذ، مبتنی بر ساختن مدلی است که توسط آموزش روی مجموعه داده های قبلی به دست می آید. چنین مدلی قابلیت تعمیم روی نمونه های بعدی را دارد و می تواند با کارایی بالایی برای دسته بندی نمونه های جدید مورد استفاده قرار بگیرد[6].

مانند بسیاری از سیستم های یادگیر رایج، در مدل های تشخیص نفوذ دو روش اصلی از یادگیری ماشین وجود دارد: یادگیری نظارت شده[۱] و یادگیری بدون نظارت.[۲] یادگیری نظارت شده بر روی اطلاعات مفید موجود در داده‌های برچسب‌خورده تکیه دارد. مدل های طبقه‌بندی معمولا رایج‌ترین مدل ها در یادگیری های نظارت شده هستند که در سیستم های IDS بیشتر مورد استفاده قرار می گیرند[7]. از نگاهی دیگر، الگوریتم‌های معمول یادگیری ماشینی استفاده شده در سیستم‌های تشخیص نفوذ به دو دسته کلی تقسیم می شوند. یکی روش های سنتی و دیگری روش های مبتنی بر یادگیری عمیق. مدل‌های سنتی یادگیری ماشین (مدل‌های کم عمق) برای سیستم‌های تشخیص نفوذ اصلی شامل شبکه‌های عصبی مصنوعی[۳](ANN)[8]، ماشین بردار پشتیبان (SVM)،[۴] [9] روش نزدیک‌ترین همسایه (KNN)[۵]، [10]، روش های مبتنی بر رگرسیون لجستیک[۶](LR) [11]، درخت تصمیم [12] و همچنین روش‌های ترکیبی [13] هستند. برخی از این روش‌ها برای چند دهه در مدل های تشخیص نفوذ، مورد مطالعه قرار گرفته‌اند.

در سال های اخیر مدل‌های یادگیری عمیق در تشخیص نفود مورد توجه بسیاری از محققین حوزه تشخیص نفوذ قرار گرفته است. از میان آن‌ها، شبکه‌های عمیق برای آموزش سریع (DBNs) [14] شبکه‌های عصبی عمیق (DNNs) [14] شبکه‌های عصبی کانولوشنی (CNNs) [15] و شبکه‌های عصبی بازگشتی (RNNs) [16] مدل‌های یادگیری نظارت‌شده هستند، که به ارائه مدل های یادگیر در تشخیص نفوذ پرداخته اند. اتوانکودرها، ماشین‌های بولتزمن محدود (RBMs) [17] و شبکه‌های مخالف مولد (GANs) [6] مدل‌های یادگیری بدون نظارت می باشند. تعداد مطالعات مبتنی بر یادگیری عمیق در زمینه سیستم‌های تشخیص نفوذ (IDSs) از سال ۲۰۱۵ به طور فزاینده ای گسترش یافته است.

در مطالعات عمیق، در حوزه تشخیص نفوذ تأکید اصلی بر روی معماری شبکه، انتخاب هایپرپارامتر و استراتژی بهینه‌سازی است. به عنوان مثال در پژوهش [18] تشخیص نفوذ با روش یادگیری شبکه عصبی عمیق پیشنهاد شده است . در این پژوهش که از مجموعه داده CICIDS-2017 استفاده شده است ، بهترین نتایج در روش DNN با ۵ لایه به دست آمده است. لو و همکاران به منظور بهبود نرخ تشخیص و کاهش نرخ خطا یک روش جدید تشخیص نفوذ ترکیبی از شبکه عصبی کانولوشن و اصلاح آستانه بر اساس منحنی مشخصه عملکرد ارائه دادند. در این روش از شبکه عصبی کانولوشن به عنوان طبقه بند استفاده می شود و آستانه طبقه بند از طریق منحنی اصلاح می شود[19]. در پژوهش [20] از یک شبکه عصبی عمیق بر روی مجموعه داده KDD-99 و دو مجموعه داده دیگر به صورت باینری استفاده شده است که در نهایت مقدار صحت ۰,۹۶۰۳گزارش شده است. لازم به ذکر است که این پژوهش توانایی تشخیص نفوذ و عدم نفوذ را در سیستم داشته و نوع حمله در آن تشخیص داده نمی شود. در پژوهش های متعدد،

