FSHAHA: FEATURE SELECTION USING HYBRID ANT HARRIS ALGORITHM FOR IOT NETWORK SECURITY ENHANCEMENT

Priyanka and Anoop Kumar

Dept. of Computer Science, Banasthali Vidyapith, India

ABSTRACT

Enhancing machine learning model performance involves selecting relevant features, particularly in high-dimensional datasets. This paper proposes a hybrid method named the Multi-Objective Ant Chase algorithm, which integrates Ant Colony Optimization (ACO) and Harris Hawk Optimization (HHO) for effective feature selection. ACO excels at exploring large search spaces using pheromone-guided navigation, while HHO focuses on targeted search with adaptive hunting tactics. Conventional algorithms, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Monarch Butterfly Optimization (MBO), often face premature convergence in high-dimensional, sparse datasets, becoming stuck in local optima. Unlike these, the ACO and HHO combination balances exploration and exploitation efficiently. ACO's broad search capability complements HHO's fast convergence, providing robust global optimization. Experimental results indicate that the Multi-Objective Ant Chase algorithm outperforms individual ACO, HHO, and other comparative algorithms across metrics like Accuracy, Sensitivity, Specificity, False Alarm Rate, and Detection Rate.

KEYWORDS

Internet of things, Feature Selection, Ant-colony algorithm, Harris hawk algorithm, Network security

1. INTRODUCTION

Feature selection is the crucial process of machine learning that helps to choose a subset of the original feature in order to reduce the dimensionality of data with retaining the most significant amount of information.[1] There are two reasons why FS is important; firstly, it rejects noise and irrelevant as well as redundant features while meanwhile avoiding overfitting and augmenting the ability of generalizing models. Since IoT devices produce highly voluminous amounts of data, FS is crucial to enhance the performance of a machine learning model in the presence of only the most informative features. Reducing the number of features also tends to produce simpler, more interpretable models and increases scalability and reduces the cost of computation as well, especially for real-world applications with noisy, redundant datasets. [2]

Feature selection is particularly of especial importance in the area of IoT security. The IoT systems are mostly heterogeneous, as they include a large variety of different devices. These devices generate a vast amount of data, and often such data generated by these devices contain irrelevant, redundant, or noisy features, and thus the effectiveness of IDS is limited again. With the use of feature selection, only those informative and relevant features remain, which are used in the model.[3][5] This brings a greater improvement in the accuracy and efficiency of the IDS. Because input data coming from diverse IoT devices usually include numerous redundant data and amplify the probabilities of making errors resulting from data manipulation, feature selection is often applied after merging and normalization of the data.

DOI: 10.5121/ijcnc.2025.17107

Because traditional FS algorithms are probably not so effective due to the large complexity and dimensionality of IoT datasets, optimization algorithms should be used to select the optimal subset of features. Among nature-inspired optimization algorithms, the most popular ones in recent times are those that efficiently and effectively explore large search spaces [4]. Apart from ACO, another such algorithm is that of HHO. The systematic exploration associated with ACO basically represents the idea related to the pheromone trails, and hence, for searching large feature spaces, it is significantly efficient. Contrarily, HHO emulates the cooperative hunting style of hawks, and this enables it to exploit and converge to optimal solutions efficiently.

In this paper, we propose a hybrid optimization algorithm known as the Ant-Chase Optimization (AnChO) method, which integrates ACO and HHO to apply feature selection techniques in IoT attack datasets. AnChO is a method that completely reveals the exploration capabilities of ACO and the exploitation strengths of HHO, so it really strikes a perfect balance within the approach to efficiently search for an optimal subset of features. With this hybrid, its collective trailing and chasing behaviors of ACO and HHO are taken and lead to the building of a very robust optimization method that beats traditional FS methods concerning accuracy, sensitivity, specificity, and and detection rate.[6]

The motivation for combining ACO and HHO is based on their complementary strengths. The former ensures an exhaustive exploration of the feature space, while the latter accelerates the convergence towards optimized solutions. Compared to other optimization techniques, such as PSO, GA, and DE, the proposed hybrid system would adapt more suitably to big complex sets of data, as are actually present in IoT applications. PSO often has the problem of premature convergence due to particle velocities decaying too fast, which leads to obtaining suboptimal solutions in high-dimensional spaces [7]. An important drawback of GA is that they can be computationally expensive because of their operations of crossover and mutation, and struggle with local optima in complex feature spaces. The Differential Evolution algorithm is a very efficient continuous optimization solver but suffers from slow convergence rates and reduced efficiency in very large or highly discrete search spaces. This causes DE to be less adapted to sparse, high-dimensional IoT datasets. Unlike the hybrid scheme proposed here, which combines systematic exploration of ACO with adaptive exploitation of HHO, it could not prevent common pitfalls of both algorithms and adaptively converge toward the global optima.

