Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.057702

ARTICLE

Prairie Araneida Optimization Based Fused CNN Model for Intrusion Detection

Nishit Patil and Shubhalaxmi Joshi*

School of Computer Science, Dr. Vishwanath Karad MIT-WPU, Pune, 411038, Maharashtra, India

*Corresponding Author: Shubhalaxmi Joshi. Email: shubhalaxmi.joshi@mitwpu.edu.in

Received: 25 October 2024 Accepted: 05 November 2024 Published: 03 January 2025

ABSTRACT

Intrusion detection (ID) is a cyber security practice that encompasses the process of monitoring network activities to identify unauthorized or malicious actions. This includes problems like the difficulties of existing intrusion detection models to identify emerging attacks, generating many false alarms, and their inability and difficulty to adapt themselves with time when it comes to threats, hence to overcome all those existing challenges in this research develop a Prairie Araneida optimization based fused Convolutional Neural Network model (PAO-CNN) for intrusion detection. The fused CNN (Convolutional Neural Netowrk) is a remarkable development since it combines statistical features that are extracted from the processed data and provide enhanced capabilities for the model to capture complicated patterns existing in intrusion datasets. The adoption of a fused architecture represents an integrated way towards intrusion detection where the model can significantly interpret various features to achieve higher accuracy. On top of this, the Prairie Araneida stage which is based on coyote behavior and social spider colonies respectively plays a role in enabling to handling of intricate optimization landscapes. The dual contribution of a fused CNN and novel optimization strategies strengthens the research's goal to design an effective intrusion detection system that can evolve with new cyber threats. When the Training percentage (TP) is set to 90, the model's performance can be assessed using metrics like Accuracy, Sensitivity, and Specificity. In this particular dataset, these metrics reach approximately 97.78%, 96.25%, and 96.15%, respectively, which are crucial values. Additionally, when using a k-fold value of 6, the model achieves metrics of 97.04% Accuracy, 97.37% Sensitivity, and 96.48% Specificity.

KEYWORDS

Fused convolutional neural network; Prairie Araneida optimization; intrusion detection; preprocessing and feature extraction

1 Introduction

Intrusion Detection (ID) serves as a pivotal cornerstone in the realm of cyber security, playing a vital role in fortifying digital ecosystems against the persistent and ever-evolving landscape of cyber threats [1]. This multifaceted approach encompasses a spectrum of strategies and techniques aimed at identifying unauthorized and potentially malicious activities that could compromise the security of computer networks and systems [2], as the digital realm becomes increasingly interconnected and complex, the need for proactive measures that can rapidly detect and respond to emerging

threats becomes paramount [3–5]. By meticulously scrutinizing network traffic, system logs, and user behaviors, Intrusion Detection distinguishes anomalies and deviations from established patterns, sounding the alarm on potential breaches or unauthorized access attempts. This dynamic defense mechanism has evolved beyond mere reactive responses, incorporating elements such as real-time monitoring, continuous learning, and automated countermeasures [6–8]. Through these means, Intrusion Detection stands as a formidable guardian, tirelessly patrolling the digital frontier to uphold the integrity, confidentiality, and availability of critical data and resources in a dynamic and interconnected world [9].

IDS (Intrusion Detection Systems) finds versatile applications across industries and sectors, playing a pivotal role in enhancing cyber security. By continuously monitoring network activities and system behaviors, IDS can swiftly detect and mitigate potential threats, contributing to data protection, operational continuity, and public safety. From safeguarding financial transactions, critical infrastructure, and healthcare systems to securing government networks, cloud environments, and Internet of Things (IoT) ecosystems, IDS acts as a proactive defense mechanism [10]. It ensures the integrity of online businesses, connected vehicles, and aerospace systems, and even extends its protective reach to individual homes and educational institutions. Additionally, IDS aids incident response teams and feeds into threat intelligence platforms, bolstering global cyber security efforts. Overall, the diverse and essential applications of IDS underscore its significance in fortifying digital landscapes against the ever-evolving array of cyber threats [11–14].

IDS offers several compelling advantages that bolster cyber security efforts. By constantly monitoring network activities and system logs in real-time, IDS enables the prompt detection of unauthorized and malicious activities, allowing for swift intervention to thwart potential threats before they escalate [15]. Their diverse range of detection techniques, including signature-based and anomaly-based methods, ensures comprehensive coverage across various attack vectors. IDS also adapts to evolving network behaviors, providing continuous protection even as the threat landscape changes [16–19]. Their ability to trigger automated responses streamlines incident handling and reduces response time, minimizing the potential impact of attacks. Furthermore, IDS aid in regulatory compliance, assist in forensic analysis and contribute to proactive defense strategies [20]. Ultimately, their role in enhancing incident response efficiency, reducing false positives, and mitigating financial and reputational costs positions IDS as a crucial component in safeguarding digital environments [21,22].

In Machine Learning (ML), ID refers to the process of utilizing algorithms and models to automatically identify unauthorized or anomalous activities within computer networks or systems. It involves analyzing patterns in data collected from various sources, such as network traffic and system logs, to distinguish between normal and potentially malicious behavior. By training models on labeled data, Intrusion Detection Systems (IDS) can recognize deviations from established patterns and generate alerts, enhancing cyber security by swiftly identifying potential threats [23]. In essence, the working of ID in machine learning involves training models to recognize patterns of normal and malicious behavior, deploying them in real-time monitoring scenarios, and generating alerts when anomalous activities are detected, contributing to maintaining the security of networks and systems [24].

Existing intrusion detection models face a range of challenges that impact their effectiveness in safeguarding digital systems. Some difficulties that need to be addressed include dealing with datasets that are skewed towards normal instances rather than anomalies, adjusting to new attack methods that were not encountered during training, and managing data drift resulting from legitimate changes in network behavior [25]. Feature engineering, essential for accurate detection, can be intricate, while the

complexity of high-dimensional data strains processing efficiency [9,10,26]. Balancing false alarms and misses is crucial, and the transferability of models across different network environments is a concern. Models must perform real-time analysis, maintain privacy, and be interpretable to build trust. Furthermore, the emergence of adversarial attacks poses a threat to the reliability of these models. Overcoming these challenges demands continuous innovation to develop robust, adaptable, and accurate ID solutions [27–29].

This research mainly contributes to creating a PAO-CNN model that significantly improves the efficiency of IDS. This novel approach involves data acquisition followed by preprocessing feature extractions and using fused CNN (Convolutional Neural Network) architecture. The optimization refines the model for greater accuracy. Significantly, the model shows adaptive behavior during testing by changing reactivity to data drift. This adaptability ensures effective retraining when drifts are identified so that the model remains viable for capturing emerging attack patterns. On the other hand, when no drift is observed things are minimized in terms of unnecessary processing and ensure that an optimized configuration for the model remains effective. This innovation helps this specific problem of successfully adjusting to evolving attack behaviors and further enhances the general effectiveness of IDS. The major contributions are listed as follows:

- **Prairie Araneida Optimization (PAO):** The PAO algorithm is incorporated with the classification model that combines the characteristics of the coyotes and social spiders and aids in enhancing the efficiency of the classification model to detect intrusions accurately.
- **Prairie Araneida Optimization Based Fused Convolutional Neural Network Model (PAO-CNN):** The standard CNN model is combined with the PAO algorithm and achieves high-efficiency outcomes of ID. Further, the inclusion of the drift mechanism is named fused, which detects the drift and ensures testing the drift-free data again. Overall, the research model achieves maximum efficacy with the contributed mechanisms.

With these contributions, the proposed methodology of the research is depicted in Fig. 1, which is illustrated as follows:

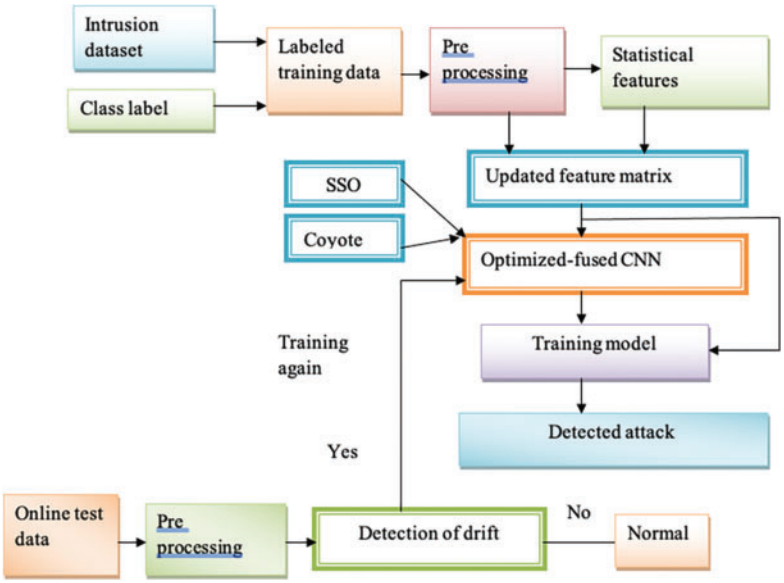

Figure 1: Architecture of the proposed intrusion detection model

Organization

The research article is organized as: Section 2 describes the literature review, Section 3 elaborates on the proposed research methodology, Section 4 depicts the research outcomes, and Section 5 ends the research with the conclusion and future work.

2 Motivation

Nowadays we are living in a digitally interconnected world and because of this increased cyber threats to data and systems pose great risks. ID models have a crucial role in preventing such threats, as they monitor and analyze network activities. The motivation is to strengthen cyber security in an effort of detecting unauthorized access and averting breaches besides ensuring smooth operation of critical services. To have a secure digital landscape, it is essential to improve intrusion detection models as attacks change.

2.1 Literature Review

Jin et al. [1] focused on the problem of prompt ID in high-speed networks and suggested deployment by the Swift IDS system. The dataset used in this paper was the CIC-IDS-2017 dataset. It was the technology that enabled the analysis of large traffic data without compromising timely detection, which promised a potential improvement in network security. Furthermore, the parallel ID approach introduced complications regarding the synchronization and control of numerous detection processes. In this digitally interconnected world, a wave of cyber threats posed significant risks to data and systems. Network ID Models were critical in combating these threats by tracking and evaluating Mohammed et al. [2] developed a new Network IDS that overcame the limitations of current systems, delivered high detection accuracy while demonstrating the potential for powerful intrusion occurrence prevention within contemporary networks idea. The datasets used in this research were NSL-KDD and CIC-IDS2017 datasets. However, it faces difficulties in the detection of accuracy. Iram et al. [3] created a powerful ID system in which ML classifiers were used to effectively distinguish between typical and intrusive activities, thus improving network security, with high-performance identification rates above 99% for diverse attack classes. The dataset employed in this work for training and testing the model was the NSL-KDD dataset. However, the performance of such an approach might depend on constantly changing ways by attackers. Naderi et al. [4] sought to increase network ID by using a hybrid approach, incorporating deep neural networks as well as sophisticated ML algorithms. This work utilized a real-time UNB ISCX 2012 dataset to validate the performance of the model. This model united the advantages of Deep Learning (DL) and ML increasing both accuracy and efficiency in handling a huge amount of data about network traffic, however, this approach might differ to be effectiveness depending on the complexity and nature of network intrusions. Tuan et al. [5] sought to improve ID in computer networks by creating a new algorithm that combines multiple approaches leading to better detection performance as well as general applicability in real network settings. In this work, the NSL-KDD dataset was employed to validate the performance of the model. Here the KDDtrain+ was used for the training set and KDDtest+ and KDDtest-21 were utilized for the validation and test set. However, the algorithm performance depends on network complexity and choice of parameters. Zhang et al. [6] attempted to improve ID by designing a unified model that combined Multi-Scale CNN (MSCNN) and Long Short-Term Memory (LSTM) networks to provide an overall understanding of spatial-temporal features for enhanced accuracy and fewer false positives. This research employed a UNSW-NB15 dataset as an experimental training and testing set to validate the model performance. However, the performance of such models could