---

[۱] Supervised Learning

[۲] Unsupervised Learning

[۳] Artificial Neural network

[۴] Support Vector Machine

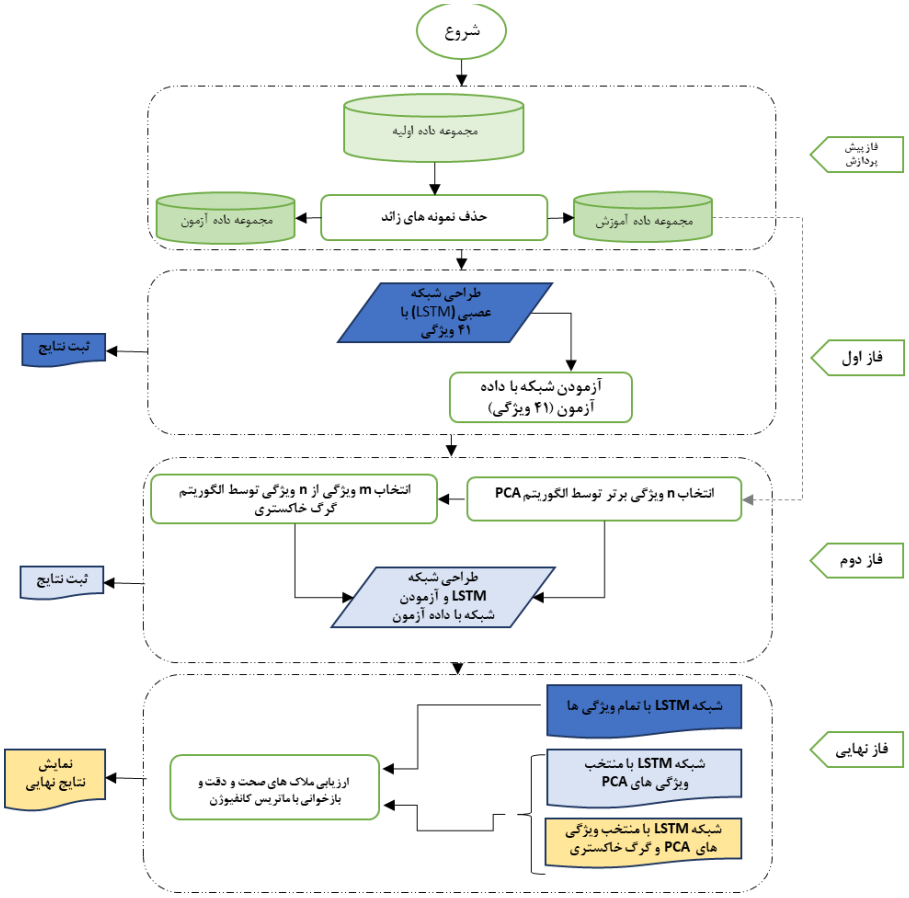
[۵] k-nearest neigbors algorithm

[۶] Logistic regression

علاوه بر استفاده از یادگیری عمیق به عنوان یک خبره، به منظور همگرایی نظرات خبرگان، از روش های ترکیب خبره ها نیز به عنوان یک متد شناخته شده [13] استفاده شده است. رأی دهی اکثریت وزن دار WMV1 یکی از متدهایی است که در پژوهش [21] مورد توجه قرار گرفته است. در این پژوهش نشان داده شده است که الگوی یادگیری گروهی مبتنی بر الگوریتم PSO عملکرد مطلوبی را از خود نشان داده و دقت تشخیص به صورت تقریبی ۹۳ درصد گزارش شده است. استفاده از الگوریتم های فرا ابتکاری نیز در تشخیص حملات و طراحی بهینه سیستم های تشخیص نفوذ مورد توجه محققین بوده است. به عنوان نمونه می توان به پژوهش [22] اشاره کرد که نویسندگان از الگوریتم بهینه سازی گله فیل-بسط تیلور مبتنی بر شبکه باور عمیق بر روی دیتا ست KDD استفاده کرده اند. دو دیتاست دیگر ارزیابی شده است و با دقت ۹۳،۸۱ درصد توانسته حملات DDoS را تشخیص دهد. در پژوهشی دیگر از الگوریتم شیر به منظور انتخاب بهترین زیر مجموعه ویژگی و از طبقه بند CNN به منظور رده بندی استفاده شده است که در مقایسه با الگوریتم کلونی زنبور عسل و کلونی مورچگان عملکرد بهتری داشته است. دقت بدست آمده در تشخیص حمله DDoS برابر با ۹۶ درصد است [23]. در پژوهشی دیگر از الگوریتم بهینه سازی DDAO که با استفاده از تابع های مختلف ارزیابی شده است ، با انتخاب ۱۰ ویژگی از میان ۴۱ ویژگی و با دقت ۹۴ درصد حملات از غیر حملات تشخیص داده شده است [24]. در پژوهش دیگر به دلیل بالا بودن قدرت جستجوی الگوریتم های فراابتکاری از سه الگوریتم کلونی زنبور عسل (ABC) الگوریتم کلونی مورچگان(ACO) و ازدحام ذرات (PSO) برای انتخاب ویژگی و از الگوریتم های ماشین بردار پشتیبان (SVM) و نزدیک ترین همسایه (KNN) به منظور رده بندی استفاده شده است[25].

۳- روش پیشنهادی

سیستم های تشخیص نفوذی که با رویکرد عادی یادگیری ارائه می شوند، غالبا با مشکل انتخاب ویژگی صحیح یا کاهش سرعت تشخیص نفوذ در اثر ازدیاد ویژگی های بلااستفاده مواجه هستند [27]. در این پژوهش ابتدا از الگوریتم PCA که یک الگوریتم خطی بدون ناظر است به منظور کاهش بعد ویژگی ها استفاده شده است. بدین منظور تعداد ۴۱ ویژگی اولیه توسط PCA به ویژگی های کمتری (مثلا n=15 ویژگی) کاهش می یابد. سپس یک شبکه عصبی حافظه طولانی کوتاه مدت LSTM با هدف تشخیص کلاس حمله و با ۱۵ ویژگی آموزش داده می شود. درگام بعدی الگوریتم گرگ خاکستری به عنوان یک الگوریتم بهینه ساز غیر خطی [26] که قابلیت تغییر در تعداد سایز ویژگی ها را ندارد ، وارد عمل می شود. در این حالت گرگ خاکستری از بین ۱۵ ویژگی ، شروع به جستجو کرده و در نهایت موفق به استخراج ۶ ویژگی برتر می شود. از شبکه LSTM مرحله قبل به عنوان تابع ارزیاب الگوریتم گرگ خاکستری استفاده می شود. این عملیات با گام پرش مشخصی n=15,20,30 تکرار می شود. تا در نهایت با انتخاب برترین ویژگی ها سرعت همگرایی شبکه در فاز آموزش و در نهایت دقت تشخیص نفوذ بهبود پیدا کند. الگوریتم مورد استفاده در این پژوهش در شکل ۱ نمایش داده شده است. فرآیند کار به تفصیل به شرح زیر می باشد.



شکل ۱: الگوریتم مورد استفاده در یک نگاه

۳-۱- فاز پیش پردازش دادگان

داده‌های مورد استفاده در این پژوهش از مجموعه داده NSL_KDD ، می باشد [26]، که توسط گروه IST[1] از آزمایشگاه MIT LINCLON به عنوان اولین داده‌های استاندارد برای تشخیص نفوذ جمع‌آوری‌شده‌اند. (این داده ها شامل سه مجموعه مستقل است. قسمت اول "کل نمونه‌ها": مشتمل بر ۴۸۹۸۴۳۱ نمونه، قسمت دوم: "۱۰٪ نمونه‌ها" بعبارتی تعداد ۴۹۴۰۲۱ به‌عنوان "نمونه‌های استاندارد" و شامل تمامی انواع حملات؛ و قسمت سوم: شامل ۶٪ نمونه ها"نمونه‌های اصلاح‌شده" شامل ۳۱۱۰۲۹ رکورد می باشد. که از این تعداد ۴۰ درصد بعبارتی ۱۲۵۹۷۳ مورد جهت داده های آموزش و حدود ۱۰ درصد بعبارتی ۲۲۵۴۳ مورد جهت داده های آزمون در نظر گرفته شده است.) اکثر محققان این نمونه‌ها را به‌عنوان مجموعه آموزشی و آزمون استفاده نموده اند. برتری مجموعه داده NSL_KDD نسبت به KDD_CUP99 به این جهت است که شامل رکوردهای تکراری در مجموعه داده یادگیری و تست نمی باشد و از همگرا شدن شبکه عصبی نسبت به رکوردهای تکراری جلوگیری می کند [28]. در این مجموعه هر نمونه داده دارای

[1] Information Systems & Technology

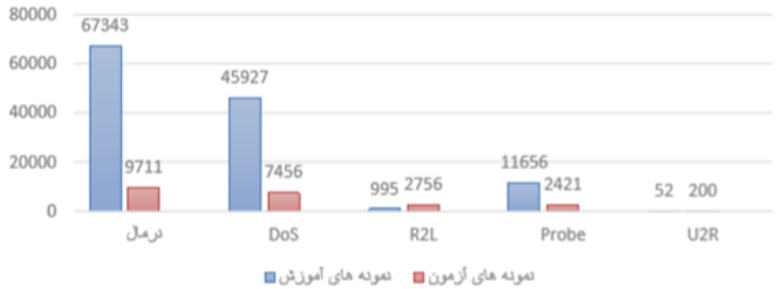
۴۱ ویژگی است که ۳۸ تای آنها دارای مقادیر عددی و ۳ تای آنها دارای مقادیر غیر عددی می باشند که در جدول ۱ نمایش داده شده است.

جدول۱: ۴۱ ویژگی حمله مجموعه داده NSL_KDD

نام ویژگی	شرح و توضیحات
Duration	زمان برقراری ارتباط برحسب ثانیه
Protocol_type	نوع پروتکل، مانند:TCP,UDP,..
Service	سرویس درخواستی ارتباط، مانند: *http،telnet*
Src_bytes	مقدار اطلاعات جابه‌جاشده از مبدأ به مقصد
dst_bytes	مقدار اطلاعات جابه‌جاشده از مقصد به مبدأ
Flag	پرچم‌ها مشخص‌کننده وضعیت خطا در ارتباط
Land	در صورت برابری درگاه مبدأ و مقصد یک و در غیر اینصورت صفر
Wrong_fragment	تعداد خطا در قطعه‌ها
Urgent	تعداد بسته‌های ضروری در طول ارتباط
Hot	تعداد نشانه‌های فعالیت مشکوک
Root_shell	با ۰و یا ۱شدن وضعیت *root shell* را مشخص می‌کند
Su_attempted	با۰و یا ۱شدن وضعیت *su root* را مشخص می‌کند
Num_root	اعداد دسترسی‌هایی که به *root* انجام‌گرفته است
Num_file_creations	تعداد فایل‌های عملیاتی ایجاد شده
Num_shells	تعداد هسته‌های آماده
Num_access_files	تعداد عملیات روی فایل‌های کنترل دستیابی
Num_outbound_cmds	تعداد دستورات خارج شده در نشست FTP
Is_host_login	با 0و یا 1شدن مشخص می‌کند که آیا *login* عضو لیست *hot* است یا نه
Is_guest_login	با 0و یا 1شدن وضعیت *guest* بودن *login* را مشخص می‌کند
Num-failed-logins	تعداد *login* های دارای نقص
Count	تعداد اتصالاتی که از یک *host* در یک اتصال جاری بیش از ۲ ثانیه طول بکشد
Serror_rate	درصد اتصالاتی که اشکال *SYN* دارند
Rerror_rate	درصد اتصالاتی که اشکال *REJ* دارند
Same_srv_rate	درصد اتصالاتی به سرویس‌های یکسان
Diff_srv_rate	درصد اتصالاتی به سرویس‌های مختلف
Srv_count	تعداد اتصالاتی که از یک سرویس در یک اتصال جاری بیش از ۲ ثانیه طول بکشد
Srv_serror_rate	درصد اتصالاتی که اشکال *SYN* در سرویس دارند