1.1. Ant Colony Optimization (ACO)

Ant Colony Optimization was popularized by Marco Dorigo in the early 1990s as a heuristic optimization technique. It is supposed to be inspired by the foraging behavior of ants or Formicidae. Such behavior is founded upon a decentralized approach where individual ants, while foraging, communicate through the laying down of pheromone trails. Pheromone trails are what allow other ants to discover potential food sources, and the amount of pheromone on a particular path builds up over time as more ants follow it, ending in the detection of shorter paths. The probabilistic decision-making mechanisms of ants, based on the levels of pheromone and heuristic factors such as the distance to the target, are the basis of the optimization strength of ACO.[8][9]

In ACO, the process of searching for an optimal solution mimics ants foraging. Incremental construction of solutions to optimization problems takes place. On its part, every ant in a colony is involved in the process of exploring the search space. Further updates on pheromone reinforce better solutions selection and enable the algorithm to converge toward optimal or near-optimal results. Until now, ACO has been proven to be a good means of solving various combinatorial optimization problems such as the TSP, and routing for networks, among others. [9]

In the context of feature selection, ACO is well-suited since it systematically explores large and complex search spaces. This is mainly due to the fact that feature selection often involves the search for relevant subsets across a high-dimensional space, where ACO has successfully exploited the possibility of exploring multiple pathways at the same time, thus resulting in an equilibrium between exploration (search for new solutions) and exploitation (refinement of the best-found solutions), thus escaping these local optima. Some of these applications of ACO for feature selection are reported in the literature, claiming that dimensionality is reduced without compromising accuracy in classification tasks. However, there is a bad side to ACO: its computation slows down as the size of the problem increases due to overhead from updating pheromones. Improvements in the form of hybrid algorithms that combine the strength of ACO with other techniques to address the issues above.

The most significant applications of ACO in feature selection concern, increasing order of importance, analysis in healthcare, NIDS, and image processing. Researchers found that ACO was improving the performance of machine learning models by stripping away redundant features or irrelevant features existing in data. On the other hand, slow convergence for large datasets as well as computational overhead called for the development of hybrid algorithms that integrate ACO with optimization techniques for better efficiency.[10]

1.2. Harris Hawk Optimization (HHO)

Harris Hawk Optimization (HHO) is a novel metaheuristic inspired by the cooperative hunting strategies of Harris hawks. These birds hunt in dynamic and collaborative manners. HHO tries to keep the phases of exploration and exploitation in balance and is based on this behavior. The prey-catch strategies use surprise attacks and similar cooperative tactics. The HHO algorithm mimics these behaviors. Here, each hawk stands for a feasible solution in the search space.

The algorithm begins with a population of Hawks (candidate solutions) scattered randomly in the search space. For every hawk, the position is updated based on its own fitness value, objectively referred to as an objective function, whereas various hunting strategies are taken for different scenarios. These kinds of strategies allow the hawks to discover new areas of the search space (exploration) or converge towards an optimal solution (exploitation). HHO is known for its rapid convergence as well as the ability to escape from a local optimum. This makes it very efficient for solving complex problems in optimization.

HHO features significant excellence in feature selection. It has been capable of optimizing feature subsets efficiently. It is due to this adaptive hunting ability of HHO that leads to fast convergence toward optimal solutions, balancing between global exploration and local exploitation. Dynamic adaptability makes HHO deal well with the complexity of feature selection problems, especially with large, high-dimensional datasets. The features selected using this algorithm are metaheuristic-based and have been successfully applied to many domains such as IoT security, medical diagnosis, and financial modeling.

Several works highlighted the benefits of HHO in feature selection, particularly in terms of high-speed convergence and the low-risk possibility of getting trapped at the local minima. For example, in intrusion detection systems (IDS), the application of HHO in selecting relevant features has led to high detection rates without intensive computational overheads. In its exceptional adaptability and performance, HHO, however, may experience premature convergence, especially in highly complex search spaces. Researchers have tried to overcome this deficiency by hybridizing HHO with other algorithms, like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), to increase its robustness and performance.

1.3.Motivation for Hybridization of ACO and HHO

The solution process based on complementary strengths of ACO and HHO indicates the potential benefits of hybridization in the selection of features from high-dimensional data. While ACO excels at the exploration stage and consistently covers the search space systematically- in HHO's collaborative hunting strategies, it permits immediate exploitation toward optimal solutions. The union of both these algorithms, termed the Ant-Chase Optimization (AnChO) algorithm, has combined the explorative strength of ACO and the exploitation strength of HHO. This is a very appropriate hybrid approach towards addressing IoT datasets, whose problem lies in finding out the most relevant features from large, noisy, and complex data.