be impacted due to dataset variability effectiveness was subjective especially when it came as per intrusion patterns. Abhishek et al. [7] sought to encourage the adoption of ensemble learning for IoT security enhancement, concentrating specifically on DoS attacks. The research targeted the urgent necessity for strong IoT security by discussing ML classifiers and promoting state-of-the-art IDS development in this work, the CIDDS-001, UNSW-NB15, and NSL-KDD datasets were employed to validate the performance of the model. However, the success of the proposed methodology might also depend on how fast was changing landscape of IoT threats and whether classifiers could be usefully adapted to specific IoT applications. Vikash et al. [8] created an innovative IDS that could accurately detect Exploits, DoS Probe and Generic categories of network activities as well as Normal category ones IDSs were capable of identifying specific threats in networks, these systems contributed greater accuracy and efficiency. This research employed a UNSW-NB15 and RTNITP18 dataset to validate the performance of the model. but their weakness layed within the lack of Apache adaptability for developing techniques. Omar et al. [30] presented an enhanced anomaly-based ID DL Multi-class classification model. The dataset used in this research is CICIDS2017 dataset. A separate DL model could be used to identify the attack type among grouped similar classes. However, the model had issues with computational complexity. Jiawei et al. [31] introduced a network ID classification model (NIDS-CNNLSTM) based on DL. The datasets used in this research were KDD CUP99, NSL-KDD, and UNSW_NB15 datasets. It was more suitable for large-scale and multi-scenario network data in the IIoT. However, it suffered from data imbalance and a limited number of training samples. Saheed et al. [32] developed an ensemble approach with the integration of a grey wolf optimizer (GWO), which effectively detected intrusion attacks in IoT networks. In the method, hybridization of principal component analysis (PCA), and information gain (IG) was implemented to efficiently eliminate the parameter dimensions. However, the reliability of the network was reduced and caused severe security threats in the IoT environment. The model worked with the Bot-IoT dataset and UNSW-NB15 dataset.

Saheed et al. [33] introduced a Bat Metaheuristic Algorithm (BMA) integrated with a Residue Number System (RNS) to detect the intrusion in the network and enhance the processing speed effectually. The provided model is evaluated with the NSL-KDD dataset. In the method, the RNS technique was used to explore the feature selection process whereas, the BMA extracted the features with the inclusion of PCA analysis. Based on these techniques, the model enhanced the processing speed and detection accuracy. However, when attacks were imposed in the network the model suffered from an imbalance problem and caused interpretability and scalability issues. The conventional methods described are precisely described in Table 1.

Table 1: Literature summary table

Ref. No.	Authors	Method	Dataset	Advantages	Disadvantages
[1]	Jin et al.	Swift IDS	CIC-IDS-2017 dataset	The model provided a potential improvement in network security.	The model raised complications regarding the synchronization and control of numerous detection processes.

(Continued)

Table 1 (continued)

Ref. No.	Authors	Method	Dataset	Advantages	Disadvantages
[2]	Mohammed et al.	New Network IDS	NSL-KDD and CIC-IDS2017 datasets	The model tried its best to detect IDS.	The model faced difficulties in the detection of accuracy.
[3]	Iram et al.	ML-based IDS	NSL-KDD dataset	The model attained high efficiency in detecting IDS.	The model needed to be updated on the advanced network attacks.
[4]	Naderi et al.	Hybrid approach with DNN	Real-time UNB ISCX 2012	The model had the efficiency in handling the high amount of data.	The efficiency of the research model varied based on the nature and complexity of the attacks.
[5]	Tuan et al.	Multiple approach-based new algorithm in IDS	NSL-KDD dataset	The generalizability as well as the efficiency of the research model was high.	The parameter choice and the attack complexity affected the performance of the research model.
[6]	Zhang et al.	Combined MSCNN and LSTM networks	UNSW-NB15 dataset	The model achieved high-efficiency performance.	The achieved performance of the model in IDS was affected by the data variability.
[7]	Abhishek et al.	Ensemble learning for IoT security	CIDDS-001, UNSW-NB15, and NSL-KDD datasets	The model worked with DDoS for which high accuracy was obtained.	The adaptability towards different attacks remained poor.
[8]	Vikash et al.	DoS Probe and Generic categories of network	UNSW-NB15 and RTNITP18 dataset	The model detected specific threats and attained efficient performance.	The model lacked Apache's adaptability.
[30]	Omar et al.	Enhanced anomaly-based ID DL Multi-class classification model	CICIDS2017 dataset	The model had high efficiency in detecting specific attacks among the grouped ones.	The model exhibited computational complexity.

(Continued)

Table 1 (continued)

Ref. No.	Authors	Method	Dataset	Advantages	Disadvantages
[31]	Du et al.	NIDS-CNNLSTM	KDD CUP99, NSL-KDD, and UNSW_NB15 datasets	The model worked with large-scale and multiple network data.	The model had data imbalance and the training samples issues.
[32]	Saheed et al.	An ensemble approach with grey wolf optimizer	Bot-IoT, UNSW-NB15	The PCA and IG incorporated in the research model eliminated the dependency of the parameters.	The model had less reliability.
[33]	Saheed et al.	A Bat Metaheuristic algorithm (BMA) integrated with a Residue Number System (RNS)	NSL-KDD dataset.	The extraction of the features aided the research model in attaining high efficiency.	The model had data imbalance issues and data interpretability.

Hence, the conventional methods in the research of ID exhibited several technical gaps. The existing models addressed the challenges concerning the dataset, and the one utilized in former research was highly imbalanced, which created a bias in the classifiers. Moreover, the existing research models incorporated hand-crafted mechanisms to perform feature extraction, which identified the irrelevant features that in turn affected the detection outcomes. In addition, the basic ID model was exhibited in the network traffic datasets, which were prone to dynamic changes that enabled the issue of drift. Moreover, the convergence issue arose due to improper training as well as the initialization of the hyperparameters of the model. Though this issue was addressed by most of the existing research through optimization, the hybrid and the advanced combination of optimization remained vacant. Thus, to address all the issues described and to achieve high-efficiency ID, the PAO-CNN model is proposed in the research.

2.2 Challenges

- ➢ Network intrusion datasets were often highly imbalanced, with a small proportion of attacks compared to normal traffic. This imbalance could lead to biased classifiers that prioritize the majority class and struggle to detect minority class attacks effectively [1].
- ➢ Identifying relevant features from a large pool of attributes is challenging. Selecting irrelevant or redundant features could lead to decreased classification performance and increased computational overhead [3].
- ➢ Balancing the trade-off between capturing complex patterns (avoiding underfitting) as well as avoiding noise and overemphasizing outliers (avoiding overfitting) was crucial [4].

- ➢ Choosing the right classification algorithm that suited the data distribution, class imbalance, and problem complexity was essential for achieving optimal performance [5].
- ➢ Changes in network traffic patterns over time could lead to concept drift, where the model's training data became less relevant. This could result in decreased classification accuracy [6].

3 Methodology for the Proposed Intrusion Detection Model Using the Prairie Araneida Optimization-Based Fused Convolutional Neural Network

The research aims at creating a PAO-CNN model to improve the performance of IDS. To start the process, an intrusion dataset is obtained and class labels are assigned for purposes of training. The labeled datasets [34–36] are then used to train an initial model. Later, data preprocessing guarantees the quality and readiness of this data for analysis. Data is processed and statistical features are obtained, which generate a new feature matrix. These features are combined into a fused CNN that is optimized with the aid of the Social Spider Algorithm and Coyote Optimization steps. Before the drift detection process, test data is preprocessed during testing. If data drift is spotted, the optimized fused CNN re-trains itself to accommodate new patterns in the evolving data. However, if there is no drift detected the model proceeds with its optimized setup minimizing unnecessary processing. This adaptive behavior means that the model is relevant and useful in capturing evolving attack behaviors. The proposed architecture is schematically represented in Fig. 1.

3.1 Input

The ID model receives input from the sources [34–36], which can be outlined as follows:

$$M = \sum_{i=1}^{a} N_i \tag{1}$$

The input data N_i comprises values ranging from 1 to a, M is the dataset. The input datasets utilized in the research are described in the following section:

Dataset Description

i) *BoT-IoT Dataset* [36]: The BoT-IoT dataset is focused on the Detection of Intrusions in IoT environments which covers realistic data that closely resembles actual traffic in IoT networks from benign devices and malicious botnets. It is used by researchers in developing and testing ID systems that are targeted toward the identification of botnet-related threats in IoT networks. The utilized data set is obtained from the https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 04 November 2024) source.

ii) *CIC-IDS2017* [35]: ID research uses the dataset CIC-IDS 2017 obtained from "https://www.unb.ca/cic/datasets/ids-2017.html" (accessed on 04 November 2024). It covers a wide array of network traffic data ranging from benign activities to malicious ones such as cyber-attacks. Data labeled aids in the evaluation of ID systems, and as it is scalable to meet various requirements, such data would be useful for research in cyber security development.

iii) *UNSW-NB15* [36]: ID research uses the UNSW-NB15 dataset from the source "https://www.kaggle.com/datasets/dhoogla/unswnb15" (accessed on 04 November 2024). It includes a wide range of network traffic information such as normal and many types of cyber attacks.

3.2 Data Labeling

After putting together the dataset, each data point is labeled with a class that describes its characteristics. These categories within the domain of ID are normal network behavior and different kinds of attacks from a Denial-of-Service (DoS) case to malware activities or an intrusion. The classification of each data point depends on its observed behavior, determining whether it signifies a normal network operation or an invasive action. Employing this annotated dataset is essential in the field of ML as it serves as a crucial component for training models. The primary goal is to train the model to recognize intricate patterns and relationships within the input data (features). When it comes to ID, this involves training a model to differentiate between normal network operations and various types of harmful cyber attacks.

3.3 Preprocessing with Data Cleaning Technique

First, a very important preprocessing phase is conducted to ensure the integrity and reliability of data. This phase entails a set of important operations, data cleaning among them all with an underlying aim to eliminate surplus noise and inconsistencies. Through identifying errors and improving the standard, preprocessed data is subject to careful refinement that ensures its suitability for feature extraction in subsequent phases.

$$M^* = \sum_{i=1}^{a} N_i^* \tag{2}$$

The preprocessed output is denoted as N_i^*.

3.4 Feature Extraction with Statistical Features

Feature extraction marks one of the most crucial parts where relevant information is obtained from the processed dataset. This entails the calculation of several statistical characteristics on the inherent properties of network traffic. These statistical characteristics provide useful information about the nature of network activities by quantifying patterns, distributions, and relationships within the data. As a result, these extracted features are sorted into a cleaned feature matrix that contains all the significant characteristics vital for further analysis and modeling.

