Enhancing the Accuracy of Intrusion Detection Systems by Reducing the Rates of False Negatives Through Using XG-boost and Optimization Algorithm

Noor Saud Abd [1] [0009-0007-8363-6176], Kamel Karoui [2] [0000-0003-2507-9571] and Mohammed Ghassan Abdulkareem[3] [0009-0007-9330-6285]

[1] PhD student at National School of Engineers of Tunis ENIT – Tunisia, and Assistant Lecturer in the Department of Cyber Security, Computer Science and Mathematics College, Tikrit University – Iraq

[2] Engineer Doctor of Philosophy (Ph.D.) Computer Science Professor at National Institute of Applied Sciences and Technology – Tunisia

[3] Department of Petroleum Project Management, College of Industrial Management of oil and Gas Basrah University for Oil and Gas

[1] noor.s.abd@tu.edu.iq
[2] kamel.karoui@insat.rnu.tn
[3] mohammed.alsultan@buog.edu.iq

Abstract. Intrusion Detection Systems (IDSs) are regarded as critical for network security, with malicious activities detected and mitigated through their implementation. However, a key challenge in IDS deployment is the high rate of false negatives, where attacks remain undetected, posing significant security risks. In this study, an enhanced IDS model is proposed, integrating XG-boost, a robust gradient boosting algorithm, with Cat Swarm Optimization (CSO) to reduce false negatives and improve detection accuracy. The scalability and performance of XG-boost are utilized to handle complex network traffic data, while CSO is applied to optimize XG-boost's hyperparameters by mimicking natural cat behaviors, ensuring optimal model performance. The proposed approach was evaluated using a benchmark dataset, and a notable reduction in false negatives was demonstrated compared to traditional IDS methods. Improved detection accuracy across various types of cyberattacks was also observed, while maintaining a low false positive rate, which is crucial for minimizing disruptions to regular network operations. The optimized XG-boost model achieved an accuracy of 98%, with a precision of 97.8% and an F1-score of 97.7%, significantly outperforming the non-optimized model (accuracy: 84.1%, precision: 86.5%, F1-score: 84.1%). These results underscore the effectiveness of the proposed method in real-world IDS deployment, where both security and operational efficiency are essential.

Keywords: XG-boost algorithms, Cat-boost algorithms, Intrusion detection, Cyber Security.

This provisional PDF is the accepted version. The article should be cited as: Security and Safety doi: 10.1051/sands/2024025

1 Introduction

Large volumes of data have become accessible in many formats within our society due to emerging technologies, such as cloud computing, social media, and big data analytics. Many security risks are introduced when such data is transmitted over a network or the internet. Although some cutting-edge intrusion prevention methods have been developed to mitigate these risks, attacks continue to occur and grow in scale. Therefore, a dependable method for detecting unauthorized traffic capable of compromising the network is necessary [1]. More complex attacks than ever before are currently challenging network security due to the rapid development of big data, the Internet, and AI. As a result, there is a growing demand for more powerful and efficient network intrusion detection systems (NIDS). Related techniques are applied by NIDS for the collection, cleaning, modeling, and identification of various network behaviors [2].

As technology and the Internet continue to advance, cyber-attacks are becoming more frequent, making cybersecurity an essential component. Numerous types of attacks, such as denial of service (DoS), DNS spoofing, U2R (user-to-root), R2L (root-to-local), probe attacks, and others, can occur on the Internet. Machine learning (ML), a rapidly advancing field, encompasses a variety of techniques and surpasses various legacy algorithms. Cybersecurity professionals could apply such techniques to detect breaches [3, 4]. Artificial intelligence (AI) has emerged as one of the most cost-effective and efficient approaches for constructing network intrusion detection systems (NIDS), as evidenced by its growing popularity in recent years. A variety of machine learning (ML) and deep learning (DL) techniques are employed to develop models capable of distinguishing between normal data packets and intrusions, utilizing diverse methodologies to achieve this goal. In this study, ML is applied to extract meaningful information from the dataset and to develop the NIDS model. Typically, several preprocessing techniques are employed in ML to select and clean the dataset prior to analysis. During this phase, missing values are addressed, and any dataset errors are corrected, ensuring an error-free dataset. The dataset is then divided into two subsets: one for training and one for testing, with 80% typically allocated for training and 20% for testing [5]. After the training and testing phases are completed, the model's efficiency can be evaluated. Various ML algorithms, such as decision trees, K-nearest neighbors, support vector machines (SVM), and K-means clustering, are commonly employed to extract meaningful data from both raw and modified datasets. These algorithms have consistently demonstrated high efficiency in producing the desired outcomes.