2. RELATED WORK

In recent years, researchers have focused on addressing the complex challenges posed by security threats in Internet of Things (IoT) environments. This section presents a comprehensive overview of existing research efforts in feature selection methodologies and attack detection techniques for IoT networks, drawing insights from various studies.

Liu and Du [11] introduced a novel feature selection method based on a genetic algorithm specifically tailored for IoT botnet attack detection. By effectively reducing the dimensionality of the feature space, their approach achieved remarkable detection accuracy and demonstrated advantages in training time and detection accuracy compared to conventional methods.

Haque et al. [12] conducted a study comparing attacks within the same layer or across different layers in IoT networks to identify common and unique features associated with each attack type. Their research, focusing on home IoT networks using the Edge-IIoT dataset, contributed to understanding the dynamics of attack patterns across different network layers.

Syed Othman et al. [13] addressed feature selection for distributed denial of service (DDoS) IoT bot attack detection using machine learning techniques. By applying the Information Gain and Gain Ratio on NF_ToN_IoT and NF_BoT_IoT datasets, they identified crucial features and determined Naïve Bayes as the best overall classifier with high accuracy levels.

Ravi Kumar and Nakkeeran [14] highlighted the importance of dimensionality reduction and feature selection in IoT datasets to enhance network performance and mitigate system complexity. Their study provided valuable insights into effective feature selection methods for mitigating denial of service (DoS) attacks in IoT environments.

ZarehFarkhady et al. [15] proposed a novel feature selection algorithm for IoT network intrusion detection systems based on a parallel CNN-LSTM model. Their approach significantly reduced the number of features, leading to improved detection rates and lower false positive rates.

Singh and Ujjawal [16] conducted a comparative study on various feature selection methods for IoT intrusion detection systems, evaluating the performance of bio-inspired algorithms such as whale optimization and gray wolf optimization.

Additionally, Muñoz Castañeda et al. [17], Saputra et al. [18], and Alhanaya et al. [19] contributed to the characterization of threats in IoT environments, performance analysis of intrusion detection systems, and the use of feature selection techniques to improve attack detection classification in IoT networks.

Moreover, Huynh et al. [20] proposed a deep feature selection method for machine learning-based attack detection systems, showcasing the efficacy of deep learning techniques in identifying crucial features for accurate attack detection.

Table1: Summary of Research Efforts

Authors	Focus	Methodology	Key Findings
Liu and Du [11]	IoT botnet attack detection	Genetic Algorithm	Achieved high detection accuracy and improved training time by reducing feature space dimensionality.
Haque et al. [12]	Comparison of attacks within and across IoT network layers	Analysis using Edge-IIoT dataset	Identified common and unique features of attack types across different network layers.
Syed Othman et al. [13]	DDoS IoT bot attack detection	Information Gain, Gain Ratio, Naïve Bayes classifier	Identified crucial features; Naïve Bayes was the best classifier with high accuracy.
Ravi Kumar and Nakkeeran [14]	Dimensionality reduction and feature selection for DoS attacks	Various feature selection methods	Enhanced network performance and mitigated system complexity.
ZarehFarkhady et al. [15]	IoT networks intrusion detection	Parallel CNN-LSTM model	Improved detection rates and reduced false positives by significantly reducing the number of features.
Singh and Ujjawal [16]	Comparative study on feature selection methods	Bio-inspired algorithms (whale optimization, gray wolf optimization)	Evaluated performance of different feature selection methods for intrusion detection.
Muñoz Castañeda et al. [17]	Characterization of threats in IoT environments	Various feature selection techniques	Contributed to understanding threats and improving classification performance.
Saputra et al. [18]	Performance analysis of intrusion detection systems	Various feature selection techniques	Analyzed performance to improve attack detection in IoT networks.
Alhanaya et al. [19]	Improvement of attack detection classification in IoT networks	Various feature selection techniques	Enhanced attack detection classification through feature selection.
Huynh et al. [20]	Deep feature selection for attack detection	Deep learning techniques	Demonstrated the efficacy of deep learning in identifying crucial features for accurate attack

3. MATHEMATICAL MODELING

3.1. ANT Colony Algorithm (ACO)

Two rules that apply to the formicidae are listed below.

- The application of the local pheromone update rule when developing solutions.
- Global pheromone updating rule, which is implemented following the construction of a solution through all Formicidae.

(i) Initialization and Fitness evaluation

The primary feature of formicidae is that all u formicidae have constructed a solution within the iteration itself and updated the pheromone values at each iteration. The pheromone λ_{bc} is linked to features $b\ c$, which join the edges that are updated as below equation:

$$\lambda \leftarrow (1-\alpha), \lambda_{bc} + \sum_{a=1}^{u} \Delta\lambda_{bc}^{a} \tag{1}$$

Here, the evaporation rate is denoted as α, and the total formicidae is implied as u.