Statistical Features

ID has statistical features which are quantifiable attributes drawn from the analysis of network traffic data. These features include a broad set of attributes ranging from data distribution to behavior patterns and relationships. These include mean, standard deviation, skewness, kurtosis, min, and max. Statistical features or characteristics assist in differentiating between normal and abnormal network activities through quantification of these aspects. This way, potential intrusions can be identified.

a) *Mean:* The mean is considered a statistical property computed from network traffic information which is the arithmetic mean of values taken within a defined data sample for an attribute in question. For example, when considering the mean of packet sizes in network traffic it shows what is an average size for a given sample. Such mean values may help provide insights into the baseline behavior related to network activities, thus facilitating the detection of anomalies or deviations from normal patterns that could indicate intrusion attempts. The mean outcome of the input N_i^* is T.
b) *Variance:* The measure of dispersion or spread of values in a dataset using statistics is known as variance. It tells how much the individual values differ from the mean or average. A higher

variance is indicative of a greater variability, meaning that the data points are further off from the mean while a low variance means they scatter close to around it. In the field of network traffic data analysis, variance can classify deviations or anomalies to attributes that assist in identifying potential intrusion efforts.

$$H = \frac{1}{a}\sum_{i=1}^{a}\left(N_i^* - T\right)^2 \tag{3}$$

where, T is the mean of the input.

c) *Standard Deviation (SD):* The SD quantifies the variability or range of values within a dataset through statistical analysis. It is an indicator of how much the values diverge from the mean of a dataset. Mathematically, the SD (σ) is calculated as the square root of the variance:

$$SD\,(\sigma) = \sqrt{\frac{1}{a}\sum_{i=1}^{a}\left(N_i^* - T\right)^2} \tag{4}$$

d) *Skewness:* Skewness is a statistical indicator that allows assessing the asymmetry of the probability distribution for a dataset. It gives clues as to how much, and in what direction, data distribution deviates from a symmetric bell-shaped curve. Positive skewness means that the distribution is shifted to yield a longer tail on the right-hand side of the curve. Negative skewness means a prominent left tail, which is longer.

$$S = \frac{\frac{1}{a}\sum_{i=1}^{a}\left(N_i^* - T\right)^3}{((a-1)\,.SD\,(\sigma))^3} \tag{5}$$

e) *Kurtosis:* Kurtosis is a statistical measure used to quantify the shape of the probability distribution for any set dataset. It describes the tails and peaks of the distribution, showing how far off from a normal distribution this data's spread is. Kurtosis assists in determining the "heaviness" or "lightness" nature of tails when compared to a normal distribution.

$$kurtosis = \frac{\frac{1}{a}\sum_{i=1}^{a}\left(N_i^* - T\right)^4}{((a-1)\,.SD\,(\sigma))^4} \tag{6}$$

f) *Min:* Min stands for the minimum of a certain attribute in the dataset or data sample. It stands for the lowest observed value of that attribute from the accessible data points. For instance, if we consider the minimum packet size in network traffic data it illustrates the smallest value of recorded packets per sample. Analyzing the bottom values of attributes in network traffic could help to understand limitations for normal behavior. Unusual or anomalously small minimum values could be indicative of abnormal activities, or potential intrusion attempts. Therefore, monitoring and comparison of the minimum values can be helpful in tracer deviations from anticipated patterns.

$$\min = \min\left(N_1^*, N_2^*, ...N_a^*\right) \tag{7}$$

g) *Max:* Max is the highest value in a dataset or data sample which represents the highest numerical value among an element in a dataset providing insight into how far one attribute can go. Max may be used in network traffic analysis to find unusually large values that could flag potential anomalies or security threats.

$$\max = \max\left(N_1^*, N_2^*, ...N_a^*\right) \tag{8}$$

3.5 Updated Feature Matrix

Once feature extraction is completed, the statistical features extracted are compiled into the updated feature matrix. This matrix is a structured depiction of the basic statistical features obtained from preprocessed data. Each row in the matrix stands for a particular data, and each column signifies an individual statistical trait such as mean-variance standard deviation skewness kurtosis minimum maximum. The importance of an updated feature matrix is that it allows for the capturing and condensing of complex information about the dataset into a form to be analyzed. By incorporating these statistical aspects, the matrix gives a general overview of how the data is distributed and variable. This is more than just a compressed view which also helps in efficient processing and the ability of the model to recognize meaningful patterns during further stages. In the general context of the study, a new feature matrix becomes an integral factor that enters into the fused CNN contributing to its adaptability and overall efficacy in ID. It essentially serves as a highly refined and informative data structure that captures the most important statistical insights gleaned from raw input data by way of performing feature extraction. The overall outcome of feature extraction is denoted as, X.

3.6 Working of Prairie Araneida Optimized Fused CNN in Intrusion Detection

Having updated the feature matrix, CNN has become a strong extractor of features in IDS. The modified feature matrix inclusive of critical statistical properties feeds into the CNN. Convolutional layers present within the network analyze local patterns and features, while activation functions introduce non-linearity to capture complex relationships. Then, there are fully connected layers that learn global patterns followed by an output layer that classifies data points either as normal or intrusive. As the CNN is trained, it optimizes its parameters using backpropagation and adapts to details of complexity present in the labeled dataset. In testing, the CNN treats new data and its adaptive nature happens during drift detection. If a data drift is detected, implying changes in the distribution of items within sets, the model can adjust itself by possibly retraining ensuring it remains relevant for capturing evolving characteristics under attack behaviors. Moreover, the fused CNN classifier is optimized with the PAO algorithm that optimizes the parameters to attain high-efficiency outcomes. Further, the usage of this optimization solves the issue with the class imbalance as the data imbalance remains the major concern in several conventional models to retain the model as the biased one. Hierarchical feature learning of CNN means that the network is capable not only of adapting itself for changing data patterns but also becomes a highly effective tool in terms of ID. The proposed model architecture is depicted in Fig. 2.

i) *Convolutional Layers:* Convolutional layers employ the input X, and let $conv\,(X, W)$ show a convolution operation on an input X with respect to a filter W. This operation involves a slide of the filter over input, where element-wise multiplication takes place and results are added together to give an output feature map.

$$i,j = Conv\,(X\,[i\colon i+i+f,\ \ j\colon j+f, W]) + b \tag{9}$$

Here, i and j are indices, f is the filter size, and b is the bias term.

ii) *Activation Functions:* They are nonlinear hence allowing the model to capture complex relationships and patterns within data. One of the common activation functions is *ReLU*, also known as Rectified Linear Activation or sigmoid Function. Mathematical Equation for *ReLU* Activation:

$$RELU\,(i,j) = \max\,(0, i, j) \tag{10}$$

iii) *Pooling Layers:* The activation functions follow the pooling layers that involve down down-sampling of data to conserve useful information with reduced dimensionality. Common pooling techniques include max-pooling and average-pooling. Pooling assists in the extraction of appropriate information and enhances the resilience of the model to variations.

$$P\left(RELU\left(i,j\right)\right) = \max\left(RELU\left(i,j\right)\left[i\colon i+p, j\colon j+p\right]\right) \tag{11}$$

Here, p is the pooling size.

iv) *Fully Connected Layers:* The data is then handed over to fully connected layers after the pooling layers. These layers link every neuron to every other neuron in the adjacent layers. It is therefore easier to learn complicated relationships and high-level representations of the data in fully connected layers.

$$FC\left(P\left(RELU\left(i,j\right)\right)\right) = W^{*}P\left(RELU\left(i,j\right)\right) + b \tag{12}$$

Here, W represents the weight matrix, while b denoting the bias term.

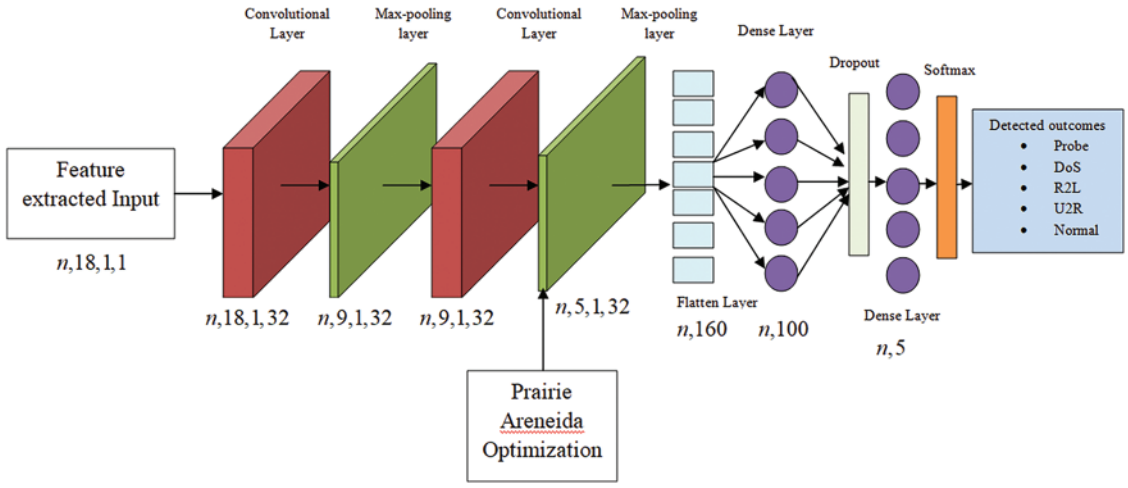

Figure 2: Architecture of the fused CNN model in the ID

3.7 Testing Phase

In the testing stage, preprocessing of test data is done to make sure it's ready for analysis. This processed test data enters the drift detection process. Significant changes in data patterns over time are called Data Drift. If such drift is traced, it implies that the behavior of the network has changed and this change can be related to new types or variations in normal activities. It is possible to retrain the optimized fused CNN in response to identifying data drift, where the retraining is based on the threshold. If the threshold obtained is less than the constant parameter of drift, then the retraining process is initiated, and when the threshold is above the constant then the drift is declared. Retraining is a process of updating the internal parameters and architecture of the model to fit with new patterns in data. This adaption allows the model to be able to readily identify new attack behaviors that have developed since initial training. On the one hand, no detected data drift points to relatively stable and consistent behavior of the network. In this scenario, the model proceeds to use its optimized setup without going into needless retraining. This method minimizes the computational burden and ensures effective use of resources. Thus, this adaptive behavior of the ID model to re-learn when significant

changes take place while maintaining its configuration under normal conditions contributes largely to improving an accuracy that captures both evolving attack patterns and normal network behaviors.

4 Proposed Prairie Araneida Optimization

The PAO model is created through the combination of features from both coyotes [37] and social spiders [38]. PAO fine-tunes classifier parameters which is an iterative process of optimizing the classifier whose goal is to tweak hyper-parameters, weights, and biases in order to find that configuration that gives optimal results for minimizing any classification error. The CNN detection model undergoes overfitting as well as less training performance, which is overcome by the PAO algorithm. The evolution of coyote-type cooperative group hunting behaviors into the foraging practices of spiders could revolutionize their ways of hunting. By using synchronized group techniques, spiders could be able to hunt tougher or bigger prey in a cooperative manner. This may involve tactful placement, herding prey for more effective capture chances, and designating certain tasks that will improve the probability of success when hunting an elusive quarry. This cooperative participation could help to save energy, reduce individual risk, and increase collective foraging effectiveness. Substantial evolutionary adaptations would be essential, but the rewards of more effective cooperative group hunting could have a profound impact on the spider ecosystem. The workflow of the proposed PAO model is shown in Fig. 3.

