ISSN 2723-6471

An Adaptive Cuckoo Search Algorithm with Deep Learning for Addressing Cyber Security Problem

J. Jeyaboopathiraja[1,*], Princess Mariajohn[2], Siti Sarah Maidin[3], Jing Sun[4]

[1]*Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore India*

[2]*Department of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore India*

[3]*Faculty of Data Science and Information Technology (FDSIT), INTI International University, Nilai, Malaysia*

[4]*Faculty of Liberal Arts, Shinawatra University, Thailand*

(Received: June 13, 2024; Revised: July 22, 2024; Accepted: September 31, 2024; Available online: November 7, 2024)

Abstract

IoT (Internet of Things) offers continued services to organizations by connecting systems, application and services using the medium of internet. They also leave themselves open to threats including virus attacks and software thefts where the risks of losing crucial information are high. These threats harm both the business' finances and reputation. This work offers a combined Deep Learning strategy using Artificial Neural Networks that can assist in detecting illegal software and malware tainted files. The proposed cyber security architecture uses data mining techniques to forecast cyber-attacks and prepare Internet of Things for suitable countermeasures. This framework uses two phases namely detections and predictions. This paper proposes Adaptive Cuckoo Search Optimization-based Algorithms for cloud network routes. Adaptive Cuckoo Search Algorithm are a bio-inspired protocol based on cuckoo birds' characteristics. Artificial Neural Networks classify assaults on cloud environments. The major goal of this work is to separate malicious servers from legitimate servers that are impacted by Denial of Service and Distributed Denial of Service assaults and thus safeguard server data and ensuring they are sent to legitimate servers. The outcome from this research proposed scheme shows better performances for protecting systems from cyber-attacks in terms of values for accuracy, Precision, Recall and F1-Measure when compared to existing algorithms.

Keywords: Internet of Things (IoT), Cyber-Attacks, Deep Learning, Artificial Neural Network (ANN), Adaptive Cuckoo Search Algorithm (ACSA), Denial of Service (DoS), Distributed Denial-of-Service (DDoS), Process Innovation, Inclusive Innovation

1. Introduction

The Internet of Things (IoTs) represents a cutting-edge technology that leverages the internet to connect various devices, enhancing personal, professional, and social aspects of life [1]. IoTs consist of networks of intelligent objects that can connect to the internet without human intervention but are vulnerable to cyber-attacks. To address this, networks employ Intrusion Detection Systems (IDS) to identify potential threats. Most modern IDS rely on machine learning (ML) algorithms, enabling them to learn and detect intrusions effectively [2].

Security vulnerabilities can arise from both external and internal sources. IDS systems analyze networks using three approaches: misuse-based detection, anomaly-based detection, and a combination of both. Misuse-based strategies detect attacks by recognizing patterns of known threats, helping to minimize false alarms when identifying known hazards and their variations [3]. Various types of cyber-attacks, along with technologies and procedures designed to protect computers, networks, software, and data from damage, unauthorized access, and other threats [4]. In the realm of cybersecurity, there have been significant advancements in technical and operational aspects, particularly through developments in data sciences [5]. ML, a key component of artificial intelligence (AI), plays a crucial role in enhancing these capabilities [6]. The field of data science is enabling a new paradigm where ML approaches can transform cybersecurity environments [7]. In grid and cloud computing systems, several heuristic algorithms have been

***Corresponding author: J. Jeyaboopathiraja (jeyaboopathi@gmail.com)**
DOI: https://doi.org/10.47738/jads.v5i4.366

 ISSN 2723-6471

developed, including the Cuckoo Search Algorithm (CSO) [8], Genetic Algorithm (GA) [9], Particle Swarm Optimization (PSO) [10], and Ant Colony Optimization (ACO) [11]. These evolutionary ML algorithms have been utilized to detect threats and develop cybersecurity strategies. Manual processes for identifying and mitigating cyber-attacks are often inefficient, necessitating automated models that can recognize and address threats effectively [12].

This research introduces a framework that analyzes historical network data using data mining techniques to identify patterns associated with cyber-attacks. These patterns are then used to predict potential future attacks, providing a proactive approach to cybersecurity. The proposed architecture can be deployed on live networks to forecast cyber-attacks. The contributions of this research, which leverages historical data, include: (1) identifying patterns of cyber-attacks, (2) utilizing these patterns to predict future attacks on live networks, and (3) collaborating with management to implement effective countermeasures to mitigate cyber threats.

2. Literature Review

Rahman [13] proposed a cybersecurity framework utilizing data mining techniques to predict and mitigate cyber-attacks. The framework comprises two main components: detection and forecasting of cyber-attacks. It employs J48 decision trees and predictive models to identify potential threats and uncover patterns associated with past attacks. The framework was evaluated using a dataset from the Canadian Institute of Cyber Security, which included various attack types such as DDoS, botnets, port scans, SQL injections, brute force, Heartbleed, and others. The proposed model demonstrated high accuracy, achieving a 99% success rate in predicting cyber-attacks, indicating its effectiveness for future threat detection.

Das and Morris [14] explored ML approaches in cyber analytics, including their application in email filtering, traffic classification, and intrusion detection. They reviewed each method based on its effectiveness and citation frequency, identifying prominent datasets that are essential for ML research. Four ML algorithms were tested using MODBUS data from a gas pipeline, and the performance of each method was analyzed based on their ability to classify different types of attacks.