Srv_rerror_rate	درصد اتصالاتی که اشکال *REJ* در سرویس دارند
Srv_diff_host_rate	درصد اتصالاتی به *host* های مختلف
Dst_host_count	تعداد *host* های مقصد
Dst_host_srv_count	تعداد سرویس *host* های مقصد
Dst_host_same_srv_rate	درصد اتصالاتی که از یک *host* با یک سرویس به یک *host* مقصد در یک بازه زمانی انجام‌شده است
Dst_host_diff_srv_rate	درصد اتصالاتی که از یک *host* با سرویس‌های مختلف به یک *host* مقصد در یک بازه زمانی انجام شده است
Dst_host_same_src_port_rate	درصد اتصالاتی که از یک *host* با یک پورت منبع انجام شده است
Dst_host_srv_diff_host rate	درصد اتصالاتی که از یک *host* به *host* دیگر با سرویس متفاوت انجام شده است
Dst_host_serror_rate	نرخ اشکالات SYN در host منبع
Dst_host_srv_serror_rate	نرخ اشکالات *SYN* سرویس *host* منبع
Dst_host_rerror_rate	نرخ اشکالات host منبع
Dst_host_srv_rerror_rate	نرخ اشکالات سرویس *host* منبع

مجموعه داده ی اولیه دارای ۵ دسته بندی برای تشخیص نفوذ بوده است که در شکل ۲ نمایش داده شده است. وجود دیتا ست نامتوازن یکی از چالش هایی است که محققین حوزه داده کاوی با آن مواجه هستند[29]. و راهکارهای متعددی در این خصوص پیشنهاد شده است.



شکل ۲: تقسیم بندی نمونه های آموزش و آزمون در حملات پایگاه داده NSL-KDD

۳-۳- فاز اول طراحی شبکه LSTM با ۴۱ ویژگی

شبکه مبتنی بر حافظه کوتاه مدت ماندگار (LSTM) یکی از انواع شبکه های عصبی بازگشتی RNN می باشد که توانایی یادگیری وابستگی های بلندمدت را دارد. شبکه LSTM برای اولین بار توسط هاکریتر و اشمیدبر در سال ۱۹۹۷ جهت حل مشکل بخاطر سپاری داده ها معرفی شدند [30] . شبکه های عصبی بازگشتی نوع توسعه یافته ی شبکه عصبی ساده است که از لایه هایی از نرونها تشکیل گردیده و مانند شبکه های عصبی بازگشتی ساده ، اتصالات بازگشتی دارد. تفاوت این شبکه با شبکه های عصبی ساده در تعداد لایه هاست.

در این شبکه به جای یک لایه شبکه عصبی ، چهار لایه وجود دارد و این لایه ها با روشی کاملا مشخص در تعامل با یکدیگر هستند. شـبکه عصبی بازگشتی LSTM دارای ۴ گیت فراموشی ، یادآوری ، یادگیری و خروجی می باشد. همچنین دارای ۳ ورودی حافظه ی بلند مدت ، حافظه کوتاه مدت ، نمونه آموزشی و داده ی جدید است. عملکرد شبکه عصبی بازگشتی LSTM در دو گام صورت می گیرد. اولین گام تصمیم گیری نسبت به اینکه چه اطلاعاتی را نمیخواهیم و باید از وضعیت سلول خارج شوند، است. در ساختار شبکه LSTM لایه ی گیت فراموشی این تصمیم گیری را راجع به کنترل حافظه و اطلاعات به عهده دارد.

در رابطه ی (۱) ، $\boldsymbol{h_{t-1}}$ و $\boldsymbol{X_t}$ برای هر عدد در وضعیت سلول $\boldsymbol{C_{t-1}}$ عددی بین صفر و یک را نشان می دهد. عدد یک نشان دهنده ی «به طور کامل نگه دار» و صفر به معنای «نگه ندار» است.

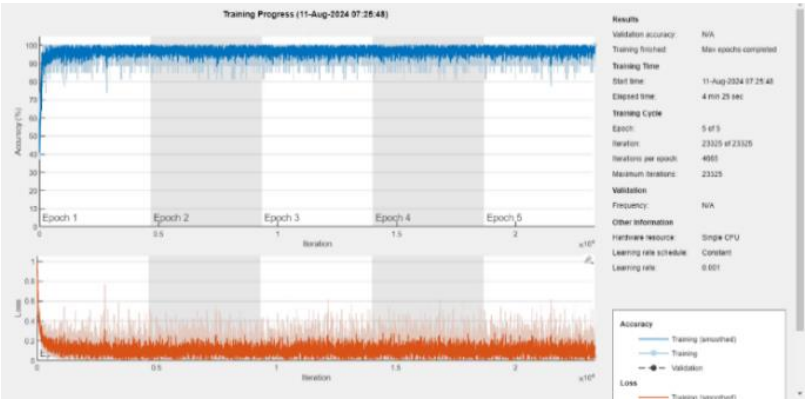
$$\boldsymbol{F_t = \sigma\left(W_f \times [h_{t-1}, X_t] + b_f\right)} \quad (۱)$$

در گام بعدی اطلاعات جدید در وضعیت سلول ذخیره می شود. این گام از دو بخش تشکیل شده است. در بخش اول یک لایه سیگموئید به نام لایه ی گیت ورودی تعیین می کند که میخواهیم کدام مقادیر را به روزرسانی کنیم.(رابطه۲) در بخش دوم یک لایه tanh برداری از مقادیر کاندید شده ی جدید ساخته می شود. این لایه می تواند به وضعیت سلول اضافه شود. (رابطه۳) در نهایت با ترکیب این دو بخش می توان در سلول اطلاعاتی افزود یا به روزرسانی کرد.

$$\boldsymbol{i_t = \sigma(W_i \times [h_{t-1}, X_t] + b_i)} \quad (۲)$$

$$\boldsymbol{\tilde{C_t} = tanh(W_c \times [h_{t-1}, X_t] + b_c)} \quad (۳)$$

در فاز اول این پژوهش، شبکه عصبی LSTM با ۴۱ ویژگی آموزش داده شده و سپس از این شبکه برای آزمودن داده های خروجی الگوریتم PCA و محاسبه ی تابع برازندگی در الگوریتم گرگ خاکستری استفاده شده است. شکل 3 فرایند طی شدن آموزش را نشان می دهد.