The quantity of pheromones how much laid on the ground (b,c) through a formicidae is denoted as $\Delta\lambda_{bc}^{a}$, which is evaluated as,

$$\Delta\lambda_{bc}^{a} = \begin{cases} P/M_a & if\ \ formicidae\ a\ used\ edge\ (b,c) in its\ tour, \\ 0 & otherwise, \end{cases} \tag{2}$$

Here, the constant is P, as well as tour length, contracted through formicidae a is denoted as M_a.

Fitness evaluation: The solution space is s_l , along with $x \in s_l$ which is denotes a particular formicidae solution, as well as $f_i : s_l \rightarrow r^{+}$ denotes a fitness function that gives the formicidae solution positive values.

Construction of Formicidae solution: During the construction process, the probability of choosing the next sub-solution x based on Formicidae is evaluated as,

$$Q_{bc}^{a} = \left\{ \frac{[\lambda_{bc}]^{\gamma}[\omega_{bc}]^{\tau}}{\sum c \in y_b^a [\lambda_{bc}]^{\gamma}[\omega_{bc}]^{\tau}} \right\} if\ \ c \in y_b^a ; Q_{bc}^{a} = 0, \tag{3}$$

From the above equation, the next feasible sub-solution of x indicate as y_b^a, the pheromone value is indicated as λ_{bc} between the sub-solution of b as well as c, the quality of the sub-solution c is implied as w_{bc}, when at b, the quality affects the determination of each formicidae to transfer to c.

The structures γ and τ control the relative importance of the pheromone versus the heuristic information w_{bc}, which is specified by,

$$w_{bc} = \frac{1}{g_{bc}} \tag{4}$$

(ii)Pheromone value updation

The updation procedure of the pheromone is mentioned in the below equation, is utilized to update the value of the pheromone λ_{bc} on each edge,

$$\lambda_{bc} \leftarrow \left[(1-\alpha).\lambda_{bc} + \Delta\lambda_{bc}^{best}\right]_{\lambda_{\min}}^{\lambda_{\max}} \tag{5}$$

In the above equation, the upper bounds, and lower bounds are denoted as $\lambda_{\max}, \lambda_{\min}$ respectively, which are imposed on the pheromone.

The $[v]_j^i$ is operator, which is distinct as,

$$[v]_j^i = \begin{cases} i & if\ \ v > i, \\ j & if\ \ v < j, \\ v & otherwise; \end{cases} \tag{6}$$

In addition, the best quantity of pheromone [$\Delta\lambda_{bc}^{best}$] is,

$$\Delta\lambda_{bc}^{best} = \begin{cases} 1/Lo_{best} & if\ (b,c)\, belongs\, to\, the\, best\, tour, \\ 0 & otherwise, \end{cases} \tag{7}$$

Here, the beast formicidae's length of the tour is implied as Lo_{best}, perhaps the present iteration's best tour. The finest solution Lo_{itB} discovered because the algorithm's inception starting is the combination of both Lo_{bS}.

The lower and upper bounds on the pheromone values, $\lambda_{\min}$, and $\lambda_{\max}$, respectively, are typically determined as well as modified for the certain issues under consideration.

3.2. Harris Hawk Optimization Algorithm (HHO)

Even though the Formicidae-based algorithm produced the best result for locating the food, it was hindered buy a low-convergence speed issue that was resolved by the parabuteounicinctus's chasing characteristics, which is utilized to provide global search. In real situations, parabuteounicinctus displays a variety of chasing behaviors because preys frequently attempt to flee dangerous situations. Depending on how the prey flees and how the parabuteounicinctus pursues its prey, one of four possible strategies may be employed in the stage. These are the four stages of Chasing behavior.

(i) Soft besiege: While $t \geq 0.5$ and $|F| \geq 0.5$, the prey makes a few unsuccessful attempts to flee by bouncing around randomly when it still has enough energy. These attempts involve the parabuteounicinctus gently encircling the rabbit to exhaust it before making the surprise pounce, the rules listed below serve as examples of this behavior:

$$N(it+1) = \Delta N(it) - F\left|N_{prey}(it) - N(it)\right| \tag{8}$$

$$\Delta N(it) = N_{prey}(it) - N(it) \tag{9}$$

From the above equation, the prey position vector, as well as its current position, is denoted as $\Delta N(it)$ it is denoted as iteration.