The primary function of an intrusion detection system (IDS) is to monitor network traffic to identify unusual or suspicious activity and to implement preventive measures against potential intrusion threats. IDS systems are generally categorized into two types: network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). NIDS is typically deployed at key network points to monitor traffic more vulnerable to attack, while HIDS operates on specific devices connected to the network. Intrusion detection strategies are further divided into two main approaches: signature-based IDS and anomaly-based IDS [1, 6, 7]. Signature-based IDS is focused on identifying specific patterns, or "signatures," of known intrusions, with regular updates to address emerging threats, including zero-day attacks. Anomaly-based IDS, also referred to as behavior-based detection, continuously monitors system activity and compares known behavioral patterns with new behaviors to identify potential threats. When an IDS alerts the system administrator, Intrusion Prevention Systems (IPS) can be applied to mitigate threats such as Trojan horses and distributed denial-of-service (DDoS) attacks [2].

The development of an effective NIDS remains one of the most significant challenges in network security. Despite advances in NIDS technology, many current systems predominantly rely on signature-based detection methods rather than anomaly-based protocols [3, 8]. Several factors contribute to the limited adoption of anomaly detection systems, including the dynamic nature of system behavior, the reliability and accuracy of training data, the collection of high-quality data, and the associated costs and error rates. These challenges can result in inaccurate and ineffective NIDS validation and solutions. Thus, the development of a more efficient intrusion detection system is essential to address the limitations of current network security practices.

This research is aimed at minimizing false negatives through the development of an intrusion detection model utilizing multiple ML techniques on selected features acquired during the modeling stage. Since the false negative (FN) and false positive (FP) rates are primary concerns in widely used procedures, efforts are being made to address these issues. FPs occur when the system incorrectly classifies normal activity as harmful, while FNs occur when malicious behavior is not classified as such. ML algorithms provide a type of solution by searching across various network activity datasets for suitable hyperparameters for classifiers to improve detection performance. Gradient boosting algorithms, such as XG-boost, have gained popularity due to their ability to handle missing values in datasets, incorporate regularization techniques, optimize parallel and distributed computing, implement optimized tree pruning, and offer a customizable objective function, all of which allow them to outperform other state-of-the-art and baseline ML algorithms [11].

The intrusion detection framework consists of four modules: data collection, preprocessing, sampling, and production. Data collection is crucial during the intrusion detection process, as original security data derived from multiple sources is heterogeneous and redundant, among other factors [5]. Direct data analysis can have a more detrimental impact, making the preparation of raw data essential. After preprocessing, the layer with the least data is doubled in size, while the layer with the most data is reduced. Finally, the intrusion detection model will be trained and tested using the (XG-boost) approach. The results and confusion matrix were then further enhanced using cat boost optimization [12].

Section 2 presents related work, Section 3 describes the XG-boost algorithms, and Section 4 covers Cat Swarm Optimization, while Section 5 provides information on the data used. The research methodology is reviewed in Section 6, along with a section on the results. Lastly, the investigation is concluded in Section 7. The motivation behind this study lies in the growing complexity and frequency of cyber-attacks in today's interconnected world, driven by the rapid expansion of technologies such as big data, AI, and the internet.

Despite advancements in network intrusion detection systems (NIDS), significant challenges persist, particularly with false negatives and false positives that limit the accuracy of detecting malicious activity. Current systems, which often rely heavily on signature-based detection, struggle to keep up with the evolving tactics of cyber attackers.

This study aims to address these gaps by applying machine learning (ML) techniques, particularly XG-boost and Cat Boost optimization, to develop a more effective NIDS model.

Through optimizing data preprocessing and classifier hyperparameters, the research seeks to enhance detection accuracy, minimize false positives and negatives, and ultimately strengthen network security. This study advances cybersecurity by developing an efficient network intrusion detection system (NIDS) that leverages advanced machine learning techniques, specifically XG-boost and Cat Boost optimization, to improve detection accuracy. Key contributions include reducing false positives and false negatives, a common limitation in existing systems, through precise hyperparameter tuning and regularization, which enhances the model's ability to distinguish between normal and malicious traffic. The study introduces a multi-stage data processing pipeline—encompassing data collection, preprocessing, and sampling—that improves data quality by handling missing values and balancing classes for more accurate training. Additionally, the proposed NIDS employs a modular, adaptable framework that can be applied across various network environments, making it a scalable solution for detecting complex and evolving cyber threats. Comprehensive evaluation confirms its effectiveness, positioning this model as a robust solution to real-world cybersecurity challenges.