Sarker [15] extensively discussed the application of ML algorithms in automating cybersecurity processes and intelligent data analysis. The study emphasized the ability of ML to extract valuable insights from cyber data, exploring real-world scenarios where data-driven intelligence, automation, and decision-making enable a more proactive defense compared to traditional methods. The study also highlighted potential future applications of ML in cybersecurity and the areas requiring further research and development.

Sari et al. [16] proposed a network monitoring tool that acts as a countermeasure against cyber-attacks using AI algorithms. The framework utilized classification and prediction techniques based on user behavior, integrating new rules into existing systems. The study employed the Gaussian Mixture Model (GMM) for enhanced effectiveness, focusing on behavioral similarities rather than geographical proximities, as seen in traditional algorithms like k-means.

HB, B. G. [17] applied deep neural networks (DNN) for attack detection across three cybersecurity use cases: Android malware classification, incident detection, and fraud detection. Each dataset contained real samples of both malicious and benign activities. By conducting multiple tests on network parameters and architectures, the study identified an effective DNN configuration, which was tested over 1000 epochs with learning rates ranging from 0.01 to 0.5. The results showed that DNNs outperformed traditional machine learning algorithms in all tested scenarios, offering more accurate identification of dataset characteristics.

Ahsan [18] proposed ML strategies to enhance cybersecurity, noting that as internet usage increases, more features and observations are incorporated into cyber incident data, making traditional defense mechanisms less effective. ML applications are evolving to promptly detect and prevent cyber threats, and combining different ML methods can enhance risk elimination. The study emphasized the need for integrating ML approaches to develop intelligent, data-driven systems that can improve service quality and strengthen security.

Ben Fredj [19] investigated DL approaches for predicting cybersecurity attacks, introducing models based on Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and Multi-Layer Perceptrons (MLPs). Using the

 ISSN 2723-6471

CTF dataset, the models were validated, with the LSTM models achieving a high F-measure value of 93%, indicating their robustness in forecasting potential attack occurrences [20].

Kadena and Gupi [21] highlighted the importance of human factors in cybersecurity, discussing typical threats and their impacts. They examined behavioral approaches, stressing how understanding human behavior is crucial to improving cybersecurity, and explored theories explaining how human factors influence security.

Roy, A. [22] proposed the Assault Countermeasure Tree (ACT), a defense mechanism using both tree and leaf nodes. Their probabilistic analysis provided a comprehensive evaluation of the costs associated with attacks and security measures, system risks, impacts of potential attacks, as well as returns on investment (ROIs) and returns on assets (ROAs). The study employed various analytical tools, such as Birnbaum importance measures, structural minimum cuts, and risk evaluations, to determine the most effective countermeasures without the need for a state-space model. A real-world case study involving a SCADA attack demonstrated the effectiveness of ACT.

Kavallieratos [23] developed a method for analyzing risk propagation and aggregation in complex cyber-physical systems (CPS). They introduced an automated approach using evolutionary programming to select optimal cybersecurity controls from available options, reducing costs and residual risks associated with implementation. The methodology was applied to the navigational systems of two types of Cyber-Enabled Ships (C-ES): autonomous and remotely controlled ships. The study's outcome was a set of cybersecurity measures applied to system components identified as vulnerable, effectively minimizing residual risk.

3. Proposed Methodology

This study offers an IDS architecture for networks based on anomaly detection and the integration of ACSAs and ANNs. Dangerous attacks such as DoS/DDoS and black hole attacks have been found. Initially, CSO is used to optimize the routes between source and destination servers based on fitness, where the fitness functions select optimal servers for communication, as shown mathematically as:

$$\mathrm{F_s(True) if\ F_s} \geq \mathrm{F_t} \tag{1}$$

$$\mathrm{Fitness} = \{\mathrm{F_t(False)\ otherwise}\} \tag{2}$$

Where F_s are selected behaviours of cuckoos andF_t imply behaviour thresholds of cuckoos which are computed on the basis of cuckoos' average values.

3.1. Anomaly Detections using ACSAs

A schematic representation of the development of the intrusion detection model is shown in figure 1. The gathering of network flow -related data results in the creation of a data set. The superfluous and dated functionality of the dataset is then turned off using ACSAs. After assigning the relevant functionality to the Core Vector Machine, the anomaly detection model [22] is created. This anomaly detection model finds anomalous or attacker packets, and depending on whether the packet is on the network, either a regular or an attack warning signal is delivered. The intrusion prevention mechanism then takes preventative action to secure the system in response to the warning message from the anomaly detections.

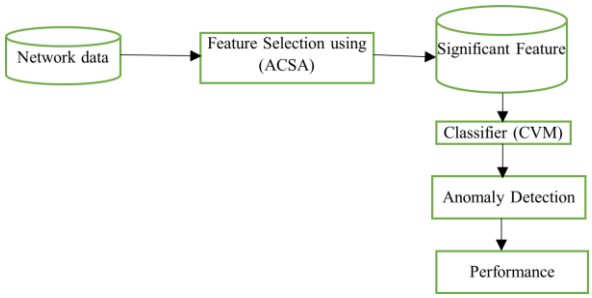

Figure 1. Schematic Diagram of Anomaly Detection System (ADS) Development

 ISSN 2723-6471

3.2. Cuckoo Search Optimization Algorithm

CSO is a meta-heuristic algorithm that was created by modelling cuckoos' sophisticated breeding behaviors. A population-based search method known as the cuckoo search algorithm is used to resolve complex cyber-attacks. The cuckoo bird's conduct of laying her eggs in the nests of other host birds served as the model for the algorithm. The cuckoos have exceptional skills, such as choosing nests that have recently given birth, removing eggs that are already present, and determining the likelihood that their eggs will hatch. Assuming that the eggs in their nest are their own, the hosts tend to them when specific birds detect them, fight against parasitic cuckoos, and destroy any discovered foreign eggs before constructing new nests in other locations. Every egg in the nest denotes a solution, and the cuckoo's egg denotes a fresh approach. Levy fly processes are used while creating new solutions. The following are the three ideal rules for the cuckoo search algorithm:

1) Cuckoos lay eggs one at a time and place them in randomly selected nests.
2) The best eggs in the best nests will be passed down to the next generations.
3) There are chances that hosts may find alien eggs despite accessible counts of host nests where hosts birds have two options: either throw the eggs out of the nests or leave the nests to build new ones.