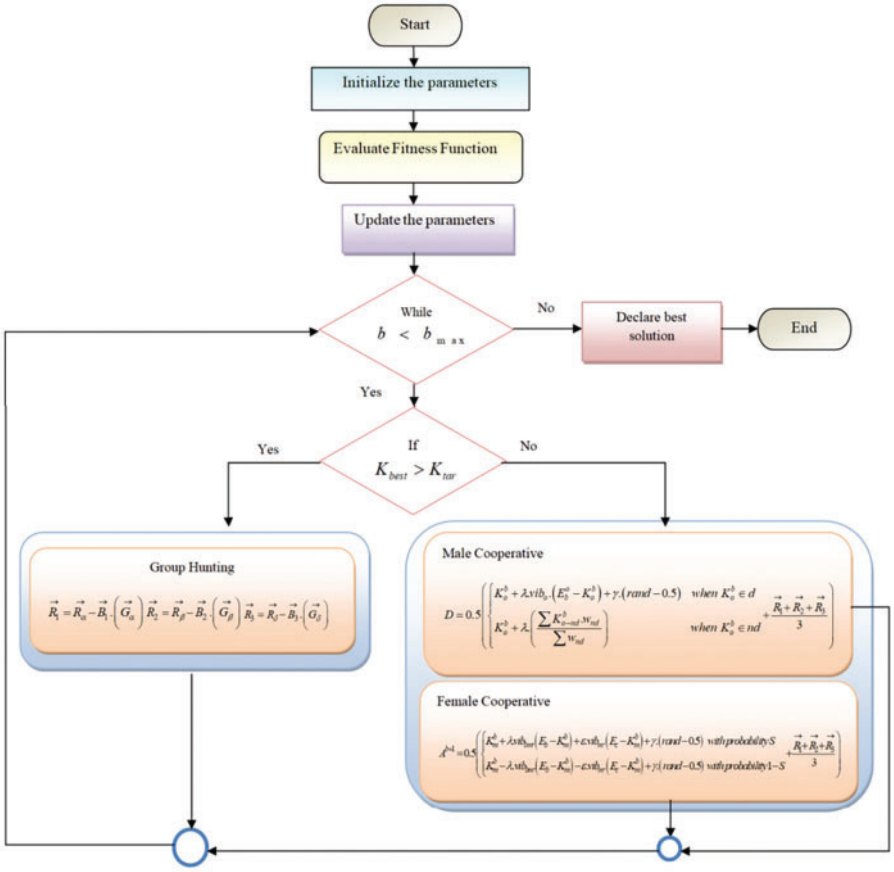

Figure 3: Workflow of PAO algorithm

Motivation: The idea behind the PAO model is based on combining characteristics borrowed from coyotes and social spiders to improve classifier performance by employing inspiration through Prairie Araneida optimization. This is an iterative process of tweaking hyperparameters, weights, and biases to reduce misclassification errors until the optimum configuration is found. The fact that the cooperative group hunting behaviors of a coyote were integrated into spider foraging indicates an innovative approach, where coordinated strategies allowed spiders to engage in collective action and tackle tougher prey. This cooperative approach saves energy, minimizes individual risk, and improves overall foraging performance.

4.1 Population Initialization

The PAO consists of random population initialization steps that intervene multiple times. PAOs are characterized by the predominance of females. Historically, the density of females K_m which is a fraction in percent values (from 65% to 90%) relative to the total K_v population was used. K_m is calculated through the use of this equation:

$$K_m = floor\left[(0.9 - rand\,(0,1) * 0.25) * K_v\right] \tag{13}$$

The male PAO count, represented as K_o is achieved by the difference of K_m and K_v.

$$K_o = K_v - K_m \tag{14}$$

4.2 Evalaution of Fitness

The competency level is measured as the ability of PAO being able to effectively carry out assigned tasks. In the specified strategy, each PAO is assigned its weight denoted as P_m characterizing the quality of a solution related to mth individual in K_v population. The fitness for separate individuals is calculated with the help of these formulas:

$$p_m = \frac{D\,(K_{vm}) - worst_{K_v}}{best_{K_v} - worst_{k_v}} \tag{15}$$

Here, we use $D\,(K_{vm})$ to denote the fitness value obtained from an evaluation of how good a PAO is at position K_{vm}. One can then employ this equation:

$$best_{K_v} = \max\left(D\,(K_{vm})\right) \tag{16}$$

$$worst_{K_v} = \max\left(D\,(K_{vm})\right) \tag{17}$$

4.3 Modeling of the Vibrations

The colony uses a communal web to communicate, with the vibrations varying based on the size and distance of the PAO producing them. The vibrations (*Egi*) coming from the node g are assumed to be individual approaching with equation:

$$Egi_{m,g} = p_g.a^{-\epsilon^2_{m,g}} \tag{18}$$

The formula determines the distance between PAO, which is represented by the symbols m and g.

$$x_{m,g} = \left\| B_m - B_g \right\| \tag{19}$$

The SSO (Social Spider Optimization) method considered three separate correlations, which correspond to three different vibrations:

a) Egj_{im} Vibration arises from information received by individuals $m\,(B_m)$ about members $i\,(B_m)$. i is the closest one to m with a higher weight.

$$Egi_{im} = p_i.a^{-x^2_{m,b}} \tag{20}$$

b) The Egj_{nm} Vibration is an individual $m\,(B_m)$ that breeds from the information given by the member $n\,(B_n)$ whereby; n signifies a secondary messenger with biggest body mass called p_n.

$$Egi_{nm} = p_n.a^{-x^2_{m,n}} \tag{21}$$

c) Egj_{ym} Vibration comes from the individual $m\,(B_m)$ as a result of information provided by member $m\left(B_y\right)$ where y stands for the closest female to m.

$$Egi_{ym} = p_y.a^{-x^2_{m,y}} \tag{22}$$

4.4 Initializing Population

In the initial phase, an iterative approach is employed that is comparable to the evolutionary algorithms used in previous PAO processes. It starts by randomly initializing the entire population, both male and female. This initialization starts by creating the collection B, which consists of K positions. Each PAO, marked as y_m *or* o_m, holds a position defined by means of a t-dimensional vector holding parameter values arranged for improvement. These parameter values are located within the initial upper bound for e_g^{high} and preliminary lower endpoint of e_g^{low}. The following equations outline this process:

$$y^c_{m,g} = e_g^{low} + rand\,(0,1) * \left(e_g^{high} - e_g^{low}\right) \qquad m = 1, 2, ...K_y \;\; and \;\; g = 1, 2, \ldots t \tag{23}$$

$$o^c_{m,g} = e_g^{low} + rand\,(0,1) * \left(e_g^{high} - e_g^{low}\right) \qquad m = 1, 2, ...K_O \; and \; g = 1, 2, \ldots t \tag{24}$$

The indexes are denoted by g and m represent individual variables, the $(rand\,(0,1))$ function generates a random number within the range of 0 to 1 as well as zero denotes the initial population.

4.5 Cooperative Operators

Factors that affect the cooperative behavior in PAO may include their gender, curiosity, and reproductive cycle in addition to random occurrences.

4.5.1 Integrating Phase

Incorporating group hunting behaviors in foraging could redesign their successful hunting practices. So, PAOs adopt coordinated group hunting strategies that allow them all to launch synchronized attacks on prey simultaneously with groups capable of taking down larger or more challenging targets collectively. This could be through strategic positioning; herding prey into positions that favor the end effect, and specialized roles within a group to provide them with an upper hand in capturing elusive or formidable game. This cooperative hunting model may also reduce energy use and lower the chance of injury to any single spider, leading toward increased foraging effectiveness as well as survival odds. Adopting such behaviors would necessitate evolutionary changes of a major scale, and the potential advantages that could be brought by increased collaborative group hunting might greatly influence PAO ecosystems.

4.5.2 Group Hunting

The hunting skills of PAO show amazing in identify and surround their prey while hunting, in this hunting dynamic, alpha usually takes a leading role sometimes accompanied by beta and delta. However, in the operation of a far-field search space where no optimal prey location is revealed, we assume that alpha beta and delta have better information about possible locations for potential prey. So, we keep the first three of, so far found best solutions and make other search agents including omegas modify their positions based on those. The following formulas clarify this strategic strategy, allowing a more efficient and harmonious search for the elusive best solution:

$$\vec{H}_{\alpha} = \left|\vec{m}_1.\vec{P}_{\alpha} - \vec{P}\right|, \vec{H}_{\beta} = \left|\vec{m}_2.\vec{P}_{\beta} - \vec{P}\right|, \vec{H}_{\delta} = \left|\vec{m}_3.\vec{P}_{\delta} - \vec{P}\right| \tag{25}$$

$$\vec{P}_1 = \vec{P}_{\alpha} - \vec{B}_1.\left(\vec{H}_{\alpha}\right), \vec{P}_2 = \vec{P}_{\beta} - \vec{B}_2.\left(\vec{H}_{\beta}\right), \vec{P}_3 = \vec{P}_{\delta} - \vec{B}_3.\left(\vec{H}_{\delta}\right) \tag{26}$$

Here, the symbol d represents the current iteration, while $\vec{B}$ and $\vec{m}$ are vectors containing coefficient values, $\vec{P}$ indicating the coyote's position in vector form.

4.5.3 Female Cooperative

The subsequent examples demonstrate how females can either attract or deter other spiders:

$$C = 0.5y_m^{b+1} + 0.5\vec{P}(d+1) \tag{27}$$

$$A = 0.5\left(\begin{cases} y_m^b + \delta.vib_{bm}.\left(B_b - y_m^b\right) + \varepsilon.vib_m.\left(B_b - y_m^b\right) \\ \quad +\gamma.(rand - 0.5)\, with\, probability\, eH \\ y_m^b - \delta.vib_{bm}.\left(B_b - y_m^b\right) - \varepsilon.vib_m.\left(B_b - y_m^b\right) + \\ \gamma.(rand - 0.5)\, with\, probability 1 - eH \end{cases} \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3}\right) \tag{28}$$

In this regard, b establishes the number of cycles, whereas $\delta, \varepsilon, \gamma$ is a random integer spanning $[0, 1]$, respectively. The B_b and B_n of the individual represents the top member of the population and the one with the highest weight.

4.5.4 Male Cooperative

Male individuals are divided into dominant and non-dominant groups based on their weights. The non-dominant individuals are drawn towards the weighted mean of the male population to learn about resources that are being used by dominant spiders. As a result, the spider position update is as follows:

$$B = 0.5o_m^{b+1} + 0.5\vec{P}(d+1) \tag{29}$$

$$D = 0.5\left(\begin{cases} o_m^b + \delta.vib_m.\left(B_y - o_m^b\right) + \rho.(rand - 0.5) \\ when\ F_{K_{y+m}} > F_{K_{y+m}} \\ o_m^b + \delta.\left(\dfrac{\sum_{q-1}^{uo} o_q^b.TS_{y+j}}{\sum_{j-1}^{uo} TS_{y+q}}\right) \\ when\ F_{K_{y+m}} \le F_{K_{y+o}} \end{cases} + \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3}\right) \tag{30}$$

In this scenario, it is insightful to point out that the individual B_y symbolizes the closest female character related to the male m and $\left(\frac{\sum_{j-1}^{uo} o_q^b.FK_{y+q}}{\sum_{q-1}^{uo} FK_{y+j}}\right)$ symbolizes the total mean characteristic of both populations in its entirety with much more accentuation being put on representatives of dominant individuals. The pseudo code of the proposed prairie Araneida optimization is depicted in Algorithm 1.