2 Related Work

Numerous academics have attempted to address such performance concerns by utilizing IDS datasets and other ML techniques. In this study, we discussed some relevant research that merged different forms of optimization algorithms with the XG-boost algorithm for identifying attacks and producing good percentages as a consequence. We also looked into the false negative rates in some works, but the rates were slightly high and must be addressed.

- Song, Y., Li, H., Xu, P., (2018). The intrusion detection model presented in this research is majorly based upon the XG-boost (Extreme Gradient Boosting), and it selects the necessary parameters using the Whale Optimization Algorithm (WOA). Before being inserted into the WOA-XG-boost algorithm, collected network data is preprocessed by using the PCA (i.e., Principal Component Analysis) dimensionality reduction technique, which improves data's intrusion detection capabilities during training. The well-known KDD CUP99 data set from the field of computer networks is used to test the experimental findings. Additionally, the model of intrusion detection employed in this approach is more accurate in the case when put to comparison with the accuracy of discoveries made by adjusting the parameters in the prior method. WOA-XG-boost model is the most successful, according to the results. The method outperforms earlier algorithms in terms of sensitivity, specificity, and accuracy. With a sensitivity of 0.9958 and a specificity of 0.9574, the average ACC is 0.9906. The findings using Grid Search SVM were the most dismal. It demonstrates that for parameter optimization, the WOA approach performs better than the Grid Search methodology [5].

- Bhattacharya, S., S, S. R. K., Maddikunta, (2020) [13]. This paper presents a hybrid firefly ML model for PCA that is intended to categorize IDS datasets. The research's dataset was obtained from Kaggle. The model starts by applying one-hot encoding to alter IDS datasets. Dimensionality is subsequently decreased by using a hybrid PCA firefly approach. XG-boost algorithm is used to classify the reduced dataset. With the use of state-of-the-art ML approaches, we thoroughly evaluate the model to show the

superiority of our suggested method. According to experimental results, the performance of the suggested model has been better than state-of-the-art ML models. They found that the specificity regarding XG-boost PCA, which had a slightly lower specificity score of 98.2%, was equivalent to that of plain XG-boost (99.9% specificity) and XG-boost PCA firefly (99.9% specificity). Regarding accuracy measurements, the results obtained from the three approaches are nearly identical. Nonetheless, the XG-boost-PCA-firefly method performs significantly better in terms of sensitivity than the other two algorithms (XG-boost 92.3%, XG-boost PCA 91.8%) [13].

- Zivkovic, M., Tair, M., (2022). This paper presents an enhanced version of the well-known Firefly method together with an application for modifying and optimizing the hyperparameters regarding the XG-boost classifier for network intrusion detection. The comparatively high false-positive and false-negative rates of NIDSs are one of the biggest causes for concern. The proposed study uses an improved Firefly approach to optimize the XG-boost classifier, resolving this issue. Before being contrasted with the original firefly method and other state-of-the-art metaheuristics, the suggested enhanced firefly algorithm was verified on 28 well-known CEC2013 benchmark examples utilizing methods from the current literature. After that, the proposed method was applied and assessed for XG-boost hyper-parameter optimization. The modified classifier was tested for network intrusion detection using the more recent USNW-NB15 dataset and the commonly utilized benchmarking dataset NSL-KDD. Based on experimental results, the suggested meta-heuristics have a great deal of promise for improving the accuracy of the classification as well as the average precision of the NIDSs, in addition to solving the ML hyperparameter optimization challenge [10].

- Liu, Z., Wang, Y., (2023). Have proposed an approach for the identification of the DDoS attacks in the SDNs combining feature engineering with Machine Learning. After CSE-CIC-IDS 2018 data-set cleaning and normalization, an enhanced binary grey wolf optimization algorithm was utilized to identify the best feature subset. The optimal DDoS attack detection classifier was selected after that and installed in the SDN controller after the ideal feature sub-set had been trained and then assessed in ML algorithms Random Forest (RF), Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (k-NN), and XG-boost. According to the results, RF outperforms other models in several performance parameters, including precision, accuracy, F1, recall, and AUC values. We also investigate how various models and algorithms compare to one another. The findings demonstrate that while the F1-score and precision of RF (96), SVM (98), XG-boost (96), KNN (96), and decision tree (95) vary, there is a FN ratio of roughly 5% from dataset samples [1].