The figure 2 illustrates a comprehensive framework for detecting and classifying anomalies within network data using machine learning techniques. The process begins with a dataset sourced from three cities: Chicago, San Francisco, and Philadelphia, which serves as the primary input. Initially, the data undergoes preprocessing, where it is cleaned and transformed to ensure suitability for analysis. A key part of this step involves a feature extraction technique known as the ACSO, which identifies and extracts relevant features from the dataset. These features are crucial for highlighting significant data patterns that aid in distinguishing normal behavior from potential anomalies.

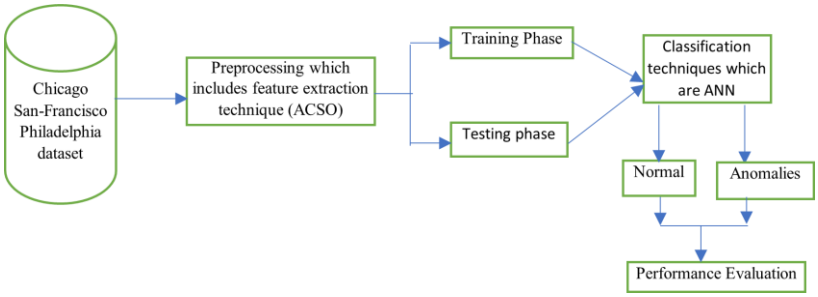

Figure 2. Proposed System Block diagram

Following preprocessing, the data is split into two parts: training and testing. During the training phase, ANN are employed to learn patterns within the data, enabling the model to differentiate between normal and abnormal behaviors. The trained model is then tested on new data in the testing phase to evaluate its ability to generalize and accurately identify deviations. The classification process produces two outputs: "Normal" and "Anomalies." Normal instances represent expected network behavior, while anomalies indicate unusual patterns that may signify potential security issues. The final step involves performance evaluation, where the accuracy and reliability of the ANN model are assessed, ensuring the system can effectively and robustly detect anomalies in real-world applications. This framework thus provides a proactive approach to identifying irregularities, contributing to enhanced cybersecurity and network integrity.

Randomly distributed initial populations of $M = [X_1; X_2; X_3; \ldots X_m]$ solutions or locations of host nests are generated. The solutions *X* are represented by D-dimensional vectors. Cuckoos randomly choose host nests to lay their eggs using Levy flight random walks and given in equations (3) and (4).

$$V_{pq}^{t+1} = V_{pq}^{t} + S_{pq} * \text{Levy}(\lambda) * \alpha \tag{3}$$

$$\text{Levy}(\lambda) = \left| \frac{\Gamma(1+\lambda) * \sin\left({}^{(\pi * \lambda)}/_{2}\right)}{\Gamma\left({}^{(1+\lambda)}/_{2}\right) * \lambda * 2\left({}^{(\lambda-1)}/_{2}\right)} \right|^{1/\lambda} \tag{4}$$

 ISSN 2723-6471

Where p, fe{1,2, ... m} and q ∈ {1,2, ... , D} are indices drawn at random from p. m is the total population of host nest sites, and D is the number of parameters that need to be optimized. t is the number of the current generation, λ refers to constants ($1 \leq \lambda \leq 3$), and α implies random integers created between [-1,1]. The step size is s>0 as well. The new solution produced will be too far from the previous solution if s is set to a value that is too large. If the adjustment is too minor, it won't make a difference and the search will be ineffective. Therefore, choosing the right step size is crucial to continuing the search as effectively as feasible. Consequently, Eq. (5) is used to compute the step size.

$$S_{pq} = V_{pq}^{t} - V_{fq}^{t} \quad (5)$$

Using Eq. (5) cuckoos select and evaluate host nests for laying eggs. Host birds identify alien eggs with probabilities associated to egg qualities using Eq. (6).

$$Pro_{p} = (0.9 * F(1)/\max(Fit)) + 0.1 \quad (6)$$

where $F(1)$ implies fitness values of solutions p which are proportional to eggs' qualities in nest positions p and pro_p indicates the probability of cuckoo eggs' survivals [23]. Host birds recognize alien eggs in a probability $P_a \in [0,1]$ selected at random is larger than prop. The host bird either kills the foreign egg or abandons the nest, and the cuckoos use Eq. (7) to locate a new host's nest (in a different location) for laying an egg. If not, the egg develops and will remain alive for the following generation according to the fitness function.

$$X_{p} = X_{p_{min}} + rand(0,1) * \left(X_{p_{max}} - X_{p_{min}}\right) \quad (7)$$

3.3. ACSAs

Customization is done to improve the performance of the default cuckoo search algorithm. A living thing's foraging route typically resembles a random walk since the next step depends on the present position and the likelihood that the next location will vary. When random walks are used their step lengths are determined by Levy distributions and Levy flights used. In this idea, when coming up with fresh solutions V_{pq}^{t} for cuckoos i, levy flights are integrated with inertia weights w, which control search abilities are given by Eq. (8).

$$V_{pq}^{t+1} = w * V_{pq}^{t} + S_{pq} * Levy\ (\lambda) * \alpha \quad (8)$$

Higher w values have greater global search abilities whereas smaller w have better local search abilities. The inertia-weighing element in the equation above might either be a fixed inertia weight or an inertia that changes over time. When dynamically shifting inertia weight is employed instead of constant inertia weight, the customised cuckoo search algorithm performs better. Eq. (9) contains the mathematical equation for the inertia weight that fluctuates dynamically.

$$M = -\frac{(w_{MAx} - w_{Min})}{N_{iter}} * i \quad (9)$$

Where w_{MAx} implies initial weights,w_{Min} refers to final weights,N_{iter} represents max iterations with current iterations as t.