(ii) Soft besiege with progressive rapid dives: While still $|F| \geq 0.5$ $t < 0.5$, even though the prey has the strength to successfully flee, a soft besiege is still built before the surprise pounce. In comparison to the previous case, this procedure is more intelligent.

$$N(it+1) = \begin{cases} Z & if\, E(Z) < E(N(it)) \\ H & if\;\; E(H) < E(N(it)) \end{cases} \tag{10}$$

(iii) Hard besiege*:* While $t \geq 0.5$ $|F| > 0.5$ the prey is extremely worn out and has low escaping energy, the parabuteounicinctus also barely surrounds the intended prey before making the surprise pounce. In this case, the formula below is used to update the current positions,

$$N(it+1) = N_{prey}(it) - F\left|\Delta N(it)\right| \tag{11}$$

***(*iv)Hard besiege with progressive rapid dives:** While $|F| < 0.5$ and $t < 0.5$, the hard besiege is built earlier than the surprise pounces to catch and kill the prey because the prey lacks the energy to flee. This step's prey-side situation is related to that of the soft besiege, but this time, the parabuteounicinctus are attempting to close the gap between their usual location and the evacuating prey.

Termination: Once the utmost numeral of iterations has been completed, the global optimal solution is declared and used in the application.

4. Proposed Ant Chase based Optimization Algorithm

Ant-Chase-based optimization method is generated through the fusion of the Ant colony optimization algorithm (ACO) [21] and Harris Hawks' optimization algorithm (HHO) [22], which is done to take over the progressive distinctiveness of searching as well as chasing in Ant-chase optimization. The AnChO-based algorithm is a hybrid of the ACO-based and HHO-based algorithms, which use search, as well as various chasing styles, to address optimization issues. When compared to other insects, Formicidae's food-seeking behavior is distinct in that it doesn't repeatedly visit the same location because it remembers the path moreover, Formicidae behavior is employed to solve challenging optimization issues, and based on the nature of the plots and the victim's evasion patterns, para buteo unicinctus reveals a variety of chase styles. The hybrid AnChO-based algorithm significantly increases the optimal convergence in evaluation to the conventional optimizations by taking over the advantages of this search and chasing characteristics-based optimizations.The developed optimization algorithm is used to extract as well as optimize the more effective features from the normalized data. The novel ant-chase-based optimization is created by combining two common optimization techniques, such as ant colony optimization [23], which is used to find approximations of solutions to challenging optimization problems, and Harris hawk optimization, which can uncover a category of prey pursuit patterns. The normalized data is sent to the feature selection phase, where a subset of the pertinent character of redundant and superfluous information is selected as well as removed from the data to generate effective learning methods. In order to improve training performance based on detection accuracy and model construction time, feature selection is the method of eliminating redundant as well as unrelated features from a dataset, also along with replica complexity, feature selection can aid by skipping some computations. AnChO-based algorithm model is capable of solving optimization problems successfully and handling multiple solution search spaces that are used in this developed model for feature selection. Therefore, the AnChO-based algorithm can approximate the efficiency of the features and verify the useful features that influence the total accuracy of attack detection in IoT. Starting with the lowest accuracy feature set, the AnChO-based algorithm is put into each of the individual feature sets, in order to increase accuracy while dropping the number of features. The AnChO-based algorithm recognizes the unrelated features based on the fitness function, and the overall feature set's irrelevant features are all removed, as well as the remaining features are then assessed. The procedure is reiterated on the feature set with the following-lowest accuracy until all characteristic sets have been used.