- Mhawi, D. N., Aldallal, (2022). We suggested a unique IDS method for handling imbalanced and high-dimensional traffic with minimal DR, using hybrid methods depending on the desired FS. To achieve the optimal subset concerning function correlation, a hybrid CFS FPA strategy utilizing a 30-feature sample and a hybrid ensemble learning technique has been suggested. The suggested system was able to manipulate and process conflicts that the previous work suffered from, like the (FNR, FAR, accuracy, and DR), by eliminating non-essential features as well as selecting just affected features using the suggested approach CFS_FPA through the combination of the

correlation feature selection and forest penalized attribute. As a result, accuracy in the testing phase increased to 87% and FNR was 0.123. The proposed model's final experimental results, with the use of the CICIDS-2017 data set, demonstrated 99.73% accuracy, 99.82% precision, 99.71% F-measure, 99.8% DR, and 0.004 FAR. Moreover, the suggested approach outperforms the previously suggested CFS–FPA–ensemble approach and the existing classification algorithms. This approach has the potential to provide a significant competitive edge in IDS business when compared with the alternative methods. As a result, offers greater robustness and high reliability while spotting intrusions and categorizing benign traffic.

- Cheng, P., Xu, K., (2022) [14]. Using a wrapper strategy that is based on a genetic algorithm for feature selection in network intrusion detection systems, Jaw and Wang offered an all-encompassing approach to intrusion detection systems. This approach allows for the selection of features in intrusion detection systems. Additionally, the use of logistic regression was utilized to accomplish ensemble learning. Their experimental findings on the CICIDS2017, NSL-KDD, and UNSW-NB15 datasets exhibited remarkable accuracy rates of 98.99%, 98.73%, and 97.997%, respectively. Additionally, their detection rates were 98.75%, 96.64%, and 98.93%, respectively. These results were derived from experiments in which just 11, 8, and 13 key characteristics were chosen from the datasets to arrive at the conclusions.

- Gupta, N., Jindal, V., (2022) [15]. To address the issue of class imbalance, Gupta et al. [15] suggested that network-based intrusion detection systems benefit from the implementation of ensemble approaches. Their method consisted of three stages: first, a deep neural network was utilized to differentiate between normal and suspect network traffic; second, eXtreme Gradient Boosting was utilized to identify large-scale attacks; and third, Random Forest was utilized to categorize less severe attacks. Although the time complexity was evaluated in hours rather than minutes, this model was able to attain accuracy rates of 99%, 96%, and 92%, respectively, when it was applied to the NSL-KDD, CIDDS-001, and CICIDS2017 datasets.

- Two phases of ensemble learning classifiers were incorporated into the hybrid feature selection technique that was implemented by Tama et al.[16]. Their system was evaluated using the CIC-IDS2017 dataset, which had 37 features, and it obtained an accuracy of 96.46% during the testing process.

Alkasassbeh and Almseidin [20] conducted a comprehensive evaluation using three distinct machine learning algorithms namely, J48, MLP, and Bayes Network classifiers—on the widely studied KDD dataset [21] with the objective of detecting and classifying various forms of network attacks, including DoS, R2L, U2R, and Probe. Among these methods, the J48 classifier was observed to yield the highest accuracy, demonstrating its effectiveness in this context.

In a related study [22], Farnaaz and Jabbar developed an intrusion detection system leveraging a random forest model augmented by a feature subset selection technique known as symmetrical uncertainty. Their experiments, carried out on the NSL-KDD dataset [23], revealed that the model performed effectively, achieving a low false alarm rate (FAR) along with a high recall rate, which highlighted the reliability of their approach for intrusion detection.

More recently, research has increasingly focused on the application of feature selection methods in combination with machine learning algorithms to improve intrusion detection systems. For instance, a study conducted by Meftah et al. [24] utilized Recursive Feature Elimination (RFE) and random forests for feature selection when working with the UNSW-NB15 dataset [25], followed by the application of various machine learning models such as Logistic Regression, Gradient Boost Machine, and Support Vector Machines. Similarly, several other researchers, including Elmasry et al. [26] have also explored the integration of feature selection techniques with machine learning algorithms for building network intrusion detection systems. Despite the significant findings of these studies, they often lacked the use of robust statistical methods for selecting the final models and encountered issues related to the time-intensive nature of their approaches.