The figure 3 illustrates a flowchart depicting a hybrid optimization process that integrates metaheuristic techniques, specifically Cuckoo Search (CS) and Iterated Local Search (ILS). The process begins by initializing a population of solutions, referred to as host nests, which are randomly generated and assigned as the current best solutions. To introduce diversity and explore the search space, some of these solutions undergo random fragmentation, and their positions are adjusted using Levy Flight random walks. This technique helps in exploring different areas of the solution space by introducing controlled randomness.

Next, the fitness of each solution is evaluated, and tasks are randomly selected from the pool of available options. A decision is made by comparing the fitness values of selected solutions. If a solution exhibits a better or equal performance compared to another, it is retained; otherwise, the inferior solution is replaced with a new task. The algorithm periodically abandons a proportion of solutions with the worst fitness scores, replacing them with new sets to maintain diversity within the population.

ISSN 2723-6471

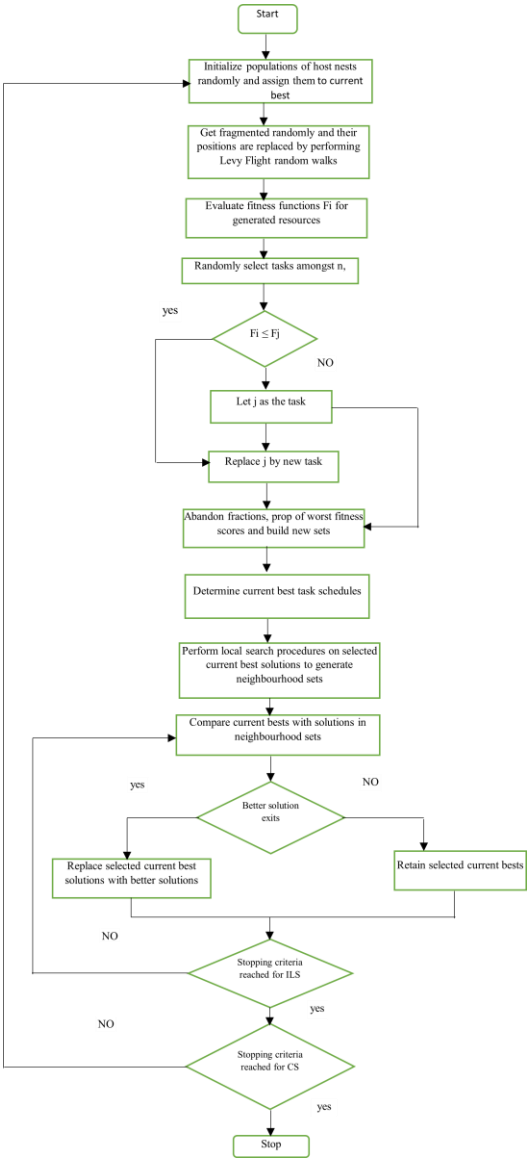

Figure 3. Flowchart of the proposed ACSAs

As the process continues, the algorithm determines the current best task schedules and performs local search procedures on these best solutions to generate neighboring sets, refining the search within a smaller region. It then compares these improved solutions with the current best to identify possible enhancements. If a better solution is found, it replaces the current best; otherwise, the existing solution is retained. The algorithm employs stopping criteria for both ILS and CS, evaluating whether the optimization process has converged. If the criteria for either technique are met, the process ends, and the final set of optimal solutions is produced. Through this iterative cycle of exploration, evaluation, and refinement, the hybrid optimization framework efficiently searches for and identifies high-quality solutions.

The standard Cuckoo Search algorithm faces limitations in controlling step sizes during the iteration process, which can impede its ability to effectively reach global minima or maxima [24]. To address this, an Adaptive Cuckoo Search Algorithm (ACSA) has been developed. Unlike the conventional approach, ACSA dynamically adjusts the step size instead of relying on a fixed value that corresponds to the physical dimensions of each host nest or the configuration of the search space. The algorithm starts by randomly initializing a set of guest nests and evaluating their fitness, with the aim of identifying the best and worst performers in the current generation. During each iteration, step sizes are computed based on a formula that accounts for the relative fitness of the solutions, allowing for more precise adjustments. The positions of the cuckoo nests are then updated using these adaptive step sizes, followed by an evaluation of their fitness. The algorithm incorporates a mechanism to abandon the worst-performing nests with a probability, replacing them with new ones to maintain diversity. The iterative process continues until a specified

 ISSN 2723-6471

stopping criterion is met, ensuring that the algorithm converges efficiently with fewer repetitions compared to the standard approach.