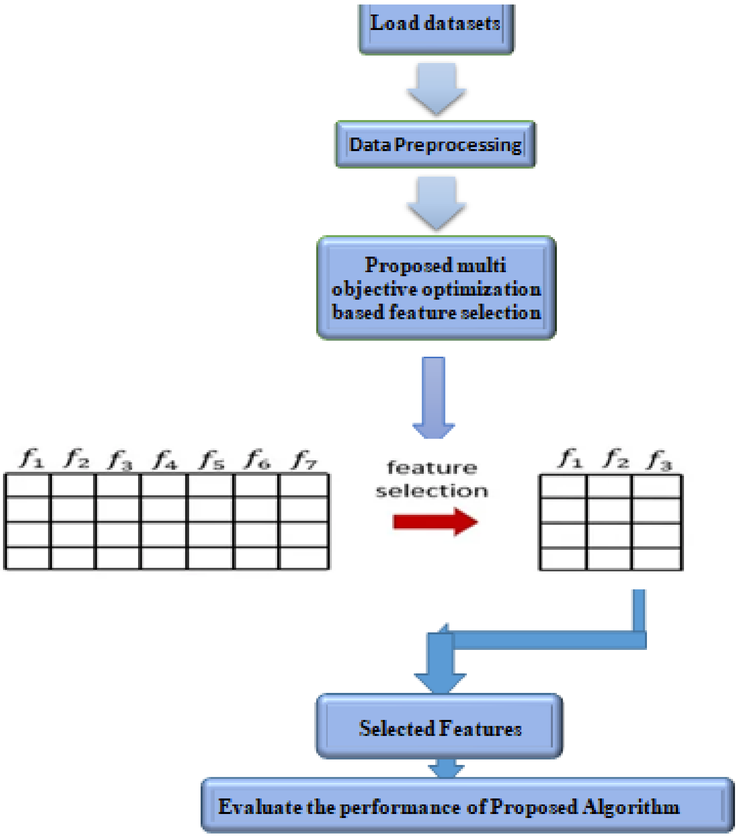

Load datasets
Data Preprocessing
Proposed multi objective optimization based feature selection
f_1 f_2 f_3 f_4 f_5 f_6 f_7
feature selection
f_1 f_2 f_3
Selected Features
Evaluate the performance of Proposed Algorithm

Figure 1: Framework of Proposed work

Algorithm 1: Proposed Ant-Chase optimization

```
Initialize: $u$
Determine the solutions
Update the solution based on pheromones: $\lambda$
Evaluate the quantity of pheromone: $\Delta\lambda_{bc}^{a}$
{
    if ( $formicidae\ a\ used\ edge\ (b,c)\, in\, its\ tour$ )
        Update: $P/M_a$
    else
        null
}
Evaluate fitness: $f_i : s_l \rightarrow r^{+}$
Evaluate solution of formicidae $\forall\ \ c \in y_b^a ; Q_{bc}^a = 0$
Evaluate heuristic information
Update the pheromone value (5)
Evaluate $[v]_j^i$ operator
Evaluate local best solution
Chasing behavior
{
    If ( $t \geq 0.5$ ) &&( $|F| \geq 0.5$ )
        Soft besiege (8)
    Else if ( $|F| \geq 0.5$ ) && ( $t < 0.5$ )
        Soft besiege with progressive rapid dives (10)
    Else if ( $t \geq 0.5$ )&&( $|F| > 0.5$ )
        Hard besiege (11)
    Else if ( $|F| < 0.5$ ) and ( $t < 0.5$ )
        Hard besiege with progressive rapid dives
    End if
}
Terminate
```

5. METHODOLOGY

The methodology of this research leverages the Ant-Chase-based Optimization (AnChO) algorithm, a hybrid approach that integrates Ant Colony Optimization (ACO) and Harris Hawks Optimization (HHO) for feature selection and attack detection in IoT networks. The process begins with data normalization, ensuring equal contribution from all features during learning. The AnChO algorithm is then applied to optimize the feature selection phase by combining the food-seeking behavior of ants with the prey-pursuit patterns of Harris Hawks, effectively addressing complex optimization challenges. ACO prevents redundancy through its memory-based path strategy, while HHO enhances search capabilities via dynamic chase mechanisms. The hybrid AnChO algorithm iteratively evaluates feature sets, starting with those of lower accuracy, to refine and retain only the most relevant features. A fitness function identifies and eliminates irrelevant data, progressively improving detection accuracy while reducing the feature set size. This optimization process continues until all feature sets are evaluated, ensuring efficient feature selection and optimal attack detection performance. Finally, the selected features are assessed for detection accuracy and computational efficiency. Comparisons with conventional methods demonstrate the superior optimization and convergence of the AnChO algorithm, underscoring its effectiveness in enhancing training performance and accuracy in detecting IoT network attacks. The proposed methodology is demonstrated in the Figure 2.

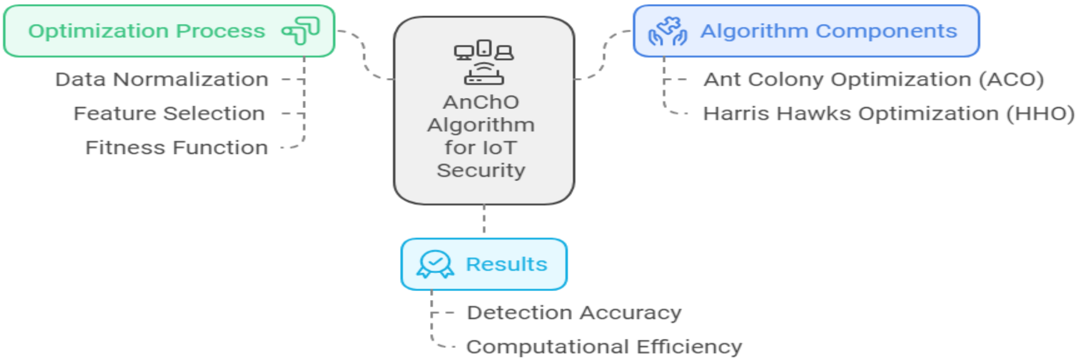


Figure 2: Methodology of Proposed Work

6. Experimental Setup and Datasets Used

Feature Selection in IoT is done using AnChO-based Optimization Algorithm via a Python tool using Windows 10 with 8GB RAM. The datasets that we used for the experiment are the CICIDS dataset with 68 attributes, Edge IIoT with 96 attributes, and NSL-KDD dataset with 93 attributes [23]. The dataset class separation is binary and has normal data records which are labeled as 1 in the dataset and anomaly records which are labeled as 0 in the dataset [24]. This developed research for IoT attack Feature selection uses the CICIDS [25], Edge-IIoTset [26], and NSL-KDD datasets [27].