To overcome the drawbacks identified in the existing body of work, employing statistical significance tests can be advantageous. Such tests enable researchers to compare the performance of various machine learning models more rigorously and quantify the probability that the observed performance scores are consistent with the assumption that they originate from the same distribution, all while considering the classification time associated with each model.

Additionally, there are other notable limitations in the related research. These include insufficient preprocessing of the NSL-KDD dataset, particularly for categorical features such as `protocol type` (comprising 3 types), `service` (with 70 types), `flag` (with 11 types), and `label` (with 23 types). Furthermore, many evaluation techniques employed in these studies do not effectively address the challenges posed by significant class imbalances within the datasets, which can compromise the robustness and reliability of the intrusion detection systems.

The study by Noor Saud Abd et al., addresses the significant challenge of detecting malicious attacks on peer-to-peer smart grid platforms, where attack methods evolve continuously. The research highlights the critical role of data science in cybersecurity, focusing on deep learning techniques, particularly Convolutional Neural Networks (CNNs). Although CNNs are primarily known for image processing due to their feature extraction and pattern recognition capabilities, the study demonstrates their effective application to non-image data, specifically text data related to cyber-attacks from the Kaggle repository. The research shows that using the CNN algorithm for this purpose enhances cybersecurity by enabling real-time analysis of large data volumes, facilitating the early detection of anomalies and potential threats. Machine learning's adaptability allows security systems to respond to new and emerging threats. The study achieved exceptional performance metrics, with accuracy, precision, and F1 scores of 0.999 and a sensitivity of 1.0. However, the primary goal was not just achieving high accuracy but minimizing false negatives and refining the confusion matrix to ensure that no malicious attack is misclassified as normal traffic. This approach contributes to more reliable and proactive cybersecurity measures.

The study by Nan Li focused on the importance of the Internet of Medical Things (IoMT) in enhancing precision health is underscored in this study, with improvements in the quality and timeliness of care and reductions in the complexities and costs associated with patient care, particularly during pandemic crises, being highlighted. Data security and prediction accuracy are identified as primary concerns in IoMT systems, necessitating that robust security strategies be developed and implemented. A

comprehensive review of IoMT systems is presented in the paper, with a focus placed on security and privacy challenges, an overview provided through a hierarchical system architecture framework, and an evaluation conducted on security metrics based on network service performance requirements. State-of-the-art security techniques are discussed in relation to potential risks, and security issues and solutions across the sensing, network, and cloud infrastructure layers of IoMT systems are analyzed to ensure alignment with system architecture and network service demands. Biometrics-based technologies are highlighted as particularly effective for authentication and key management in IoMT, with details and future improvement recommendations provided. Current challenges are identified, and future research directions are proposed, with emphasis placed on the potential for employing emerging technologies to further bolster IoMT security [39].

The above papers contributed to the writing of this paper, adding my practical experience and improvements through my use of hybridizing XG-boost with one of the optimization algorithms (cat swarm) to reduce false negatives to a minimum, as described in the results section.

3 XG-Boost Algorithms

An ML system called XGBoost makes use of gradient-boosted decision trees (GBDT). The finest ML software for classification, regression, and ranking problems is this one, which also incorporates parallel tree boosting [27]. Regarding prediction accuracy, gradient boosting is widely acknowledged to be superior to RF. An ensemble model provides strong prediction accuracy since it consists of several DTs. Since the DT forecasts the residual error regarding the preceding DT, it learns the model and prevents overfitting. Categorical boosting (CatBoost), extreme gradient boosting (XGBoost), and light gradient boosting machine (LightGBM) are the three methods [28]. ML packages LightGBM, XGBoost, and CatBoost have been based on the gradient boosting method. After being created in 2014, XGBoost has quickly become well-known as a result of its exceptional results on large data sets as well as multiple wins in the competitions of data science [27]. The main XGBoost objectives, incorporating the gradient-boosting DTs, are performance and speed. It represents a machine augmentation method example that has been developed via Tianqi Chen and has since then been embraced by several developers. Through the prediction of the preceding DT's residual error, DT avoids overfitting. CatBoost, XGBoost, and LightGBM are the three options [29]. Gradient boosting-based Machine Learning libraries include LightGBM, XGBoost, and CatBoost. Since it was created in 2014, XGBoost has been well-known because of its remarkable performance on large data sets and numerous victories in competitions of data science [27]. XGBoost uses gradient-boosting decision trees with the primary goal of increasing performance and speed. Numerous developers have now adopted Tianqi Chen's method of adding machine augmentation, or augmenting machines. XGBoost is built using algorithms based on trees. The tree techniques use the qualities of the dataset, also referred to as features or columns, and use such features as the internal or conditional nodes. The state at the root node determines whether the tree splits into edges or branches. Splitting is typically done to arrive at a choice.