In conjunction with this optimization method, Artificial Neural Networks (ANNs) can be effectively integrated without the need for complex heuristics, setting them apart from traditional decomposition methods. ANNs are trained on diverse datasets, enabling them to accurately detect anomalies or potential attacks across a vast range of data points. During the training process, data points are selected randomly from all classes to ensure balanced learning. The role of ANNs is to classify and identify patterns from data collected across different locations, such as Chicago, San Francisco, and Philadelphia. A clustering mechanism is utilized to optimize the selection of data points, with the cuckoo search method aiding in determining optimal cluster centroids. This enhances the efficiency of the training process by ensuring that the neural networks learn from well-distributed and representative data samples. By training the model on clustered data points, ANNs can effectively identify patterns and anomalies, making the combination of ACSA and ANNs a robust and efficient approach for optimizing solutions and analyzing complex datasets.

3.4. Applications of DL in ANNs

3.4.1. Deep Learning

The fundamental structure of a DL model, as illustrated in figure 4, consists of three main layers: input, hidden, and output layers. Each layer receives and processes information from the preceding layers. Convolutional Neural Networks (CNNs) are a type of DL model that have been extensively used in fields such as computer vision and natural language processing [25]. In the context of cybersecurity, raw data, including potential attack patterns, can be directly fed into a CNN model without extensive pre-processing. The CNN then employs convolutional operations to analyze and extract key features from this data. This ability to process large volumes of data efficiently makes deep learning methods highly suitable for cybersecurity applications.

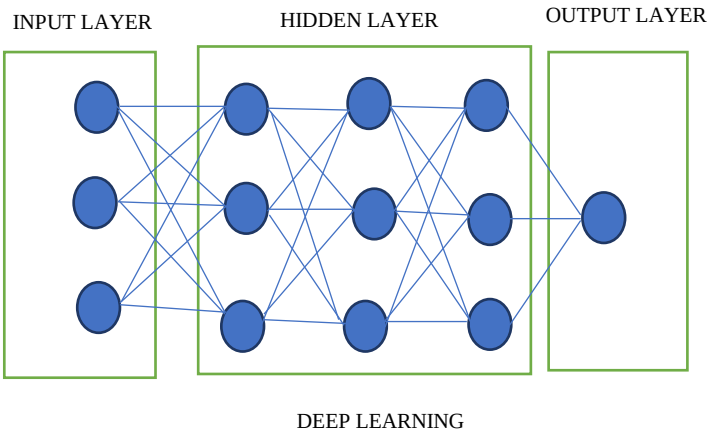

Figure 4. Architecture of DL Models

Deep learning represents an advanced evolution of traditional machine learning, with the capability to extract optimal feature representations from raw input samples. It has been successfully applied across various scenarios within cybersecurity, demonstrating its effectiveness in detecting and classifying threats. Another notable DL model is the RNN, which has shown promise in text processing and natural language processing (NLP) tasks [26], [27], [28)]. RNNs, particularly their enhanced versions known as LSTM networks, are capable of learning patterns in sequential data. This makes them effective for categorizing and classifying cyber threats, as they can identify patterns within data sequences and accurately distinguish between different types of attacks [29].

3.4.2. ANN

Artificial Neural Networks (ANNs) are computational models inspired by the structure and function of neurons in the human brain [30]. Their architecture consists of interconnected processing units that collaborate to solve specific tasks. Within this architecture, multiple processing units work together to tackle complex problems, much like neurons in a biological system. This study employs ANNs to classify servers, distinguishing between intrusion (attacker) servers and regular (normal) servers based on optimized server attributes. The optimization process leverages the ACSA, which

 ISSN 2723-6471

enhances the basic Cuckoo Search algorithm by incorporating adaptive mechanisms. These mechanisms include dynamic parameter adaptation, self-adaptive control, and adaptive population size. Dynamic parameter adaptation adjusts algorithm parameters, such as step sizes and probability values, during runtime based on the characteristics of the problem or the progress of the search. Self-adaptive control parameters update automatically based on the algorithm's performance in previous iterations, while adaptive population size modifies the number of solutions depending on convergence status or solution diversity.

In the proposed method, ANNs are initialized using training data that consists of optimized server attributes. The training process, outlined as Algorithm 2, begins by configuring the ANNs in MATLAB with parameters including 50 neurons, 10 hidden layers, and 100 epochs. The training data is categorized into groups, distinguishing between attacker nodes and legitimate (normal) nodes. During the initialization phase, the new command is used to set up the ANNs, which are then trained using the train function. The trained model classifies servers by analyzing cloud server properties to determine whether each server is normal or indicative of an attacker. Through this approach, the ANNs can accurately identify patterns and classify servers based on their attributes. The final output is a list of servers, clearly distinguishing between normal and attacker servers. This integration of ANNs and ACSA provides an efficient and adaptive method for intrusion detection, enabling proactive identification and management of cybersecurity threats.

4. Results and Discussion

This section discusses the dataset used for the system implementation, the proposed classification DL algorithms, and the proposed system are also discussed.

4.1. Datasets

The initial benchmark dataset for the IDS was compiled from data collected in Philadelphia, Chicago, and San Francisco. Various types of attacks were modeled and categorized under the heading of anomalies. For this study, two types of network traffic were analyzed: normal traffic and anomalous (attack) traffic. The San Francisco dataset, available at https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry, contains 2,142,685 records of crime incidents that occurred between January 1, 2003, and November 8, 2017. The dataset from Chicago, accessible at https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2, includes 5,541,398 records of criminal occurrences from 2003 to 2017. Additionally, the Philadelphia dataset, available at https://www.opendataphilly.org/dataset/crime-incidents, comprises 2,371,416 entries spanning from January 1, 2006, to December 31, 2017. When combined, these datasets include 41 features, each labeled to indicate whether the entry represents normal activity or an attack, along with the specific type of attack.