Algorithm 1: Pseudo code of the proposed Prairie Araneida optimization

S.No. Pseudocode

1 Population initialization: K

2 Assignation of fitness: $p_m = \dfrac{D\left(B_m\right) - worst_B}{best_B - worst_B}$

3 Modeling of the vibrations: The formula calculates the distance between PAO represented by m and g

$x_{m,g} = \left\| B_m - B_g \right\|$

4 **Population initializing:**

$y_{m,g}^c = e_g^{low} + rand\,(0,1) * \left(e_g^{high} - e_g^{low}\right) \qquad m = 1,2,....K_y \text{ and } g = 1,2,\ldots t$

$o_{m,g}^c = e_g^{low} + rand\,(0,1) * \left(e_g^{high} - e_g^{low}\right) \qquad m = 1,2,....K_O \text{ and } g = 1,2,\ldots t$

5 ***Cooperative Operators:***

i) **Hunting:** $\vec{P}_1 = \vec{P}_\alpha - \vec{B}_1.\left(\vec{H}_\alpha\right)\vec{P}_2 = \vec{P}_\beta - \vec{B}_2.\left(\vec{H}_\beta\right)$

ii) **Female coordinates:** $C = 0.5y_m^{b+1} + 0.5\vec{P}\,(d+1)$

6 **Male cooperative:**

i) $B = 0.5o_m^{b+1} + 0.5\vec{P}\,(d+1)$

$$D = 0.5\left(\begin{cases} o_m^b + \delta.vib_m.\left(B_y - o_m^b\right) + \rho.\,(rand - 0.5) \\ when\ F_{K_{y+m}} > F_{K_{y+m}} \\ o_m^b + \delta.\left(\dfrac{\sum_{q-1}^{uo} o_q^b.TS_{y+j}}{\sum_{j-1}^{uo} TS_{y+q}}\right) \\ when\ F_{K_{y+m}} \le F_{K_{y+o}} \end{cases} + \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3}\right)$$

7 **Termination:**

5 Result and Discussion

The PAO-CNN model creates an ID system, which is then evaluated for its effectiveness compared to other existing methods.

5.1 Experimental Setup

The ID experiment is carried out using Python with 8 GB of RAM and the Windows 10 operating system. The clock speed of the system is 3 GHz, the storage of the implemented system is 128 GB, and the RAM is 16 GB. Further, the implementation is carried out in PyCharm Software of the latest version 2024.2.1. The hyperparameters of the proposed model as well as the optimization are provided in the following Table 2.

Table 2: Hyperparameter details

Learning rate	0.01
Batch size	32
Epoch	500
Default optimizer	Adam optimizer
Proposed optimizer	Prairie araneida optimization
Fitness of the model	Categorical loss entropy
Fitness of the proposed optimization	Accuracy

5.2 Performance Metrics

The performance of the PAO-CNN model is evaluated with metrics such as accuracy, sensitivity, and specificity. Accuracy measures the overall effectiveness of the ID in correctly identifying both legitimate and malicious activities. It is calculated as the ratio of correctly identified instances to the total number of instances. A higher accuracy indicates that the IDS is generally reliable in distinguishing between normal and intrusive activities. Sensitivity measures the IDS's ability to correctly identify actual intrusions. It is the ratio of correctly identified intrusions to the sum of missed intrusions. Specificity measures the IDS's ability to correctly identify legitimate activities, avoiding false alarms. It is the ratio of correctly identified legitimate activities to the sum of incorrectly identified intrusions. High specificity indicates that the IDS is good at minimizing false alarms, ensuring that normal activities are not incorrectly flagged as intrusions.

5.3 Comparative Methods

The PAO-CNN model is compared with several benchmark classifiers such as SVM [39], BiLSTM [40], deep CNN [41], PSO-based deep CNN [42], SSO-based deep CNN [43], ASO-opt deep CNN [44], and KNN [45], to assess its performance.

5.3.1 Comparative Analysis Concerning TP for BoT-IoT

Fig. 4 visualizes the TP 90 metric, which serves as a basis for comparing the effectiveness of the PAO-CNN model against other comparative techniques. In Fig. 4a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 96.97% with a TP of 90, surpassing the ASO-opt deep CNN by 1.05%. Fig. 4b illustrates the ID sensitivity of the PAO-CNN model, which reaches a sensitivity of 97.09% with 90 TP, surpassing the ASO-opt deep CNN by 2.13%. Fig. 4c displays the ID specificity of the PAO-CNN, which achieved a specificity of 97.16% with a TP of 90. This surpassed the ASO-opt deep CNN by a notable margin of 0.57%.

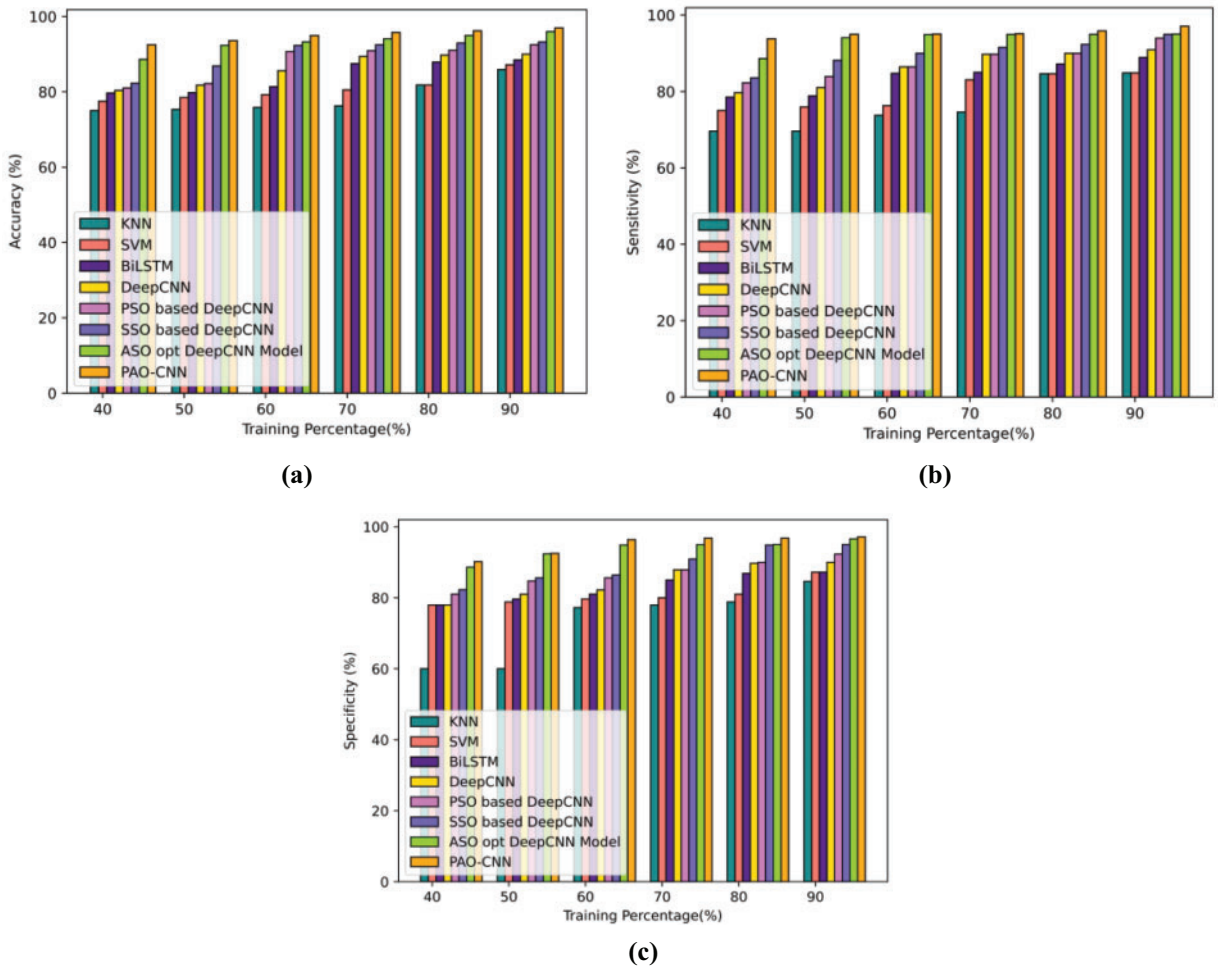

Figure 4: Comparative analysis based on TP for BoT-IoT (a) accuracy, (b) sensitivity, and (c) specificity

5.3.2 Comparative Analysis Concerning TP for CIC-IDS 2017

In Fig. 5a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 97.12% with a TP of 90, surpassing the ASO-opt deep CNN by 1.56%. Fig. 5b displays the ID sensitivity of the PAO-CNN model, which reaches a sensitivity of 97.98% with a TP of 90. This surpasses the ASO-opt deep CNN by 2.10%. Fig. 5c illustrates the ID specificity of the PAO-CNN, which achieves a specificity of 97.78% with a TP of 90. This is significantly higher than the ASO-opt deep CNN by 0.84%.

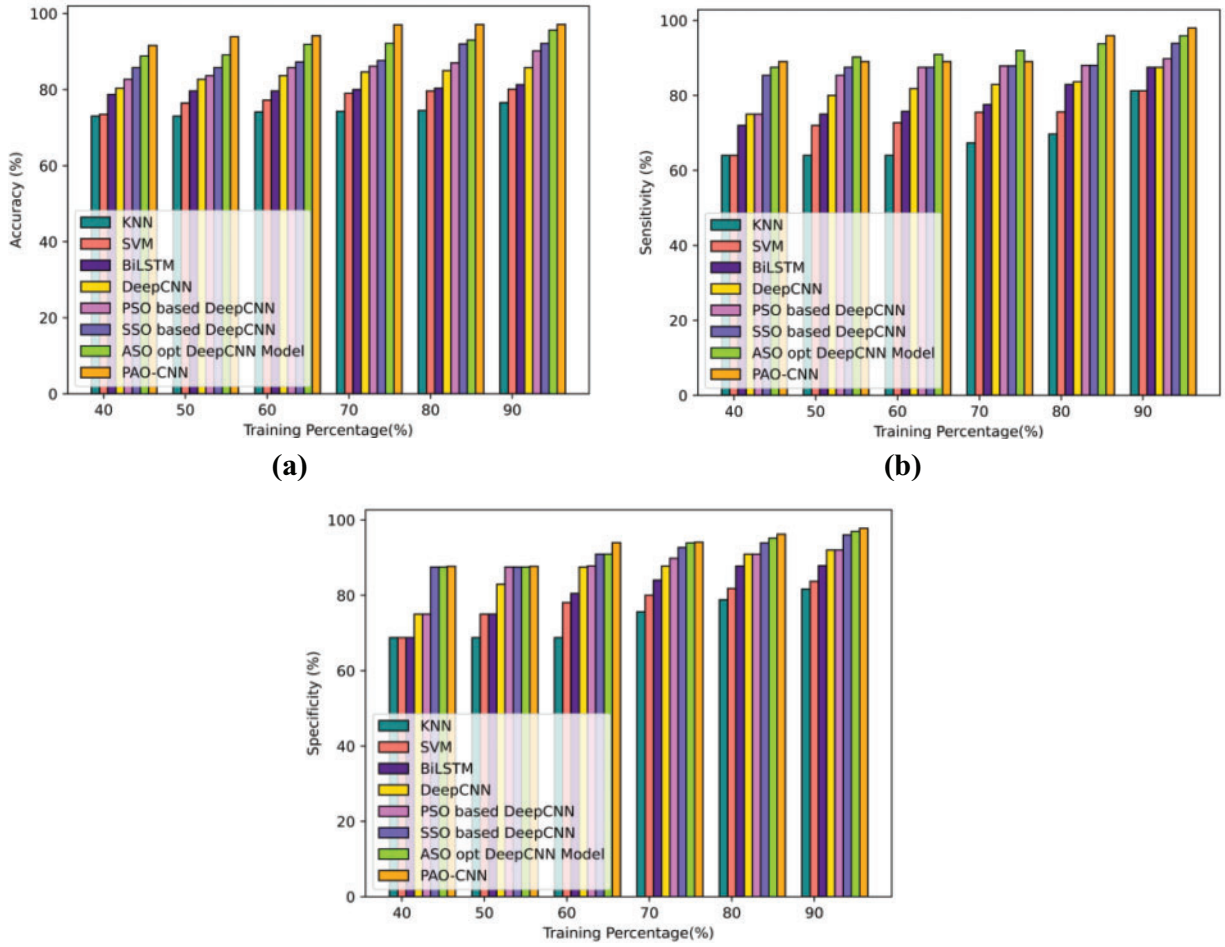

Figure 5: Comparative analysis based on TP for CIC-IDS 2017 (a) accuracy, (b) sensitivity, and (c) specificity

5.3.3 Comparative Analysis Concerning TP for UNSW-NB15

In Fig. 6a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 96.25% with a TP of 90, surpassing the ASO-opt deep CNN by 1.30%. Fig. 6b shows the ID sensitivity of the PAO-CNN model, which reaches 96.15% sensitivity with a TP of 90, slightly surpassing the ASO-opt deep CNN by 2.24%. Fig. 6c displays the ID specificity of the PAO-CNN. The PAO-CNN achieves a specificity of 98.20% with a TP of 90, outperforming the ASO-opt deep CNN by a significant difference of 2.24%.