The leaf node is the branch's terminal which does not produce any additional edges [30]. XGBoost model utilizes an additive training approach for optimizing objective functions; hence, the result of one optimization stage influences the subsequent step's optimization procedure. The objective function of the model is represented as [31]. Large datasets can be effectively trained using scalable ML models, which is why scalability is appropriate. XGBoost is highly flexible because it has a wide range of hyperparameters that may be changed to enhance performance. Missing Value Handling: Working with real-world data that commonly contains missing values is made easier by XGBoost's built-in support to handle missing values. Interpretability: XGBoost places a strong emphasis on feature importance, making it easier to understand which factors to take into account when making predictions, unlike some ML algorithms that could be challenging to understand. Unfortunately, XGBoost is too computationally intensive for systems with limited resources, especially when training large models. An objective function is made up of a regularization term and a loss function is optimized via XGBoost. [32] defines the objective function.

$$Obj(\Theta) = sum_{i=1}^{n}{l\ (y_i,{\ y^}_i)}\ k=1k\Omega(fk) \quad (1)$$

Θ denotes the model parameters for each tree in the ensemble.

n denotes the number of the training samples.

yi represents the true label of ith instance.

y^i denotes the predicted value for *ith* instance.

L(yi,yi^) represents the loss function, which measures the difference between the predicted value and the true label.

K denotes the number of trees in the ensemble.

Ω(fk) denotes the regularization term applied to each tree in the ensemble.

Loss Function: The choice regarding loss function is based on the type of problem being solved. Common loss functions include:

Regression: Mean Squared Error (MSE).

Binary Classification: Binary Logistic Loss (logistic regression).

Multiclass Classification: Softmax Cross-Entropy

Regularization Term: XG-boost supports both L1 (Lasso) and L2 (Ridge) regularization terms to prevent overfitting. The regularization term is applied to each tree and is defined as:

$$\Omega(fk)=\gamma T+21\lambda \| w \| 2 \quad (2)$$

T denotes the number of leaves in a tree *fk*.

4 Cat Swarm Optimization

Yahoo created CatBoost optimization (CSO) in the year 2017, and it's quite good at handling categorical variables. It is a better method that allows for quick learning on both GPU and CPU and automatically uses regularization to prevent overfitting [33]. Chu et al. (2006) and Tsai and Chu (2007) introduced CSO as a population-based and heuristic method for maintaining cats' natural behaviour. A cat's innate desire to hunt is powerful. To find the optimal solution to a complex optimization problem, two submodes are being mathematically modelled [34]. In the case when they are resting, cats move very slowly and are constantly alert. We call this type of behaviour "seeking

mode." Cats are quite aggressive and swift when they detect the presence of a bird. The term "tracing mode" refers to this mode [1]. There are 2 operation modes for the CSO algorithm, which are: seeking and tracing. A solution set has been represented by every cat, which has its own fitness value, position, and flag. The position in search space consists of M dimensions, every one of which has a different velocity; the flag indicates whether cats are seeking or tracing; the fitness value indicates how well the solution set (i.e., the cat) performs. Therefore, before we can run cats through the algorithm, we must first decide the number of cats that should be included in the iteration. The best cat from every one of the iterations is retained in memory, and the optimal solution is represented by the cat from the last iteration [16]. The six steps that make up the CSO process are as follows [35] [36]:
(1) Set lower and upper boundaries for solution sets.
(2) N cats (solution sets) should be randomly generated and distributed over an M-dimensional space, every one of which has a random velocity value that should not exceed the maximum velocity value that has been defined.
(3) Sort the cats into tracing and seeking modes at random based on MR. A mixture ratio, or MR, is selected within the range [0, 1]. For instance, if N = 10 and MR = 0.2, then a random selection of 8 cats will enter the seeking mode, while the remaining 2 cats will enter the tracing mode.
(4) All cats' fitness values can be determined by using a domain-specific fitness function. After that, the best cat is selected and retained in memory.
(5) After that, cats switch to tracing or seeking mode.
(6) Divide the cats into seeking and tracing groups at random for the following iteration based on MR.
(7) Check to see if the condition of termination is met; if not, restart Steps 4–6 and end the program.