Several methods can be employed to enhance the performance of IDS by combining optimization and classification algorithms. These include PSO with Decision Tree (PSO+DT) [31], PSO with K-Nearest Neighbors (PSO+KNN) [3)], as well as ACSA integrated with Core Vector Machine (ACSA+CVM) and Artificial Neural Networks (ACSA+ANN) [32], [33]. These hybrid approaches leverage the strengths of each method, enabling more accurate and efficient detection of anomalies in network traffic.

4.2. Performance Evaluation

Metrics like Accuracy, Precision, Recall, F1-Measure Score, and others can be used to assess the effectiveness of classifiers [34], [35]. The foundation for computing various parameters is a confusion matrix. The number of occasions that a classification model properly or mistakenly estimates can be tabulated using a confusion matrix. The True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) values, which represent the equation from (10) to (13), are often used to depict the confusion matrix [36].

TP: Denotes instances which are prognosis as normal appropriately.

FN: Denotes incorrect prognosis for occurrences, it identifies occurrences that attack in real-world as normal.

FP: This springs a clue of the quantity of identified attacks that are normal in existence.

TN: This entails occurrences that are appropriately identified as an attack.

Accuracies: Ratios of successfully categorized data to total data.

 ISSN 2723-6471

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{FP} + \text{TN} + \text{TP} + \text{FN}} \quad (10)$$

Recalls (Sensitivities): They give counts of patients accurately identified with ASDs.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

Precisions: Ratios of patients correctly identified from ASDs out of all patients suffering from ASDs.

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \quad (12)$$

F-measures (F-scores/F1-scores): They represent harmonic-means of sensitivities and precisions expressing overall successes.

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \quad (13)$$

The results of the proposed method (ACSA+ANN) were compared with the precision results of other known other methods are illustrated in figure 5. The proposed ACSA+ANN gives higher precision results of 96.22%, whereas other methods such as PSO+DT, PSO+KNN and ACSA+CVM also gives higher precision for proposed ACSA+ANN based different methods such as 91.15%,92.21%,95.09% and respectively for San-Francisco, Chicago, and Philadelphia dataset. The proposed algorithm gives highest results than the existing methods respectively.

The results of the proposed method (ACSA+ANN) were compared with the recall results of other known other methods are illustrated in figure 6. The proposed ACSA+ANN gives higher recall results of 96.33%, whereas other methods such as PSO+DT, PSO+KNN and ACSA+CVM also gives higher recall for proposed ACSA+ANN based different methods such as 89.22%,92.19%,94.01% and respectively for San-Francisco, Chicago, and Philadelphia dataset. The proposed algorithm gives highest results than the existing methods respectively.

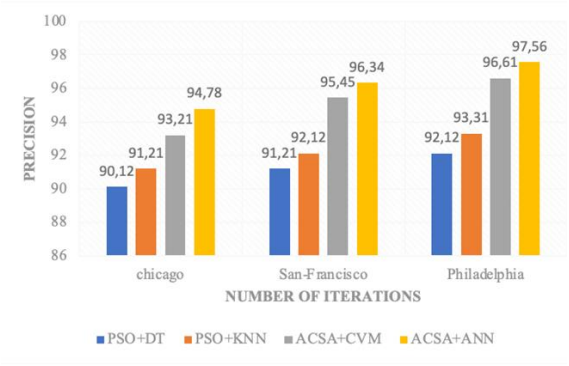

Figure 5. Comparison of Precision with Different Methods and datasets

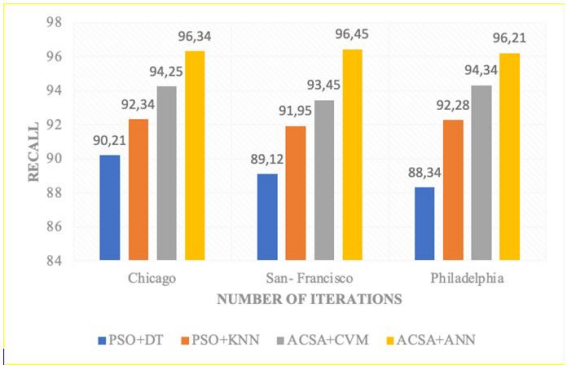

Figure 6. Comparison of Recall with different Methods and datasets

The results of the proposed method (ACSA+ANN) were compared with the F1-Measure Score results of other known other methods are illustrated in figure 7. The proposed ACSA+ANN gives higher F1-Measure Score results of 96.34%, whereas other methods such as PSO+DT, PSO+KNN and ACSA+CVM also gives higher f1-Measure Score for proposed ACSA+ANN based different methods such as 94.23%,93.56%,94.67% and respectively for San-Francisco, Chicago, and Philadelphia dataset. The proposed algorithm gives highest results than the existing methods respectively.

The results of the proposed method (ACSA+ANN) were compared with the Accuracy results of other known other methods are illustrated in figure 8. The proposed ACSA+ANN gives higher Accuracy results of 93.56%, whereas other methods such as PSO+DT, PSO+KNN and ACSA+CVM also gives higher accuracy for proposed ACSA+ANN based different methods such as 88.30%,90.01%,91.78% and respectively for San-Francisco, Chicago, and Philadelphia dataset. The proposed algorithm gives highest results than the existing methods respectively.