شکل ۳ : فرایند آموزش شبکه عمیق LSTM با ۴۱ ویژگی

۳-۳- فاز دوم : کاهش بعد توسط PCA

همانطور که پیش تر اشاره شد دیتاست مورد استفاده (NSL_KDD) در این پژوهش داری ۴۱ ویژگی است که این ویژگی ها بر اساس اهمیت شان توسط الگوریتم PCA مرتب می شوند. هدف و انگیزه از این کار کاهش ابعاد یک مجموعه داده با d بعد با طرح‌ریزی آن در یک زیر فضا با k بعد (که در آن k کوچکتر از d است) و به منظور بالا بردن بازدهی محاسباتی و به طوری است که بخش مهم و حیاتی اطلاعات باقی بماند. به عبارتی وظیفه اصلیِ PCA در کم

کردن ویژگی بر اساس واریانس داده ها است[31]. PCA ، می‌تواند مؤلفه‌های اصلی را شناسایی کند تا به جای بررسی همه ی ویژگی ها فقط یک سری ویژگی‌های مهم تر و با ارزش تر بررسی شوند. پس از دریافت مجموعه داده ی آموزشی NSL_KDD، نمایش مجموعه داده Train و Test در یک ساختار صورت می پذیرد. هر سطر مربوط به یک نمونه از داده ها و هر ستون مربوط به مقادیر یک ویژگی است. در مرحله بعد استاندارد سازی داده ها در یک ستون از داده های آموزش و تست انجام می شود و ویژگی هایی با واریانس بالا در مقایسه با ویژگی هایی با واریانس کمتر انتخاب می شوند و محاسبه ماتریس کوواریانس انجام می شود. سپس، مقادیر ویژه و بردارهای ویژه محاسبه و مرتب‌سازی می‌شوند. مقادیر ویژه به صورت نزولی مرتب شده و بردارهای ویژه بر اساس نتایج مرتب‌سازی مقادیر ویژه در ماتریس P قرار می‌گیرند. در نهایت، ویژگی‌های کم‌اهمیت یا بی‌اهمیت از مجموعه داده NSL_KDD حذف می‌شوند و از میان ۴۱ ویژگی موجود، ویژگی های مهم تر به عنوان ویژگی‌های پر اهمیت شناخته می‌شوند.

۳-۴- فاز دوم : استخراج ویژگی های برتر توسط گرگ خاکستری

الگوریتم گرگ خاکستری یکی از الگوریتم های فرابتکاری پر کاربرد است که از نحوه حملهٔ گرگ ها در زمان شکار الهام می گیرد. گرگ های خاکستری در زندگی اجتماعی دارای سلسله مراتب می باشند. جفت آلفا در آن ها به عنوان رهبر گروه شناخته می شوند و تصمیم گیری درباره ی شکار و دیگر موارد را بر عهده دارند. دومین رده در سلسله مراتب یک دسته، متعلّق به گرگ های بتا است. این گرگ ها در تصمیم گیری ها و سایر فعالیت های دسته به گرگ های آلفا کمک می کنند. پایین ترین مقام متعلق به گرگ های امگا است که نقش پیش مرگ را در دسته بازی می کنند. به گرگ هایی که در سلسله مراتب بالا ذکر نشده است، گرگ های دلتا گفته می شود. گرگ های دلتا تحت فرمان آلفا و بتا بوده، ولی نسبت به امگا برتری دارند[32]. در این پژوهش به منظور الگو کردن سلسله مراتب اجتماعی گرگ های خاکستری، بهترین پاسخ به عنوان گرگ آلفا در نظر گرفته می شود. همچنین دومین و سومین پاسخ مناسب را پس از آلفا، بتا و دلتا می نامیم. سایر پاسخ ها در گروه امگا قرار می گیرند. این فرآیند با استفاده از کاهش بردار a الگو می شود. برای پیاده سازی الگوی فرآیند الگوریتم گرگ خاکستری (ساز و کار شکار) از روابط زیر استفاده می کنیم :

$$\vec{D} = | \vec{C} \overrightarrow{X_p}(t) - \overrightarrow{X}(t) | \quad (۴)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \vec{D} \quad (۵)$$

که t بیانگر شماره ی تکرار است و از آنجا که A برداری تصادفی است ، با کاهش a ، بردار ضرایب A هم کاهش می یابد. اگر A<1 باشد ، گرگ آلفا و سایر گرگ های گله به شکار نزدیک می شوند و اگر A>1 باشد گرگ ها از شکار دور می شوند. در رابطه ی ذکر شده بردار C به عنوان موانع موجود در طبیعت که نزدیک شدن گرگ ها را به شکار کند میکنند، در نظر گرفته می شود. بردار C به شکار وزن داده و آن را برای گرگ ها غیرقابل دستیابی تر میکند. این بردار برخلاف a به صورت خطی کاهش نمی یابد. Xp بردار مکان طعمه و x بردار مکان گرگ خاکستری است. بردارهای A و C (بردارهای ضریب) به صورت زیر محاسبه می شوند :

$$\vec{A} = 2\vec{a} \overrightarrow{r_1} - \vec{a} \quad (۶)$$

$$\vec{C} = 2 \overrightarrow{r_2} \quad (۷)$$

که a به صورت خطی و در طی تکرارها از مقدار دو به صفر کاهش می یابد و r_1 و r_2 بردارهای تصادفی در بازه ی ۰ و ۱ هستند که در هر مرحله مرتباً تولید می شوند. برای شبیه سازی نحوه ی شکار گرگ های خاکستری فرض می کنیم که آلفا ، بتا و دلتا اطلاعات بهتری نسبت به موقعیت شکار دارند. در نتیجه سه پاسخ برتر کسب شده تا بدین جای کار را ذخیره کرده و سایر گرگ ها را وادار می کنیم تا موقعیت خود را با توجه به این سه پاسخ برتر به روزرسانی کنند. روابط زیر این به روز رسانی را نشان می دهند[33].

$$\overrightarrow{D_\alpha} = \left| \overrightarrow{C_1} \overrightarrow{X_\alpha} - \overrightarrow{X} \right|, \ \overrightarrow{D_\beta} = \left| \overrightarrow{C_2} \overrightarrow{X_\beta} - \overrightarrow{X} \right|, \ \overrightarrow{D_\delta} = \left| \overrightarrow{C_3} \overrightarrow{X_\delta} - \overrightarrow{X} \right| \quad (۸)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} . \left(\overrightarrow{D_\alpha}\right), \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} . \left(\overrightarrow{D_\beta}\right), \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} . \left(\overrightarrow{D_\delta}\right) \quad (۹)$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \quad (۱۰)$$

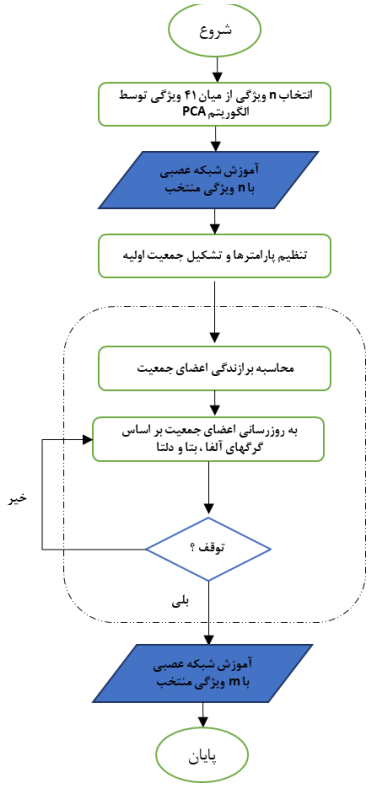
همانطور که ذکر شد پس از کاهش بعد توسط PCA به N ویژگی، شبکه عصبی LSTM1 آموزش داده می شود. سپس با ایجاد جمعیت اولیه به صورت تصادفی ، هر بار یک مجموعه از N ویژگی انتخاب شده و پس از آن در هر مرتبه محاسبه برازندگی اعضای جمعیت، توسط گرگ خاکستری صورت می پذیرد. جهت محاسبه برازندگی عامل ها در روش پیشنهادی از رابطه ی زیر استفاده می کنیم :

$$\boldsymbol{Fitness(Agent_i) = W_1 \times MP + W_2 \times \frac{TF}{SF}} \quad (۱۱)$$