4.1 Seeking Mode. Four key parameters—counts of dimension to change (CDC), seeking memory pool (SMP), seeking a range of selected dimensions (SRD), and self-position considering (SPC)—have significant impacts on the seeking mode, which mimics the resting behavior of the cats. The user uses trial and error to fine-tune and define each of these parameters. The seeking memory size for cats is determined by SMP, which also defines the number of candidate positions that every cat will choose to go to. For instance, if SMP is set to 5, after that each cat will generate five new random positions, one of which will be selected to be the cat's next position. The other 2 parameters, which are SRD and CDC, will determine the way to randomize the new placements. The number of dimensions that should be changed is specified by CDC and falls between 0 and 1. For instance, if the CDC is set to the value of 0.20 and the search space consists of 5 dimensions, after that for every cat, four of the dimensions must be changed at random while the fifth dimension remains unchanged. SRD stands for mutative ratio for the chosen dimensions; that is, it indicates the degree of modification and mutation for the dimensions that the CDC chose. Last but not least, SPC is a Boolean value indicating whether or not a cat's present location will be selected as a candidate position for the following iteration. Because the present position is regarded as one of them, for instance, in the case where the SPC flag is set to true, we must produce the (SMP1) number of candidates for each cat rather than the SMP number. The steps for seeking mode are as follows:

(1) Making as many as SMP copies of Catk's current position.

(2) For each one of the copies, as many CDC dimensions are selected randomly to be mutated. In addition to that, subtract or add SRD values randomly from current values, which replace old positions as can be seen in the following equation:

$$Xjdnew\ (1 + rand * SRD) * Xjdold \quad (3)$$

In which Xjdold denotes the current position; Xjdnew denotes the next position; j represents the number of a cat d represents dimensions; and rand represents a random number in the [0, 1] interval.

(3) Evaluation of fitness value (FS) for all candidate positions.

(4) Based upon the probability, perform the selection of a candidate point to be the next position for the cat where the candidate points that have a higher value of the FS have a higher chance to be chosen as can be seen from eq. (3). Nonetheless, if all of the fitness values are equal, then all selecting probability of every one of the candidate points needs to be set to 1.

$$Pi = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}, \text{ where } 0 < i < j \quad (4)$$

If minimization is the objective, then $FSb = FSmax;$ else, $FSb = FSmin$.

4.2. Tracing Mode. This mode mimics how cats trace their surroundings. In the initial iteration, all of the cat's position dimensions are assigned random velocity values. Yet, the values of velocity need to be changed for the following steps. The following are cats that move in this mode:

(1) Update velocity values (Vk,d) for all dimensions based on eq. (5).

(2) If a value of velocity outranged the maximum value, then it equals the maximum velocity value.

$$Vk,d = Vk,d + r1c1\ (Xbest,d - Xk,d) \quad (5)$$

(5) Update Catk position based on the following equation:

$$Xk,d = Vk,d + Xk,d \quad (6)$$

The figure (1) shows the structure of the cat swarm optimization algorithm.

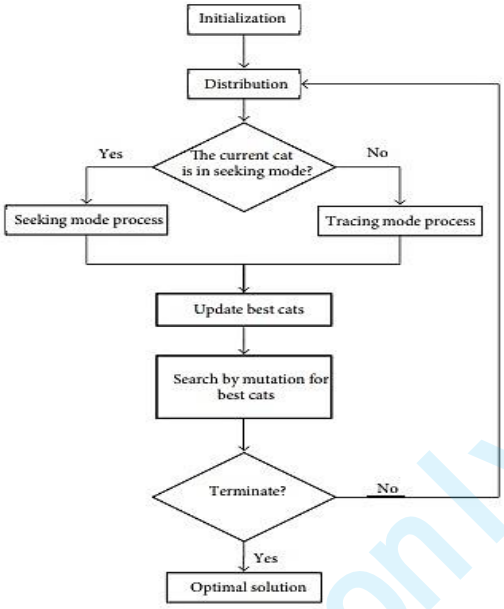

Figure (1) Cat swarm optimization algorithm general structure.