 ISSN 2723-6471

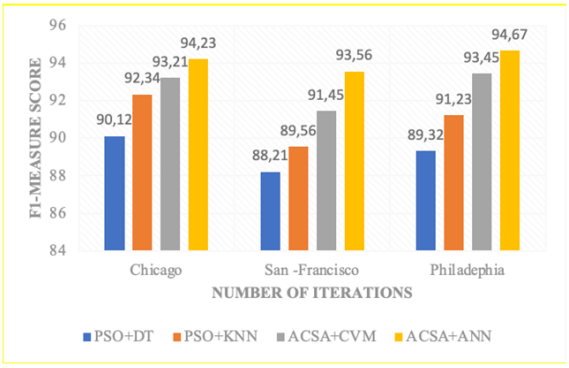

Figure 7. Comparison of F1-Measure Score with Different Methods and Datasets

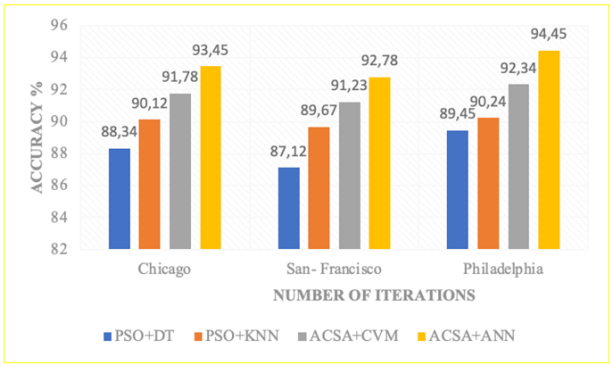

Figure 8. Comparison of Accuracy with Different Methods and Datasets

5. Conclusion

This paper discussed a new algorithm ACSA-ANN is modelled with the primary aim to improve security in accuracy of Anomaly Detections and secure cloud networks. The searching behavior of cuckoos for depositing eggs in the host nest is simulated and used to solve the IDS problem. The ACSO seeks the best current answer through iterations. Many algorithm settings of the cuckoo search algorithm were investigated to obtain the optimum response. The ANNs were trained by providing an optimum path (identified based on a specified fitness function) acquired after using the CS algorithm. The simulation results demonstrated that the proposed IDS performed significantly better than the previous technique, with considerable improvements in Accuracies, Precisions, Recalls, and F1- scores. Hybrid DL models can be used in the future to improve detection efficiency in cyber security.

6. Declarations

6.1. Author Contributions

Conceptualization: J.J., P.M., J.S., dan M.C.; Methodology: J.S.; Software: J.J.; Validation: J.J., J.S., dan M.C.; Formal Analysis: J.J., J.S., dan M.C.; Investigation: J.J.; Resources: J.S.; Data Curation: J.S.; Writing Original Draft Preparation: J.J., J.S., dan M.C.; Writing Review and Editing: J.S., J.J., dan M.C.; Visualization: J.J.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. H. Hussein, "Internet of things (IOT): Research challenges and future applications," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, pp. 77-82, 2019.

[2] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-inspired Information and Communications Technologies (BIONETICS)*, vol. 9, no. 1, pp. 21-26, 2019.

[3] Y. Wei, H. B. Hashim, S. H. Lai, K. L. Chong, Y. F. Huang, A. N. Ahmed, "Comparative Analysis of Artificial Intelligence Methods for Streamflow Forecasting," *in IEEE Access*, vol. 12, no. 1, pp. 10865-10885, 2024, doi: 10.1109/ACCESS.2024.3351754.

[4] S. Aftergood, "Cybersecurity: the cold war online," *Nature*, vol. 547, no. 7661, pp. 30-42, 2017.

[5] M. Wazid, A. K. Das, V. Chamola, and Y. Park, "Uniting cyber security and machine learning: Advantages, challenges and future research," *ICT Express*, vol. 8, no. 3, pp. 313-321, 2022.

[6] Z. Zhang, H. Ning, F. Shi, F. Farha, Y. Xu, J. Xu, F. Zhang, and K. K. R. Choo, "Artificial intelligence in cyber security: research advances, challenges, and opportunities," *Artificial Intelligence Review*, vol. 2022, no. 10, pp. 25, 2022.

[7] K. V. V. N. L. S. Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, "Building an intrusion detection system for IoT environment using machine learning techniques," *Procedia Computer Science*, vol. 171, no. 1, pp. 2372–2379, 2020.

[8] M. Praneesh and R. A. Saravanan, "Deep stack neural networks-based learning model for fault detection and classification in sensor data," in *Deep Learning and Edge Computing Solutions for High Performance Computing*, vol. 2021, no. 101, pp. 10, 2021.

[9] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained cuckoo search optimization-based protocol for routing in cloud network," *International Journal of Computer Networks and Applications*, vol. 8, no. 6, pp. 795-803, 2021.

[10] Y. Lu, M. Liu, J. Zhou, and Z. Li, "Intrusion detection method based on adaptive clonal genetic algorithm and backpropagation neural network," *Security and Communication Networks*, vol. 2021, no. 12, pp. 12-24, 2021.

[11] R. O. Ogundokun, J. B. Awotunde, P. Sadiku, E. A. Adeniyi, M. Abiodun, and O. I. Dauda, "An enhanced intrusion detection system using particle swarm optimization feature extraction technique," *Procedia Computer Science*, vol. 193, no. 1, pp. 504-512, 2021.

[12] S. Vanitha and P. Balasubramanie, "Improved ant colony optimization and machine learning based ensemble intrusion detection model," *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 849-864, 2023.

[13] M. A. Rahman, Y. Al-Saggaf, and T. Zia, "A data mining framework to predict cyber-attack for cyber security," in *2020 15th IEEE Conf. Industrial Electronics and Applications (ICIEA)*, vol. 15, no. 1, pp. 207-212, 2020.