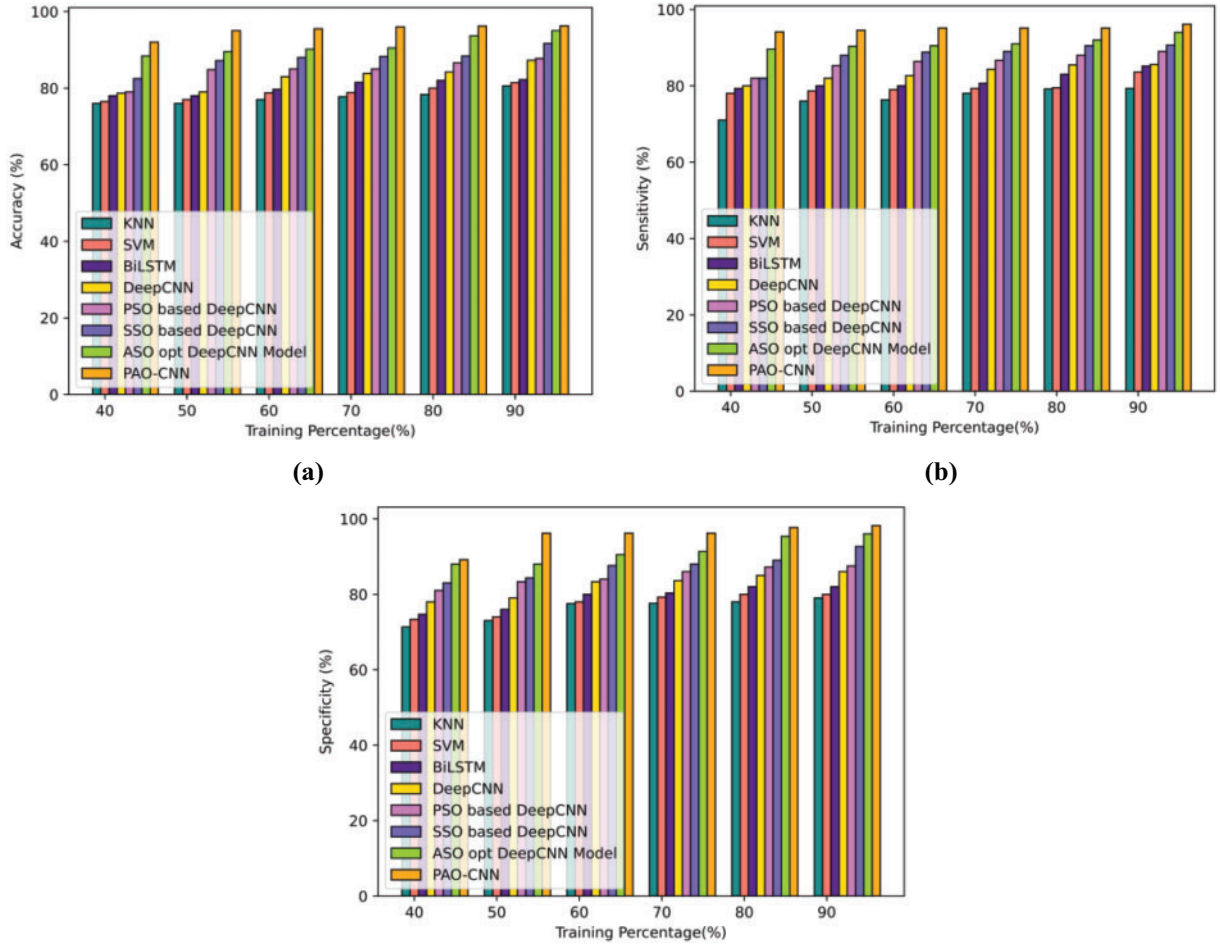

Figure 6: Comparative analysis based on TP for UNSW-NB15 (a) accuracy, (b) sensitivity, and (c) specificity

5.3.4 Comparative Analysis Concerning K-Fold for BoT-IoT

Fig. 7 visualizes the k-fold 6 metric, which serves as a basis for comparing the effectiveness of the PAO-CNN model against other comparative techniques. In Fig. 7a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 97.70% with a k-fold 6, surpassing the ASO-opt deep CNN by 3.05%. Fig. 7b illustrates the ID sensitivity of the PAO-CNN model, which successfully achieves a sensitivity of 96.80% with a k-fold 6. This outperforms the ASO-opt deep CNN by 1.29%. Fig. 7c displays the ID specificity of the PAO-CNN, highlighting its performance. The PAO-CNN achieves a specificity of 97.23% using a k-fold 6, outperforming the ASO-opt deep CNN by a significant margin of 4.23%.

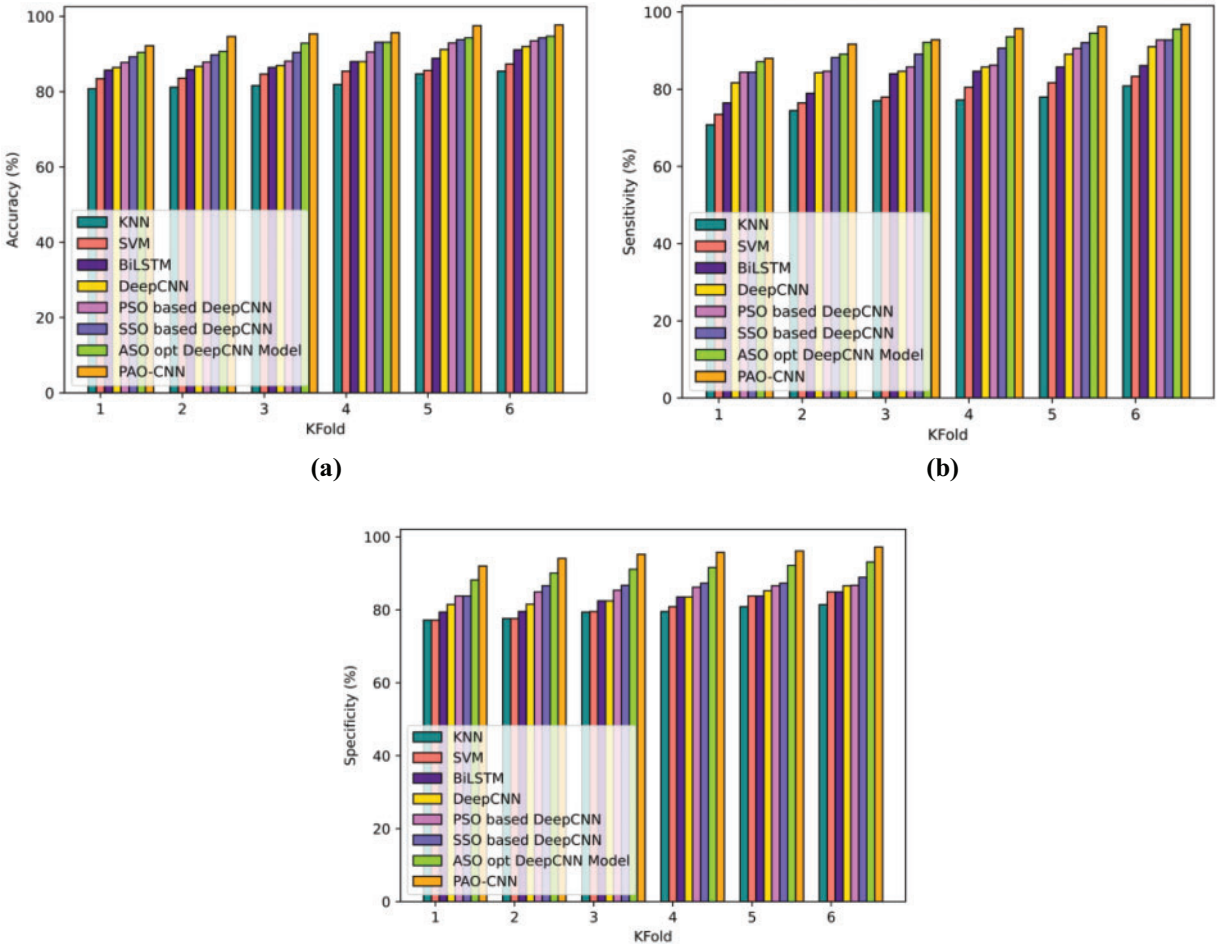

Figure 7: Comparative analysis based on k-fold for BoT-IoT (a) accuracy, (b) sensitivity, and (c) specificity

5.3.5 Comparative Analysis Concerning K-Fold for CIC-IDS 2017

In Fig. 8a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 97.75% with a k-fold 6, surpassing the ASO-opt deep CNN by 2.17%. Fig. 8b displays the ID sensitivity of the PAO-CNN model, which reaches 96.43% using k-fold 6. This outperforms the ASO-opt deep CNN by 0.62%. Fig. 8c displays the ID specificity of the PAO-CNN model. Using a k-fold 6, the PAO-CNN shows a specificity of 97.75%, which is 2.81% higher than the ASO-opt deep CNN.