همانطور که رابطه (۱۱) نشان می دهد ، برای هر یک از عامل های مؤثر در تابع ارزیابی، وزنی در نظر گرفته شده است w_1 و w_2 به ترتیب میزان اهمیت پارامتر اول و دوم می باشد. به طوری که $w_1+w_2=1$ خواهد بود و میتوان آن را با توجه به میزان اهمیت عامل تنظیم کرد. در این رابطه TF و SF به ترتیب تعداد کل ویژگی ها و تعداد ویژگی های انتخاب شده می باشد. به این ترتیب اگر این ترم از رابطه اهمیت بیشتری داشته باشد می توانیم مقدار w_2 را بیشتر انتخاب کنیم مثلا $w_2=0.9$ و $w_1=0.1$ و در صورتی که ترم اول رابطه اهمیت بیشتری داشته باشد این مقدار دهی می تواند برعکس باشد. و مقدار MP همان نتیجه ی بدست آمده از شبکه عصبی است که از رابطه ی (12) محاسبه می شود :

$$\boldsymbol{Acuracy = \frac{Correct\ Detect\ Sample}{Total\ Sample}} \quad (12)$$

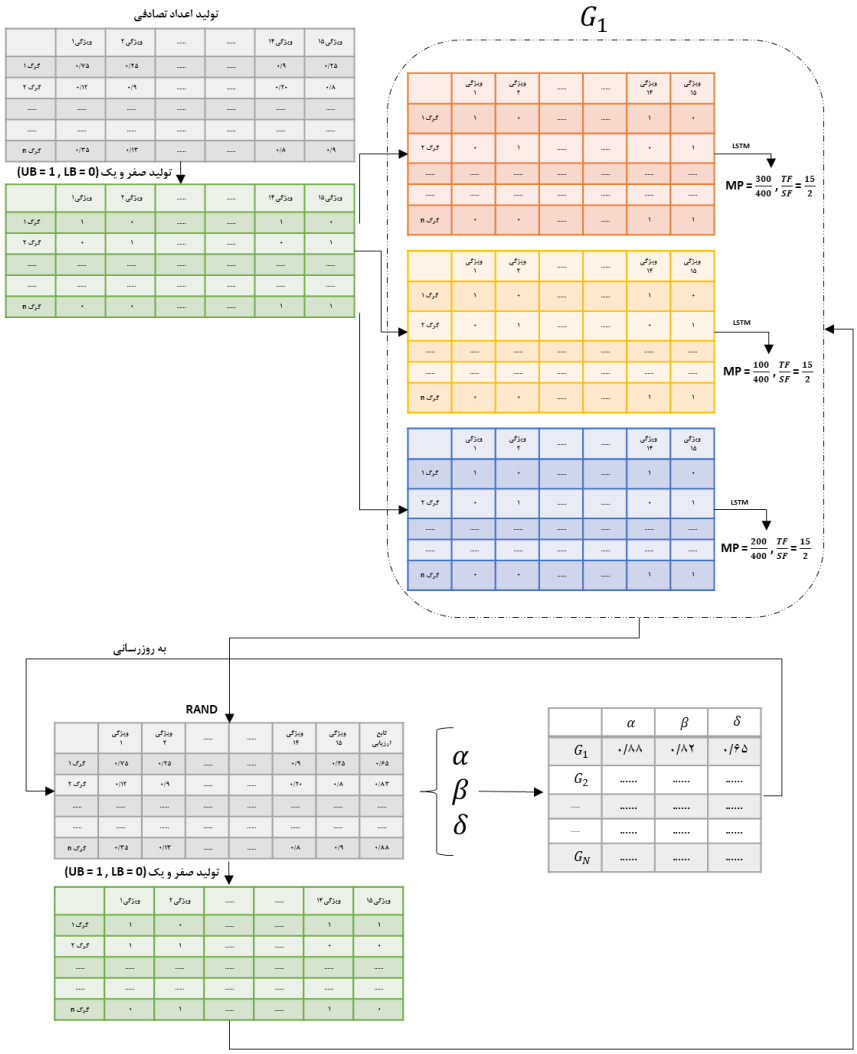
Correct Detect Sample بیانگر تعداد نمونه های درست و Total Sample بیانگر کل ویژگی ها است.
در شکل زیر روند کلی روش پیشنهادی مندرج در فاز دوم (در شکل ۱) به تفصیل نشان داده شده است.



شکل ۴ : روند انتخاب برترین ویژگی ها توسط الگوریتم PCA و الگوریتم گرگ خاکستری

همانطور که شکل ۴ نشان می دهد، پس از انتخاب ویژگی توسط PCA برای ایجاد الگوریتم گرگ خاکستری، ابتدا مجموعه ای از اعداد تصادفی به عنوان جمعیت اولیه تولید می شوند که در بازه ی صفر تا ۱ قرار می گیرند. در واقع طول هر عامل(گرگ) برابر N می باشد که در هر یک از آن ها تعداد M ویژگی انتخاب شده است. هر عدد مربوط به یک ویژگی و نشان دهنده درجه اهمیت آن ویژگی است. هر چه این عدد بیشتر باشد، شانس انتخاب آن ویژگی بیشتر می شود. در مرحله بعد به منظور حذف ویژگی های کم اهمیت و استخراج ویژگی های مفید، اعداد تصادفی مرحله قبل باید به صورت دودویی (یک = انتخاب ویژگی ، صفر = عدم انتخاب ویژگی) تبدیل شوند. اگر حد آستانه در انتخاب ویژگی ۰,۵ در نظر گرفته شود به این معناست که ویژگی دارای وزن ۰,۵ بعنوان ویژگی انتخابی در نظر گرفته شده است. پس از آن برای محاسبه تابع ارزیابی هر سطر از مجموعه اولیه (هر گرگ) را در تمام نمونه ها ضرب کرده تا ویژگی های مفید، فعال (۱) و ویژگی های غیرمفید نیز غیرفعال (۰) شوند. مجموعه داده ی جدید به شبکه عصبی LSTM1 اعمال می شود تا مقدار MP محاسبه شود. مقدار TF ثابت و برابر N و مقدار SF در هر گرگ متغیر است. پس از محاسبه تابع برازندگی (با استفاده از رابطه ۱۱) تمامی گرگ ها و تشخیص گرگ های دسته ی آلفا، بتا و دلتا، مقادیر مربوط به جمعیت اولیه به روزرسانی شده و روند قبلی برای این مرحله تکرار خواهد شد و آخرین آلفای حاصل از تکرار به عنوان برترین گرگ در نظر گرفته خواهد شد.

شکل ۵ نحوه تولید جمعیت اولیه و فرایند تکرار هر مرحله به منظور کشف برترین ویژگی های مندرج در شکل ۴ را نشان می دهد.



شکل ۵: چرخه تولید نسل و محاسبه برازندگی توسط گرگ خاکستری

از آنجا که یکی از مشکلات اصلی در الگوریتم گرگ های خاکستری ایجاد توازن بین اکتشاف و استخراج است، شاخص a به عنوان شاخصی برای مدیریت اکتشاف و استخراج در نظر گرفته شده است. این شاخص یک رفتار خطی دارد به صورتیکه در آغاز اجرای الگوریتم تمایل به اکتشاف و در تکرارهای آخر میل به استخراج دارد. از این رو ، رابطه زیر برای به روزرسانی این متغیر در هر تکرار استفاده می شود.

$$\boldsymbol{a = 2 - it \times \left(\frac{2}{MAXIT}\right)} \qquad (۱۳)$$

که در آن it به معنای شماره تکرار و maxit بیشترین تکرار می باشد.

۵- فاز سوم : ارزیابی روش پیشنهادی

در این پژوهش سعی شده است با ارائه یک راه حل ترکیبی و مبتنی بر شبکه عصبی عمیق ، گامی در جهت ایجاد افزایش امنیت در شبکه های کامپیوتری برداشته و با تعداد ویژگی های کمتری از مجموعه داده NSL-KDD به تشخیص نفوذ بپردازیم. روش پیشنهادی توسط نرم افزار متلب ۲۰۲۳ بر روی سیستمی با پردازنده Intel Core i5 ، ۱۲ گیگابایت رم ، سیستم عامل ویندوز ۱۰ شبیه سازی شده و بر روی مجموعه داده مورد نظر توسط معیار ارزیابی، مورد ارزیابی قرار گرفته است. به منظور آنالیز پاسخ های هر یک از شبکه ها و محاسبه دقت مدل ها از پارامترهای مختلفی استفاده شده است .