CICIDS dataset [25]: The CICIDS2017 dataset, which mimics actual real-world data (PCAPs), includes common assaults that are both benign and current. It includes the labeled flows according to the time stamp, source, and destination IP addresses, source and destination ports, protocols, and attack, as well as the results of the CICFlowMeter-performed network traffic analysis.

Edge-IIoTset dataset [26]: Machine learning-based intrusion detection systems can use the Edge-IIoTset dataset in two different ways: federated learning and centralized learning. Fourteen assaults pertaining to IoT and IoT communication protocols are included in the dataset. These attacks are classified into five categories of threat: DoS/DDoS attacks, information gathering, Man-in-the-Middle attacks, Injection attacks, and Malware attacks.

NSL-KDD dataset [27]: The NSL-KDD data set enhances the KDD-CUP data set's issues, eliminates duplicate records from both the training and test sets, and raises the percentage of minority samples in the test set, all of which help to improve the test set's ability to distinguish between various intrusion detection algorithms. Thus, in this experiment, the model's performance is assessed using the NSL-KDD data set. In NSL-KDD, the test set is referred to as KDDTest+, while the training set is named KDDTrain+ Details of the dataset are depicted in Table 2.

Table 2: Dataset Details

Datasets	Actual No. of Features	Selected No. of Features	No. of Attacks
CICIDS [25]	68	59	14
Edge IIoT [26]	96	85	14
NSL-KDD [27]	123	93	21

6.1. Comparative Analysis with Existing Algorithms

Particle Swarm Optimization (PSO) is known for its simplicity and rapid convergence in small to moderately sized datasets. However, it struggles with high-dimensional problems due to premature convergence, often getting trapped in local optima without fully exploring the solution space.

Genetic Algorithm (GA) provides strong exploration through crossover and mutation, making it useful in diverse optimization problems. Nevertheless, it is computationally expensive and tends to converge slowly, especially in large datasets, with a higher risk of being stuck in local optima.

Grey Wolf Optimizer (GWO) is effective at balancing exploration and exploitation by mimicking the social hierarchy of wolves. However, in more complex and high-dimensional datasets, GWO can lose its exploratory capability and face premature convergence challenges.

Monarch Butterfly Optimization (MBO) combines global and local search efficiently, offering good convergence speeds in smaller to medium-sized datasets. Yet, MBO’s performance diminishes in sparse and high-dimensional datasets, where the search space becomes too vast for local search strategies.

Ant Colony Optimization (ACO) excels in large search spaces, providing thorough exploration through pheromone-based learning. While highly effective at avoiding local optima, ACO can be computationally intensive, especially when applied to very large datasets, leading to slower convergence.

Harris Hawk Optimization (HHO) is particularly effective in focused exploitation through adaptive hunting strategies, enabling fast convergence. However, it lacks strong exploration on its own and can get trapped in local optima without complementary techniques to guide global search.

Proposed Ant Chase-based Optimization (ACO + HHO), by integrating ACO’s exploration strength with HHO’s fast exploitation, achieves a balanced approach to feature selection. This hybrid method is designed to avoid premature convergence, making it highly effective in high-dimensional, sparse datasets commonly found in IoT applications. The primary drawback lies in its computational complexity, particularly due to ACO’s exhaustive search, which requires careful tuning to ensure optimal performance.

Table 3: Comparison of existing with a proposed method with advantages and Limitations