[14] R. Das and T. H. Morris, "Machine learning and cyber security," in *2017 International Conference on Computer, Electrical and Communication Engineering (ICCECE)*, vol. 2017, no. 1, pp. 1-7, 2017.

[15] I. H. Sarker, "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects," *Annals of Data Science*, vol. 2022, no. 10, pp. 26, 2022.

[16] F. A. O. Sari, A. A. H. Alrammahi, A. S. Hameed, H. M. B. Alrikabi, A. A. Abdul-Razaq, H. K. Nasser, and M. F. AL-Rifaie, "Networks cyber security model by using machine learning techniques," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 257-263, 2022.

[17] B. G. H. B, P. Poornachandran, and S. K. P., "Deep-Net: Deep neural network for cyber security use cases," *arXiv preprint*, arXiv:1812.03519, vol. 2018, no. 12, pp. 35, 2018.

[18] M. K. Ahsan, *Increasing the predictive potential of machine learning models for enhancing cybersecurity*, Doctoral dissertation, North Dakota State University, vol. 2021, no. 8, pp. 124, 2021.

[19] O. Ben Fredj, A. Mihoub, M. Krichen, O. Cheikhrouhou, and A. Derhab, "CyberSecurity attack prediction: a deep learning approach," in *13th International Conference on Security of Information and Networks*, vol. 2020, no. 11, pp. 6-13, 2020.

[20] A. Kumar, R. S. Umurzoqovich, N. D. Duong, P. Kanani, A. Kuppusamy, M. Praneesh, et al., "An intrusion identification and prevention for cloud computing: From the perspective of deep learning," *Optik*, vol. 270, no. Nov., pp. 1-12, 2022.

[21] E. Kadena and M. Gupi, "Human Factors in Cybersecurity: Risks and Impacts," *Security Science Journal*, vol. 2, no. 2, pp. 51-64, 2021.

[22] A. Roy, D. S. Kim, and K. S. Trivedi, "Cyber security analysis using attack countermeasure trees," in *Proc. 6th Annu. Workshop Cyber Security Information Intelligence Research*, vol. 2010, no. 4, pp. 4, 2010.

[23] G. Kavallieratos, G. Spathoulas, and S. Katsikas, "Cyber risk propagation and optimal selection of cybersecurity controls for complex cyberphysical systems," *Sensors*, vol. 21, no. 5, pp. 1-21, 2021.

 ISSN 2723-6471

[24] R. A. Putawa and D. Sugianto, “Exploring User Experience and Immersion Levels in Virtual Reality: A Comprehensive Analysis of Factors and Trends,” *Int. J. Res. Metav.*, vol. 1, no. 1, pp. 20-39, 2024

[25] K. Jawad, R. Mahto, A. Das, S. U. Ahmed, R. M. Aziz, and P. Kumar, "Novel cuckoo search-based metaheuristic approach for deep learning prediction of depression," *Applied Sciences*, vol. 13, no. 9, pp. 1-27, 2023.

[26] S. R. Pokuri and N. Devarakonda, "Hybridization of adaptive cuckoo's search algorithm with core vector machine for feature selection," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 4, pp. 4740-4748, 2022.

[27] Y. Xu, X. Zhang, C. Lu, Z. Qiu, C. Bi, Y. Lai, J. Qiu, and H. Zhang, "Network threat detection based on group CNN for privacy protection," *Wireless Communications and Mobile Computing*, vol. 2021, no. 18, pp. 18, 2021.

[28] R. Zarai, "Recurrent neural networks and deep neural networks based on intrusion detection system," *Open Access Library Journal*, vol. 7, no. 3, pp. 1-11, 2020.

[29] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, no. 1, pp. 35365-35381, 2018.

[30] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, 2017.

[31] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *Journal of Big Data*, vol. 8, no. 1, pp. 65, 2021.

[32] Henderi and Q. Siddique, “Anomaly Detection in Blockchain Transactions within the Metaverse Using Anomaly Detection Techniques”, *J. Curr. Res. Blockchain.*, vol. 1, no. 2, pp. 155–165, Sep. 2024.

[33] D. Napoleon, M. Praneesh, S. Sathya, and M. SivaSubramani, "An efficient modified fuzzy possibilistic c-means algorithm for segmenting color-based hyperspectral images," in *IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM-2012)*, Nagapattinam, India, vol. 2012, no. 3, pp. 5, 2012.

[34] R. O. Ogundokun, J. B. Awotunde, P. Sadiku, E. A. Adeniyi, M. Abiodun, and O. I. Dauda, "An enhanced intrusion detection system using particle swarm optimization feature extraction technique," *Procedia Computer Science*, vol. 193, no. 1, pp. 504-512, 2021.

[35] S. Budilaksono, A. A. Riyadi, L. Azhari, D. D. Saputra, M. A. Suwarno, I. G. A. Suwartane, A. Ramadhan, A. P. Utomo, and A. Fauzi, "Comparison of data mining algorithm: PSO-KNN, PSO-RF, and PSO-DT to measure attack detection accuracy levels on intrusion detection system," in *Journal of Physics: Conference Series*, vol. 1471, no. 1, pp. 8-15, 2020.

[36] A. S. Paramita and T. Hariguna, “Comparison of K-Means and DBSCAN Algorithms for Customer Segmentation in E-commerce,” *J. Digit. Mark. Digit. Curr.*, vol. 1, no. 1, pp. 43-62, 2024.