این پارامترها عبارتند از :

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \quad (۱۴)$$

$$precision = \frac{TP}{TP+FP} \quad (۱۵)$$

$$recall = \frac{TP}{TP+FN} \quad (۱۶)$$

$$F1\_Score = \frac{2 \times TP}{2 \times TP+FP+FN} \quad (۱۷)$$

در روابط فوق :

True Positive (TP): تعداد نمونه‌هایی که به درستی به عنوان مثبت شناسایی شده‌اند

True Negative (TN) : تعداد نمونه‌هایی که به درستی به عنوان منفی شناسایی شده‌اند.

False Positive (FP): تعداد نمونه‌هایی که به اشتباه به عنوان مثبت شناسایی شده‌اند. به این معنی که مدل یک نمونه را به اشتباه مثبت پیش‌بینی می‌کند در حالی که آن نمونه در واقع منفی است.

False Negative (FN): تعداد نمونه‌هایی که به اشتباه به عنوان منفی شناسایی شده‌اند. در این حالت، مدل یک نمونه مثبت را به اشتباه منفی پیش‌بینی می‌کند، در حالی که آن نمونه در واقع مثبت است.

معیار حساسیت یا صحت بیانگر درصد تشخیص دادن درست مدل مثبت است. سنجش F1_score نشان دهنده تعادل بین کارایی هر طبقه است.

درنهایت با استفاده از ماتریس آشفتگی[۱] که در برگیرنده آنالیزهای فوق هستند جدول ۲ طراحی شد که هر یک از مراحل روش پیشنهادی را نشان می دهد. شایان ذکر است تعداد تکرار در الگوریتم گرگ خاکستری ۱۵ دوره و وزن های W_1 و W_2 به ترتیب ۰,۹۹ و ۰,۰۱ در نظر گرفته شده اند. به منظور اطمینان از صحت محاسبات هر آزمایش ۵ بار تکرار شده و میانگین دقت لحاظ و در جدول ثبت شده است.

[۱] Confusion Matrix

جدول ۲: نتایج حاصل از انتخاب ویژگی ها پس از اعمال PCA و گرگ خاکستری

گزارش داده‌های آموزش											
معیار ترکیبی	زمان	تابع ارزیابی گرگ خاکستری	F1 Score	حساسیت (recall)	دقت (precision)	صحت (Accuracy)	حملاتی که ارسال /تشخیص داده شده‌اند	زمان‌هایی که حمله تشخیص داده شده‌اند	حملاتی که درست تشخیص شده‌اند	زمان‌هایی که به درستی تشخیص داده شده	تعداد ویژگی های انتخاب شده
۴۸۶۹۳	۳۸	---	٪۹۶.۹۸	٪۹۴.۶	٪۹۹.۵	٪۹۳.۳	۲۷۹	۳۱۵۶	۵۵۴۷۶	۶۷۰۶۴	۴۱
۴۶۶۱۷	۳۳	۱/۲۲۲۲	٪۹۲.۸۳	٪۸۸.۶	٪۹۷.۵	٪۹۳.۶	۱۳۳۳	۶۷۰۴	۵۱۹۲۶	۶۶۰۱۰	۴۱-۲
۴۸۱۹۶	۳۵	---	٪۹۶.۳۹	٪۹۴.۲	٪۹۸.۷	٪۹۶.۷	۷۲۹	۴۲۲۲	۵۵۲۰۸	۶۶۶۱۴	۳۰
۴۵۰۷۸	۳۰	۱۰۳۸	٪۹۰.۵۵	٪۹۲.۳	٪۸۳.۹	٪۹۰۶	۱۰۶۶۵	۱۰۶	۵۷۰۲۴	۵۶۲۷۸	۳۰-۸
۴۸۶۸۶	۳۹	---	٪۹۷.۳۷	٪۹۵.۸	٪۹۹	٪۹۷۶	۵۶۹	۲۴۸۶	۵۶۱۴۶	۶۶۷۷۴	۲۵
۴۸۶۲۶	۱۵۲	۱۰۳۰	٪۹۷.۲۵	٪۹۵.۲	٪۹۹.۲	٪۹۷۴	۴۲۸	۲۸۱۱	۵۵۸۱۹	۶۶۹۱۵	۲۵-۶
۴۸۳۲۱	۳۶	---	٪۹۶.۶۴	٪۹۴.۶	٪۹۹	٪۹۶۹	۵۷۶	۲۵۸۸	۵۵۳۴۴	۶۶۷۶۷	۱۵
۴۸۶۰۶	۲۱۵	۰/۹۸۵۳	٪۹۶.۸۱	٪۹۴.۹	٪۹۸.۸	٪۹۷۱	۶۸۲	۲۹۶۱	۵۵۶۴۹	۶۶۶۶۱	۱۵-۶
گزارش داده‌های آزمون											
۳۵۶۶۷	۳۸	---	٪۷۱.۳۳	٪۵۷.۷	٪۹۳.۶	٪۷۳.۶	۵۳۵	۵۴۲۲	۷۲۱۱	۹۱۸۵	۴۱
۳۷۸۵۳	۳۳	۱/۲۲۲۲	٪۷۵.۹	٪۶۳.۶	٪۹۶.۵	٪۷۷.۶	۲۹۳	۶۸۰۳	۸۰۳۰	۹۲۱۷	۲۱-۲
۳۵۰۵۱۱	۲۵۲	---	٪۷۱.۰۲	٪۵۷.۸	٪۹۲.۱	٪۷۳.۱	۶۳۵	۵۴۱۸	۷۲۱۵	۹۰۷۵	۳۰
۳۸۰۲۱۱	۳۰	۱۰۳۸	٪۷۶.۸۲	٪۶۵.۹	٪۹۲.۱	٪۷۷.۶	۷۲۹	۴۲۷۳	۸۴۶۰	۸۹۸۱	۳۰-۸
۳۶۷۸۱	۳۸	---	٪۷۳.۵۶	٪۶۱.۲	٪۹۲.۲	٪۷۵	۶۶۵	۴۹۷۵	۷۸۵۸	۹۰۴۵	۲۵
۳۹۰۲۰۱	۱۵۲	۱۰۳۰	٪۷۸.۲	٪۶۶.۷	٪۹۵.۳	٪۷۹.۲	۴۲۴	۲۲۶۸	۸۵۶۵	۹۲۸۶	۲۵-۶
۳۶۷۰۱	۳۲	---	٪۷۳.۶	٪۶۱.۶	٪۹۱.۶	٪۷۴.۸	۳۲۹	۲۹۵۱	۷۸۸۲	۸۹۷۱	۱۵
۳۸۵۸۱	۲۱۵	۰/۹۸۵۳	٪۷۷.۱۶	٪۶۶.۶	٪۹۱.۷	٪۷۷.۶	۷۷۸	۲۲۸۰	۸۵۵۳	۸۹۲۲	۱۵-۶

همانطور که جدول ۲ نشان می دهد ردیف های فرد جدول (۱،۳،۵،۷) هم در دادهای آزمون و هم در داده های تست بیانگر حالتی است که فقط از PCA جهت تشخیص نفوذ استفاده شده است اما در ردیف های زوج (۲،۴،۶،۸) از گرگ خاکستری استفاده شده است تا بهترین ویژگی ها را بیابد. به عنوان مثال در ردیف ۳ داده های آموزش الگوریتم PCA تعداد ۳۰ ویژگی برتر را انتخاب کرده است و بر اساس آن تشخیص نفوذ با پارامتر ارزیابی F1_score مقدار ۹۶،۳۹ حاصل شده است. ردیف ۴ همان جدول نشان می دهد، که وقتی از گرگ خاکستری استفاده می شود تا از ۳۰ ویژگی مفروض بهترین ویژگی ها را انتخاب کند فقط تعداد ۸ ویژگی انتخاب شده و پارامتر f1_score با مقدار ۹۰/۱۵ بدست می آید.