The feature engineering method in this study is enhanced by utilizing XG-boost's capabilities to optimize data processing and model performance. Key techniques are included, such as handling missing values directly within the model, ensuring data integrity without extensive preprocessing, and focusing on feature importance for selecting impactful variables, which improves interpretability and accuracy. Performance is optimized, and overfitting is controlled through hyperparameter tuning, including adjustments to learning rates, maximum depth, and regularization terms (L1 and L2). Regularization techniques are applied to each tree in the ensemble, allowing a balance between complexity and generalizability. The objective function is optimized based on the problem type, with specific loss functions like Binary Logistic Loss being used for classification. Additionally, an iterative learning process is enabled by XG-boost's tree-based structure and additive training approach, where each tree is adjusted based on previous residual errors, refining accuracy at each step. Through these methods, a scalable and interpretable solution ideal for constructing an effective network intrusion detection system is provided.

Advantages such as efficient handling of categorical variables and prevention of overfitting are offered by Cat-boost optimization (CSO). However, several limitations are associated with it. First, parameters that require careful tuning (e.g., counts of dimension to change, seeking memory pool (SMP), and seeking a range of selected dimensions are relied upon by the CSO algorithm, making the process time-consuming and computationally expensive, particularly in high-dimensional datasets. Additionally, exploration of the solution space may be limited by the division of cats into seeking and tracing modes in CSO. The stochastic nature of CSO can also lead to variable results across runs, impacting consistency and model reliability. Moreover, the computational

intensity of updating each cat's velocity and position at every iteration, particularly for large datasets and complex optimization tasks, can be considerable. Lastly, although strengths are present, substantial resources and careful parameter management are required by CSO, which can restrict its scalability and efficiency in certain applications. Finally, the basic steps of the model can be explained in general in Figure (2).

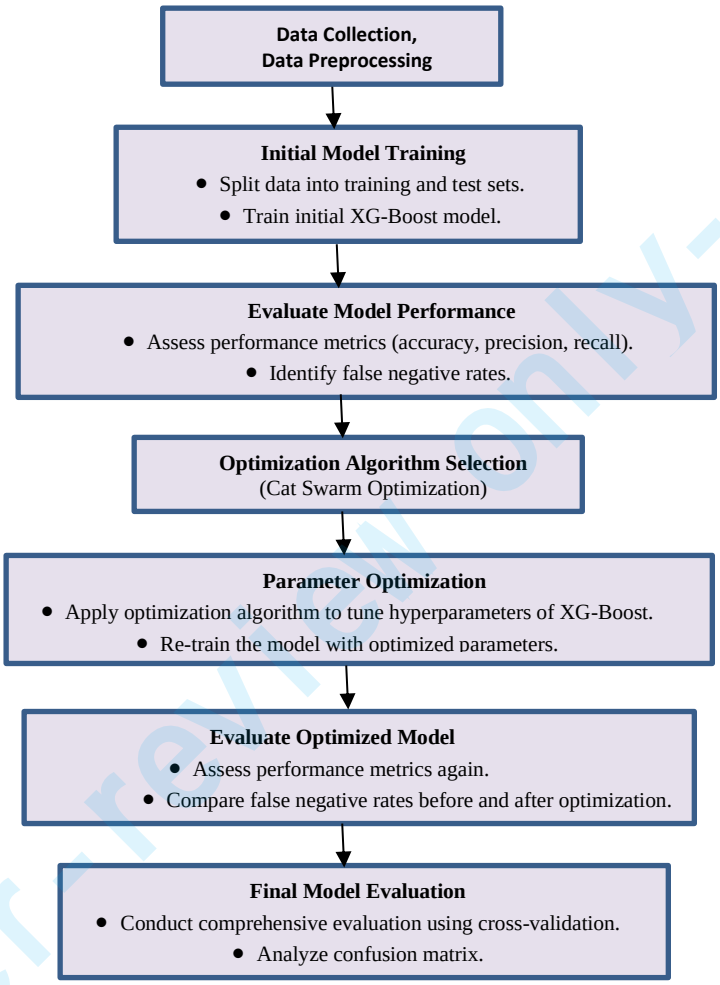

Figure (2) Model of the XG-boost and Optimization Algorithm.

5 Dataset

The present data contains different kinds of IoT intrusions. The categories of the IoT intrusions enlisted in the data are (DDoS, Spoofing, Brute Force, Web-based, Mirai, and DoS, Recon). The dataset contains 1191264 network intrusion incidents, each with 47 characteristics. The dataset can be used to develop a predictive model capable of