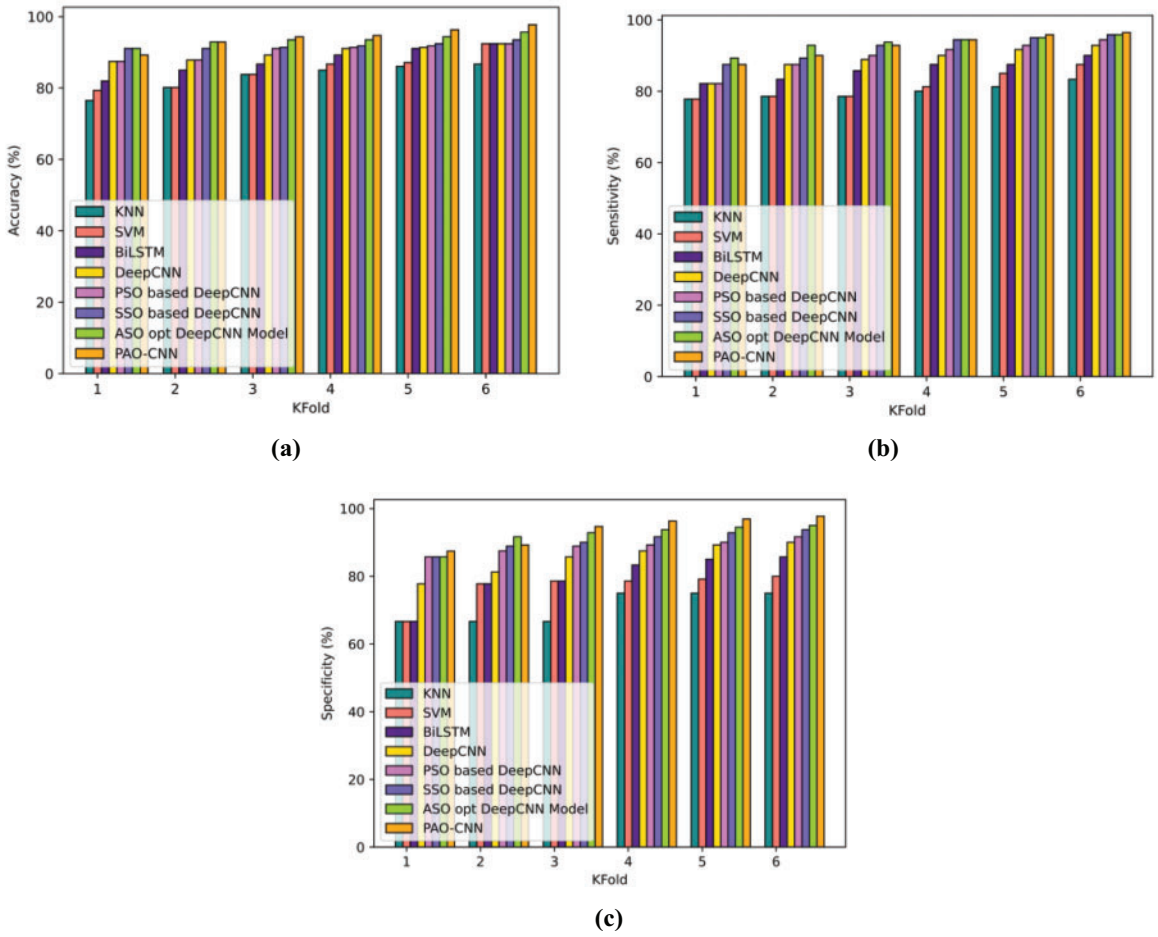

Figure 8: Comparative analysis based on k-fold for CIC-IDS 2017 (a) accuracy, (b) sensitivity, and (c) specificity

5.3.6 Comparative Analysis Concerning K-Fold for UNSW-NB15

In Fig. 9a, the ID accuracy of the PAO-CNN model is showcased. The PAO-CNN demonstrates an accuracy of 97.04% with a k-fold 6, surpassing the ASO-opt deep CNN by 2.00%. In Fig. 9b, the ID sensitivity of the PAO-CNN model is presented. The PAO-CNN achieves a sensitivity of 97.37% with a k-fold 6, edging ahead of the ASO-opt deep CNN by 2.43%. In Fig. 9c, the ID specificity of the PAO-CNN is showcased. With a k-fold 6, the PAO-CNN demonstrates a specificity of 96.48%, surpassing the ASO-opt deep CNN by a substantial margin of 0.50%.

5.4 Comparative Analysis on GD, MFE, Spacing, Spread, and Weighted Sum

Table 3 shows the comparison of the proposed method with other existing methods based on quality metrics such as generation distance (GD), maximal Perito front error (MFE), spacing, spread, and weighted sum. Based on the described quality metrics, the comparison is carried out, which evaluates the quality of the model as well as the combination of the proposed PAO algorithm into the model.

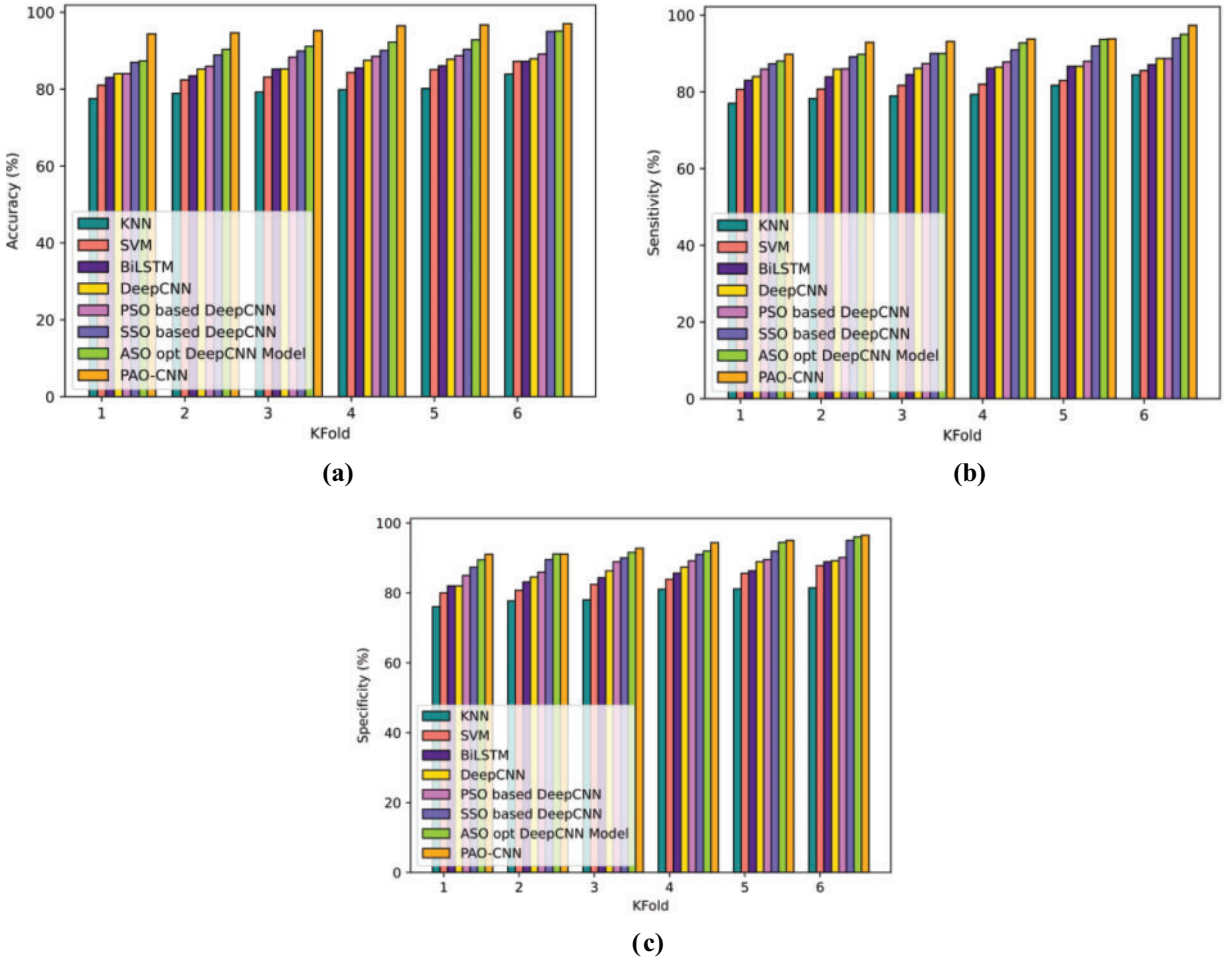

Figure 9: Comparative analysis based on k-fold for UNSW-NB15 (a) accuracy, (b) sensitivity, and (c) specificity

Table 3: Comparative analysis on GD, MFE, spacing, spread and weighted sum

Models	GD	MFE	Spacing	Spread	Weighted sum
PSO-based Deep CNN [42]	0.11	0.37	0.10	0.097	0.097
SSO-based Deep CNN [43]	0.09	0.29	0.06	0.057	0.057
ASO-opt Deep CNN Model [44]	0.06	0.26	0.04	0.038	0.038
PAO-CNN	0.02	0.20	0.03	0.027	0.027

5.5 Comparative Discussion

The proposed PAO-CNN model effectively addresses the existing challenges by utilizing the PAO technique, which integrates the cooperative group hunting behaviors into their foraging, it could completely change their hunt. Using synchronized group strategies, Prairie Araneida could work together to attack bigger or more challenging prey. This could include tactical placement, herding of

prey for more effective hunting opportunities, and certain roles that need to be played out to perform effectively in capturing difficult prey. This could save on energy, reduce individual risk levels, and increase overall productivity in foraging. These enormous rewards of cooperative group hunting would likely prompt substantial evolutionary adaptations. In the field of ID research, the table allows for a structured and organized manner in which to compare various techniques used by those attempting to administer successful system invasion detections and characterize the accomplishments made possible through the proposed model. The added comparative methods exhibited various challenges, which are described as follows, KNN was slow, especially with large datasets, as it required calculating the distance between the query and all training samples [45]. SVMs were computationally expensive, especially with large datasets, and required careful tuning of parameters like the kernel, regularization, and margin parameters [39]. BiLSTMs were slow to train due to their complexity and they overfit if not properly regularized, especially with small datasets [40]. Deep CNN was prone to overfitting, especially with small datasets. Further, the model required a large amount of labeled data and extensive tuning of hyperparameters [41]. PSO-deep CNN was computationally expensive and had increased complexity [3], Similar to PSO-deep CNN, ASO-deep CNN, and SSO-deep CNN exhibited the same drawbacks. Thus to tackle these limitations, the PAO-CNN model is proposed in the research. The comparative discussion of TP and K-fold analysis of PAO-CNN is depicted in Tables 4 and 5, respectively.

Table 4: Comparative discussion table for TP 90%

Model	TP 90%								
	BOT-IOT			CIC-IDS 2017			UNSW-NB15		
	Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)
KNN [45]	85.90	84.85	84.62	76.56	81.25	81.63	80.60	79.33	79.00
SVM [30]	87.18	84.85	87.18	80.07	81.25	83.67	81.40	83.60	80.00
BiLSTM [31]	88.46	88.89	87.18	81.27	87.50	87.88	82.20	85.20	82.00
Deep CNN [32]	90.00	90.91	90.00	85.75	87.50	92.00	87.25	85.60	86.00
PSO-based Deep CNN [33]	92.50	93.94	92.31	90.16	89.80	92.00	87.75	89.00	87.50
SSO-based Deep CNN [34]	93.22	94.95	95.00	92.12	93.88	96.00	91.67	90.67	92.67
ASO-opt Deep CNN Model [35]	95.95	95.00	96.61	95.61	95.92	96.96	95.00	94.00	96.00
PAO-Deep CNN	96.97	97.07	97.16	97.12	97.98	97.78	96.25	96.15	98.20

Table 5: Comparative discussion table for K-fold 6

Model	K-fold = 6								
	BOT-IOT			CIC-IDS 2017			UNSW-NB15		
	Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)
KNN	85.43	80.85	81.41	86.70	83.33	75.00	83.89	84.44	81.45
SVM	87.32	83.30	84.88	92.44	87.50	80.00	87.22	85.56	87.78
BiLSTM	91.07	86.12	84.88	92.44	90.00	85.71	87.22	87.10	88.89
Deep CNN	91.98	91.00	86.61	92.44	92.86	90.00	87.91	88.71	89.19
PSO-based Deep CNN	93.50	92.75	86.71	92.44	94.44	91.67	89.11	88.71	90.10
SSO-based Deep CNN	94.27	92.75	88.92	93.50	95.83	93.75	95.00	94.00	95.00
ASO-opt Deep CNN Model	94.71	95.55	93.12	95.63	95.83	95.00	95.09	95.00	96.00
PAO-Deep CNN	97.70	96.80	97.23	97.75	96.43	97.75	97.04	97.37	96.48

5.5.1 Limitations

Though the research attained high performance in the detection of intrusions, the model failed to analyze the Cybil attacks as well as various types of attacks, which remained the major drawback of the proposed model.