این می تواند بدان معنی باشد که وجود برخی از ویژگی ها تاثیر چندانی در افزایش دقت حملات ندارد. ما آزمایش را ادامه دادیم و اجازه دادیم تا الگوریتم گرگ خاکستری از بین ویژگی انتخاب شده توسط PCA بهترین ویژگی ها و مؤثرترین آنها را انتخاب کند. در نهایت الگوریتم گرگ خاکستری ۶ ویژگی از بین ۲۵ ویژگی را انتخاب می کند که بهترین عملکرد را در بین سایر حالت ها دارد. آنچه که می تواند مورد توجه قرار گیرد مدت زمان آموزش است. زمان آموزش با کاهش تعداد ویژگی های مسئله کاهش می یابد. از آنجایی که در سیستم های تشخیص نفوذ، وجود حملات جدید محتمل است و سیستم های آنلاین باید بروز شده و در بازه زمانی مشخص آپدیت شوند و آموزش ببینند[32]. لذا مدت زمان آموزش اهمیت پیدا می کند.

ستون آخر جدول پارمتر زمان و f1_score را باهم ترکیب می کند و بر اساس رابطه ۱۴ محاسبه شده است:

(14)

$$F1_{Time} = W_1 \times F_{1_{score}} + W_2 \times \frac{1}{Time}$$

در این رابطه هرچقدر مقدار F1_Time بیشتر باشد نشانه برتری ویژگی انتخاب شده می باشد. مقادیر w1 و w2 میزان اهمیت زمان و f1_score است که مقدار ۰،۵ انتخاب شده است. البته با توجه به تفاوت سخت افزارهای مورد استفاده در سایر پژوهش ها ، مقایسه زمان اجرای کدها با آن ها قابل استناد نمی باشد. اما مقایسه دقت و زمان روشهای به کار رفته در این پژوهش ، حاکی از سرعت بهتر روش پیشنهادی نسبت به دقت و زمان ارائه شده در سایر الگوریتم ها می باشد. گزارش تعداد تشخیص های صحیح و خطا در نمونه های آموزش و آزمون در جدول ۳ نشان داده شده است.

۶- نتیجه گیری و کارهای آینده

آنچه که در این مطالعه مد نظر نویسندگان بوده است ترکیب دو روش خطی و غیر خطی به منظور استخراج ویژگی بوده است. نوآوری اصلی این مطالعه در ترکیب دو الگوریتم PCA و GWO برای بهبود در تشخیص نفوذ در شبکه‌ها است. این ترکیب به منظور کشف ویژگی‌های برتر از داده‌ها طراحی شده است، به‌طوری‌که در آن شناسایی دقیق‌تر و مؤثرتر ویژگی‌ها جهت افزایش دقت تشخیص، در نظر گرفته می‌شود. در حالی که PCA به طور سنتی بر اساس واریانس داده‌ها عمل می‌کند، GWO می‌تواند به کشف ویژگی‌های مهم‌تر با واریانس پایین، که ممکن است در الگوریتم‌های انتخاب ویژگی سنتی نادیده گرفته شوند، کمک کند.

ترکیب PCA با GWO به چند دلیل مورد توجه بوده است:

اول اینکه PCA به طور ذاتی به دنبال ویژگی‌هایی با بالاترین واریانس است، اما ویژگی‌هایی با واریانس کم نیز ممکن است حاوی اطلاعاتی مهمی باشند. GWO می‌تواند کمک کند تا این ویژگی‌ها شناسایی شوند و بر اساس اهمیت آن‌ها در نتایج نهایی بهینه‌سازی واقعی انجام دهد.

دوم اینکه قابلیت بهینه‌سازی گرگ خاکستری، به دلیل روش قرارگیری اجتماعی توانایی خوبی در جستجو و بهینه‌سازی فضایی متنوع دارد. و این امکان را می‌دهد که ویژگی‌های بهینه را از میان گزینه‌ها مفروض پیدا کند، در حالی که PCA به طور پیشفرض ممکن است همه ویژگی‌ها را در فرآیند انتخاب، محدود کند.

اگرچه استفاده از سایر الگوریتم های فرا ابتکاری می تواند مفید باشد، و حتی ممکن است منجر به نتایج بهتری شود، لیکن فرض اولیه مطرح شده در این پژوهش توجه به ترکیب روشهای خطی و غیر خطی به خصوص روش های فرا ابتکاری است. گرگ خاکستری صرفا به دلیل سادگی مفاهیم و مراحل پیاده‌سازی، انتخاب شده است که بتواند اجرای سریع‌تر داشته و با پیچیدگی کمتری به نتایج مطلوب برسد.

در این مطالعه الگوریتم PCA با الگوریتم گرگ خاکستری برای پیدا کردن ویژگی‌های بهینه پیشنهاد شده است. که می تواند با دقت بالایی از نظر زمان و دقت تشخیص، نوع حملات را شناسایی کند. استفاده از الگوریتم PCA این امکان را می‌دهد که با حفظ بیشترین واریانس، ابعاد داده‌ها کاهش پیدا کند. این امر باعث کاهش زمان محاسبات و افزایش سرعت همگرایی الگوریتم‌های یادگیری می‌شود. پس از کاهش ابعاد، الگوریتم گرگ خاکستری به عنوان یک تکنیک بهینه‌سازی برای انتخاب بهترین ویژگی‌ها عمل می کند.

در واقع، "گرگ" نماد نهادینه‌سازی در فرآیند جستجو است که به الگوریتم اجازه می‌دهد در فضای ویژگی‌ها برای یافتن بهترین ترکیب به کار برود. رفتار گروهی گرگ‌ها به آن‌ها این امکان را می‌دهد که با تبادل اطلاعات، از بهترین موقعیت‌های موجود بهره‌برداری کنند و بدین ترتیب به سمت بهترین حالت پیشروی کنند. این روش این امکان را می دهد تا مجموعه‌ای از ویژگی‌ها که بیشترین تأثیر را بر روی نتایج مدل دارند، شناسایی شوند. با انتخاب ویژگی‌های مهم و کاهش ابعاد داده‌ها، احتمال اورفیتینگ (overfitting) کاهش می‌یابد و مدل می‌تواند عملکرد بهتری روی داده های جدید داشته باشد. ترکیب این دو روش می‌تواند دقت مدل را افزایش و بخصوص مدت زمان آموزش را کاهش دهد. و از

آنجایی که در مدل های تشخیص نفوذ فرایند آپدیت کردن مدل ها نیاز به آموزش مداوم دارد زمان آموزش اهمیت پیدا می کند. جدول ۳ مقایسه عملکرد روش پیشنهادی با سایر پژوهش های مشابه را نشان می دهد.

جدول ۳ : مقایسه عملکرد روش پیشنهادی با سایر پژوهش ها

ردیف	شماره مرجع	طبقه‌بندی	دقت	زمان ثانیه	تعداد کلاس	سال پژوهش	مجموعه داده
۱	[29]	یادگیری ماشین و الگوریتم سنجاقک	٪ ۸۹	۶۹۹	۵	۲۰۲۲	KDD-99
۲	[34]	تکنیک های یادگیری عمیق	٪ ۷۳,۹۲	--	۵	۲۰۱۹	NSL-KDD
۳	[35]	منطق فازی	٪ ۹۱,۲۶	--	۲	۲۰۲۱	KDD-99
۴	[36]	نزدیک ترین همسایه (KNN)	٪ ۹۴	۹۰۸	۲	۲۰۲۰	CICIDS2017
۵	[36]	ازدحام ذرات (PSO)	٪ ۹۴	۳۷۵	۲	۲۰۲۰	CICIDS2017
۶	[37]	Bagging-REPTree	٪ ۸۳,۲۲	—	—	۲۰۱۸	NSL-KDD
۷	[38]	SVM Quadratic	٪ ۸۷,۴۶	—	—	۲۰۲۱	ISCX 2012
۸	[39]	ANFIS	٪ ۷۹,۲۶	۲۲.۵	۴	۲۰۲۲	CAIDAs
۹	روش پیشنهادی	PCA_GWO	٪ ۹۷,۲۵	۴۵۴	۲	۲۰۲۴	NSL-KDD

اگرچه در جدول فوق پارامتر زمان به عنوان یک معیار ارزیابی استانداردی مطرح نمی باشد و محققین حوزه تشخیص نفوذ روی این مورد تاکید زیادی ندارند، لیکن فقط به منظور مقایسه با برخی از متدهای رایج، این پارامتر محاسبه شده است. در نهایت پارامتر دقت به عنوان یکی ازپارمترهای شاخص محاسبه شده است. ایراداتی نیز ممکن است در روش پیشنهادی وجود داشته باشد که می تواند در کارهای آتی مورد توجه قرار گیرد. به عنوان مثال PCA به طور ذاتی برای تحلیل داده‌های خطی طراحی شده است و در مواردی که داده‌ها غیرخطی هستند، ممکن است قابلیت‌های آن محدود باشد. همچنین احتمال حذف ویژگی‌های با اهمیت که در واریانس‌های پایین‌تر قرار دارند، محتمل است. استفاده از تکنیک‌های دیگر مانند Kernel PCA می تواند جایگزین مناسبی باشد. همچنین الگوریتم گرگ خاکستری، به عنوان یک روش بهینه‌سازی، ممکن است به خوبی با ویژگی‌های منتخب از PCA سازگاری نداشته باشد و استفاده از سایر الگوریتم های بهینه سازی می تواند مورد توجه در کارهای آتی قرار گیرد. افزایش تشخیص سایر کلاس ها با روش پیشنهادی نیز می تواند در فرایندهای آینده مورد توجه قرار گیرد.

۷- سپاسگزاری

این مقاله مستخرج از طرح پژوهشی به شماره ۱۴۰۳/۱۴/۰۹/۵۹۶۵ دانشگاه آزاد اسلامی واحد تربت حیدریه می باشد.

منابع

[1] Aghdam, M. H., & Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. *Int. J. Netw. Secur.*, *18*(3), 420-432.

[2] Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of network and computer applications*, *34*(4), (1184-1199).

[3] Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., & Perkasa, C. D. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert systems with Applications*, *38*(1), (306-313).

[4] Toosi, A. N., & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer communications*, *30(10)*, (2201-2212).

[5] Maroosi, A., Zabbah, I., Mogharebi, M., Yasrebi, E., & Layeghi, K. (2022). Improving Diagnosis of Breast Cancer Disease Using Adaptive Neuro-fuzzy Inference System. *Karafan Quarterly Scientific Journal*, *19*(3), (377-391).

[6] Yahalom, R., Steren, A., Nameri, Y., Roytman, M., Porgador, A., & Elovici, Y. (2019). Improving the effectiveness of intrusion detection systems for hierarchical data. *Knowledge-Based Systems*, *168*, (59-69).

[7] Talaei Khoei, T., & Kaabouch, N. (2023). A comparative analysis of supervised and unsupervised models for detecting attacks on the intrusion detection systems. *Information*, *14*(2), (103).

[8] Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. (2019, January). Network intrusion detection using supervised machine learning technique with feature selection. In *2019 International conference on robotics, electrical and signal processing techniques (ICREST)* (643-646).

[9] Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert systems with applications*, *39(1)*, (424-430).

[10] Wazirali, R. (2020). An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation. *Arabian Journal for Science and Engineering*, *45*(12), (10859-10873).

[11] Besharati, E., Naderan, M., & Namjoo, E. (2019). LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. *Journal of Ambient Intelligence and Humanized Computing*, *10*, (3669-3692).

[12] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access*.

[13] Maroosi, A., Zabbah, E., & Ataei Khabbaz, H. (2020). Network Intrusion Detection using a combination of artificial neural networks in a hierarchical manner. *Electronic and Cyber Defense*, *8*(1), (89-99).

[14] Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. *Security and communication networks*, *2020*(1), (8890306).

[15] Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences*, *513*, (386-396).

[16] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, (21954-21961).

[17] Heidari, A., Navimipour, N. J., & Unal, M. (2023). A secure intrusion detection platform using blockchain and radial basis function neural networks for internet of drones. *IEEE Internet of Things Journal*, *10*(10), (8445-8454).

[18] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access, 7*, (41525-41550).
[19] Luo, J., Chai, S., Zhang, B., Xia, Y., Gao, J., & Zeng, G. (2020). A novel intrusion detection method based on threshold modification using receiver operating characteristic curve. *Concurrency and Computation: Practice and Experience, 32(14)*, (e5690).
[20] Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science, 167*, (1561-1573).
[21] Arivudainambi, D., KA, V. K., & Chakkaravarthy, S. S. (2019). RETRACTED ARTICLE: LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks. *Neural Computing & Applications, 31(5)*, (1491-1501).
[22] Velliangiri, S., & Pandey, H. M. (2020). Fuzzy-Taylor-elephant herd optimization inspired Deep Belief Network for DDoS attack detection and comparison with state-of-the-arts algorithms. *Future Generation Computer Systems, 110*, (80-90).
[23] Wilson, A. J., & Giriprasad, S. (2020). A Feature Selection Algorithm for Intrusion Detection System Based On New Meta-Heuristic Optimization. *Journal of Soft Computing and Engineering Applications, 1(1).*
[24] Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science, 89*, (213-217).
[25] Khorram, T., & Baykan, N. A. (2018). Feature selection in network intrusion detection using metaheuristic algorithms. *International Journal of Advanced Research, Ideas and Innovations in Technology, 4(4)*, (704-710).
[26] Khosravian, E. (2022). Design Optimal Adaptive Trajectory Tracking Control for Station Keeping and Attitude Control of Quadrotor Using Gray Wolf Optimization. *Karafan Journal, 19(3),* (663-694).
[27] Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep learning models for cyber security in IoT networks. *In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC) (0452-0457). IEEE.*
[28] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. *In 2009 IEEE symposium on computational intelligence for security and defense applications (1-6). Ieee.*
[29] Zabbah, I., Layeghi, K., & Ebrahimpour, R. (2024). A Multi-level Deep Neural Network to Diagnose Coronavirus Disease with Imbalanced Data.10(3), (33-42).
[30] Hochreiter, S. (1997). Long Short-term Memory. Neural Computation MIT-Press.
[31] Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing, 18*, (178-184).
[32] Purushothaman, R., Rajagopalan, S. P., & Dhandapani, G. (2020). Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering. *Applied Soft Computing, 96*, (106651).
[33] Babagoli, M. (2023). Propose a meta-heuristic model of intrusion detection using feature selection based on improved gray wolf optimization and random forest. *Signal and Data Processing, 20(1),* (133-144).
[34] Faker, O., & Dogdu, E. (2019, April). Intrusion detection using big data and deep learning techniques. *In Proceedings of the 2019 ACM Southeast conference* (86-93).
[35] Akhlaghpour, M. (2021). Providing a Solution Based on Fuzzy Logic to Reduce False Positive Alarms in the Intrusion Detection System. *Intelligent Multimedia Processing and Communication Systems (IMPCS), 2(4),* (45-50).

[36] Netaj Salehdar, Mohammad Hassan, (2019). Improving the performance of intrusion detection systems using intelligent feature reduction algorithms, *The 13th International Conference of Iranian Operations Research Society, Shahrud*,

[37] Pham, N. T., Foo, E., Suriadi, S., Jeffrey, H., & Lahza, H. F. M. (2018, January). Improving performance of intrusion detection system using ensemble methods and feature selection. *In Proceedings of the Australasian computer science week multiconference* (1-6).

[38] Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Computer Networks, 188*, (107840).

[39] Hassan Nataj Solhdar, M. (2022). Investigation of a new ensemble method of intrusion detection system on different data sets. *Electronic and Cyber Defense, 10(3)*, (43-57).