5.5.2 Future Scope

The proposed PAO-CNN model in the research of ID achieves high proficiency outcomes. Though the outcomes are highly efficient, the model can include advanced data correlation mechanisms as well as alert mechanisms. Further, the model can work with several advanced attack mechanisms and source determination algorithms.

6 Conclusion

In conclusion, the research aims to improve ID systems through its novel PAO-CNN model that shows better performance in combating dynamic cyber threats. A fused CNN is considered a significant contribution because statistical features are effectively integrated, enabling an in-depth analysis of intrusion datasets. Also, the innovative application of Prairie Araneida optimization for CNN optimization is one key advantage. Motivated by nature, these optimization layers make the model adaptive to changing data patterns, thus enabling it to deliver robust performance in an ever-changing cybersecurity scene. The responsiveness of the model's adaptive behavior in response to drift detection during testing highlights this. The research offers a practical, efficient solution minimizing unnecessary processing when no drift is detected and retraining where necessary. Finally, the combination of sophisticated CNN architecture and biological inspiration optimization methods forms a universal approach that makes the PAO-CNN model an effective tool for improving IDS in practical instances. The model's performance is assessed through metrics such as Accuracy, Sensitivity, and Specificity, particularly noteworthy during TP 90 where the values reach 96.25%, 96.15%, and 98.20% on the dataset. Additionally, in the context of k-fold 6, the achieved metrics stand at 97.04% for Accuracy, 97.37% for Sensitivity, and 96.48% for Specificity.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Nishit Patil, Shubhalaxmi Joshi. Data collection: Nishit Patil, Shubhalaxmi Joshi. Analysis and interpretation of results: Nishit Patil, Shubhalaxmi Joshi. Draft manuscript preparation: Nishit Patil. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets analyzed during the current study are available in the UNSW (https://research.unsw.edu.au/projects/bot-iot-dataset), UNB (https://www.unb.ca/cic/datasets/ids-2017.html) and Kaggle (https://www.kaggle.com/datasets/dhoogla/unswnb15) repository, accessed on 12 January 2023.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] D. Z. Jin, Y. Q. Lu, J. C. Qin, Z. Cheng, and Z. S. Mao, "SwiftIDS: Rbeal-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism," *Comput. Secur.*, vol. 9, 2020, Art. no. 101984.

[2] A. Y. Mohammed, H. L. Arash, A. Ali, and A. Ghorbani, "The sound of intrusion: A novel network intrusion detection system," *Comput. Electr. Eng.*, vol. 104, 2022, Art. no. 108455.

[3] A. Iram, Z. Ayub, F. Masoodi, and A. M. Bamhdi, "A machine learning approach for intrusion detection system on NSL-KDD dataset," in *2020 Int. Conf. Smart Electron. Commun.(ICOSEC)*, IEEE, 2020, pp. 919–924.

[4] M. S. Naderi and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, 2021. doi: 10.1007/s10207-020-00508-5.

[5] N. M. Tuan and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 113, pp. 418–427, 2020.

[6] J. W. Zhang, Y. Ling, X. B. Fu, X. K. Yang, G. Xiong and R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," *Comput. Secur.*, vol. 89, no. 11, 2020, Art. no. 101681. doi: 10.1016/j.cose.2019.101681.

[7] V. Abhishek and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wirel. Pers. Commun.*, vol. 111, pp. 2287–2310, 2020. doi: 10.1007/s11277-019-06986-8.

[8] K. Vikash, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Clust. Comput.*, vol. 23, pp. 1397–1418, 2020. doi: 10.1007/s10586-019-03008-x.

[9] B. B. Zarpelãoa, R. S. Mianib, C. T. Kawakania, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, 2017. doi: 10.1016/j.jnca.2017.02.009.

[10] M. Al-Quatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018. doi: 10.1109/ACCESS.2018.2869577.

[11] A. S. Sohal, R. Sandhu, S. K. Sood, and V. Chang, "A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments," *Comput. Secur.*, vol. 74, pp. 340–354, 2018. doi: 10.1016/j.cose.2017.08.016.

[12] W. Kong, J. Shen, P. Vijayakumar, Y. Cho, and V. Chang, "A practical group blind signature scheme for privacy protection in smart grid," *J. Parallel Distrib. Comput.*, vol. 136, pp. 29–39, 2020. doi: 10.1016/j.jpdc.2019.09.016.

[13] J. D. Ren, J. W. Guo, W. Qian, H. Yuan, X. B. Hao and J. J. Hu, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, 2019. doi: 10.1155/2019/7130868.

[14] P. Li and W. H. Zhou, "Hybrid intrusion detection algorithm based on k-means and decision tree," *Comput. Mod.*, vol. 37, pp. 12–16, 2019.

[15] T. A. Tchakoucht and M. Ezziyyani, "Multilayered echo-state machine: A novel architecture for efficient intrusion detection," *IEEE Access*, vol. 6, pp. 72458–72468, 2018. doi: 10.1109/ACCESS.2018.2867345.

[16] R. Yahaloma, A. Sterena, Y. Nameria, M. Roytmana, A. Porgadorb and Y. Elovici, "Improving the effectiveness of intrusion detection systems for hierarchical data," *Knowl.-Based Syst.*, vol. 168, pp. 59–69, 2019. doi: 10.1016/j.knosys.2019.01.002.

[17] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on optimal hybrid kernel extreme learning machine," *Knowl.-Based Syst.*, vol. 195, pp. 1–17, 2020. doi: 10.1016/j.knosys.2020.105648.

[18] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Syst. Appl.*, vol. 148, pp. 1–14, 2020. doi: 10.1016/j.eswa.2020.113249.

[19] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Comput. Netw*, vol. 148, pp. 164–175, 2019. doi: 10.1016/j.comnet.2018.11.010.

[20] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," *J. Ambient. Intell. Humaniz. Comput.*, vol 12, pp. 1–18, 2020.
[21] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection system for Internet of Things networks," *Future Internet*, vol. 12, no. 3, pp. 1–14, 2020. doi: 10.3390/fi12030044.
[22] B. B. Gupta, M. Misra, and R. C. Joshi, "FVBA: A combined statistical approach for low rate degrading and high bandwidth disruptive DDoS attacks detection in ISP domain," in *2008 16th IEEE Int. Conf. Netw.*, IEEE, 2008, pp. 1–4.
[23] I. H. Sarker, Y. B. Abushark, F. Alsohami, and A. I. Khan, "IntruDTree: A machine learning based cyber-security intrusion detection model," *Symmetry*, vol. 12, no. 5, pp. 1–15, 2020. doi: 10.3390/sym12050754.
[24] K. Chisholm, C. Yakopcic, M. S. Alam, and T. M. Taha, "Multilayer perceptron algorithms for network intrusion detection on portable low power hardware," in *10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2020, pp. 901–906.
[25] T. B. Prasad, P. S. Prasad, and K. P. Kumar, "An intrusion detection system software program using KNN nearest neighbors approach," *Int. J. Sci. Res. Innov. Eng. (IJSRIE)*, vol. 1, pp. 1–6, 2020.
[26] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets," *Secur. Commun. Netw.*, vol. 2020, no. 1, pp. 1–9, 2020. doi: 10.1155/2020/4586875.
[27] H. Chen, Z. D. Wang, S. X. Yang, X. Luo, D. J. He and S. Chan, "Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments," *Alex. Eng. J.*, vol. 111, pp. 78–91, 2025. doi: 10.1016/j.aej.2024.10.014.
[28] Z. Gou, M. A. B. Ahmadon, S. Yamaguchi, and B. B. Gupta, "A Petrinet-based framework of intrusion detection systems," in *2015 IEEE 4th Glob. Conf. Consum. Electron. (GCCE)*, IEEE, Oct. 2015, pp. 579–583.
[29] B. Gupta, D. P. Agrawal, and S. Yamaguchi, *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*. Hershey, PA, USA: IGI Global, 2016.
[30] E. Omar, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *J. Supercomput.*, vol. 79, no. 12, pp. 13241–13261, 2023. doi: 10.1007/s11227-023-05197-0.
[31] D. Jiawei, K. Yang, Y. Hu, and L. Jiang, "NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning," *IEEE Access*, vol. 11, pp. 24808–24821, 2023. doi: 10.1109/ACCESS.2023.3254915.
[32] Y. K. Saheed and S. Misra, "A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things," *Int. J. Inf. Secur.*, vol. 23, no. 3, pp. 1557–1581, 2024.
[33] Y. K. Saheed, T. O. Kehinde, M. A. Raji, and U. A. Baba, "Feature selection in intrusion detection systems: A new hybrid fusion of bat algorithm and residue number system," *J. Inf. Telecommun.*, vol. 8, no. 2, pp. 189–207, 2024.
[34] "BOT-IOT dataset," Accessed: Jan. 12, 2023. [Online]. Available: https://research.unsw.edu.au/projects/bot-iot-dataset
[35] "CICIDS2017 dataset," Accessed: Jan. 12, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html
[36] "UNSW-NB15 network dataset," Accessed: Jan. 12, 2023. [Online]. Available: https://www.kaggle.com/datasets/dhoogla/unswnb15
[37] P. Juliano and L. D. S. Coelho, "Coyote optimization algorithm: A new metaheuristic for global optimization problems," in *2018 IEEE Congr. Evol. Comput. (CEC)*, 2018, pp. 1–8.
[38] C. Erik, M. Cienfuegos, D. Zaldívar, and M. P. Cisneros., "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6374–6384, 2013.
[39] A. Alsarhan, M. Alauthman, E. A. Alshdaifat, A. R. Al-Ghuwairi, and A. Al-Dubai, "Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular *ad hoc* networks," *J. Ambient. Intell. Humaniz. Comput.*, vol. 14, no. 5, pp. 6113–6122, 2023.

[40] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao and F. Xu, "A network intrusion detection model based on BiLSTM with multi-head attention mechanism," *Electronics*, vol. 12, no. 19, 2023, Art. no. 4170.
[41] V. Hnamte and J. Hussain, "Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach," *Telemat. Inform. Rep.*, vol. 11, no. 9, 2023, Art. no. 100077. doi: 10.1016/j.teler.2023.100077.
[42] D. Kilichev and W. Kim, "Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO," *Mathematics*, vol. 11, no. 17, 2023, Art. no. 3724. doi: 10.3390/math11173724.
[43] W. Zhang, J. Hao, and F. Liu, "Effective social spider optimization algorithms for distributed assembly permutation flowshop scheduling problem in automobile manufacturing supply chain," *Sci. Rep.*, vol. 14, no. 1, 2024, Art. no. 6370.
[44] S. Vineeta and V. D. Kaushik, "DTCWTASODCNN: DTCWT based weighted fusion model for multi-modal medical image quality improvement with ASO technique & DCNN," *J. Sci. Ind. Res.*, vol. 81, no. 8, pp. 850–858, 2022.
[45] P. P. He, G. F. Feng, H. X. Li, and L. Yang, "A network intrusion detection method based on a fast KNN algorithm," *Acad. J. Sci. Technol.*, vol. 8, no. 2, pp. 81–83, 2023. doi: 10.54097/ajst.v8i2.14953.