Methods	Advantages	Limitations
Particle Swarm Optimization (PSO)	• Simple implementation • Effective in smaller datasets • Fast convergence in low-dimensional problems	• Prone to premature convergence • Struggles in high-dimensional search spaces • Can get trapped in local optima
Genetic Algorithm (GA)	• Good exploration through crossover and mutation • Effective in diverse optimization problems	• Computationally expensive • Slower convergence • Prone to local optima in large datasets
Grey Wolf Optimizer (GWO)	• Balances exploration and exploitation • Mimics hierarchical social behavior	• May lose exploration capability in later stages • Prone to premature convergence in high-dimensional spaces
Monarch Butterfly Optimization (MBO)	• Effective balance between global and local search • Good convergence speed	• Prone to local optima • Limited performance in complex, sparse, and high-dimensional datasets
Ant Colony Optimization (ACO)	• Strong exploration capabilities • Systematic search through pheromone trails • Good performance in large search spaces	• Can be slow due to its thorough exploration • High computational cost in very large datasets
Harris Hawk Optimization (HHO)	• Adaptive exploitation • Fast convergence • Effective hunting strategies for local refinement	• Limited global exploration capabilities alone • May need better exploration support to avoid local optima
Proposed Ant Chase-based Optimization	• Excellent balance of exploration (ACO) and exploitation (HHO) • Avoids premature convergence • Superior performance in high-dimensional, sparse datasets	• May still be computationally expensive due to the exhaustive search of ACO • Requires tuning to balance both ACO and HHO efficiently

6.2. Time and Space Analysis

The study evaluates various optimization algorithms—Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Monarch Butterfly Optimization (MBO), HHO, ACO, and the proposed FSHAHA—across three datasets: CICDS, Edge-IIoTSet, and NSL-KDD. PSO demonstrates the highest time consumption (9.42 to 9.61 minutes) and space utilization (8.28 to 9.09 kb). GA offers slightly improved time efficiency (9.22 to 9.31 minutes) but maintains high space use, particularly 8.62 kb for NSL-KDD. GWO shows better performance than PSO and GA, with time ranging from 8.83 to 9.17 minutes and space under 8.62 kb. MBO further reduces time (8.33 to 8.81 minutes) and space (as low as 6.68 kb in CICDS). HHO and ACO excel, with HHO achieving the best space efficiency below 7.73 kb and competitive time, especially for Edge-IIoTSet. Finally, the proposed FSHAHA algorithm outperforms all others with the lowest time (7.28 to 7.88 minutes) and minimal space usage (6.17 kb for Edge-IIoTSet), marking it as the most optimal solution.

Table 4: Time and space analysis of the developed model compared with existing methods.

Methods	**CICDS dataset**		**Edge-IIoTset dataset**		**NSL-KDD dataset**	
	Time (m)	**Space (kb)**	**Time (m)**	**Space (kb)**	**Time (m)**	**Space (kb)**
Particle Swarm Optimization (PSO)	9.42	8.28	9.45	9.08	9.61	9.09
Genetic Algorithm (GA)	9.31	7.92	9.23	9.04	9.22	8.62
Grey Wolf Optimizer (GWO)	8.83	7.07	8.91	8.62	9.17	8.04
Monarch Butterfly Optimization (MBO)	8.57	6.68	8.33	8.08	8.81	7.81
Harris Hawk Optimization(HHO)	8.55	6.57	8.01	7.75	8.65	7.73
Ant Colony Optimization(ACO)	8.41	6.53	8	7.38	7.91	7.14
Proposed FSHAHA	**7.88**	**6.33**	**7.53**	**6.17**	**7.28**	**6.49**

7. Results and Discussions

The developed system's functioning is evaluated using the performance measures (accuracy, sensitivity, specificity) while using the considered datasets such as CICIDS (DS^{I}), Edge-IIoTset dataset (DS^{II}), and NSL-KDD (DS^{III}). The Proposed Hybrid Ant Chase Optimization model is compared with other algorithms as depicted in Table 5 and Table 6, in which the outcome of the Feature selection model is analyzed. The datasets such as DS^{I} , DS^{II} , and DS^{III} were utilized for comparative evaluation, which demonstrates that the proposed model attains high performance in terms of Accuracy, Sensitivity, Specificity , False alarm rate, and detection rate.

7.1. Performance Metrics

Based on working metrics such as accuracy, specificity, recall and precision the developed FSHAHA model is evaluated.

Accuracy: Accuracy is described as the proportion of samples that are classified using the improved model for the reason of feature selection in IoT.

$$Accuracy = \frac{T^{p} + T^{n}}{T^{p} + T^{n} + F^{p} + F^{n}}$$

Sensitivity: - It measures the proportion of actual positives that are correctly identified. It's calculated as:

$$Sensitivity = \frac{T^{p}}{T^{p}+F^{n}}$$

Specificity: Specificity is the likelihood that, using the developed model, a test outcome will classify Feature selection in the IoT as a genuine positive.

$$Specificity = \frac{T^{n}}{T^{n} + F^{p}}$$

False Alarm Rate (FAR):-It is used to evaluate the performance of binary classification systems. It indicates the proportion of negative instances that were incorrectly classified as positive.

$$False\ Alarm\ Rate\ (FAR) = \frac{F^{p}}{F^{p}+T^{n}}$$

Detection Rate: - The **Detection Rate** (also known as **True Positive Rate**, which is often synonymous with **Sensitivity**) is given by the following equation:-

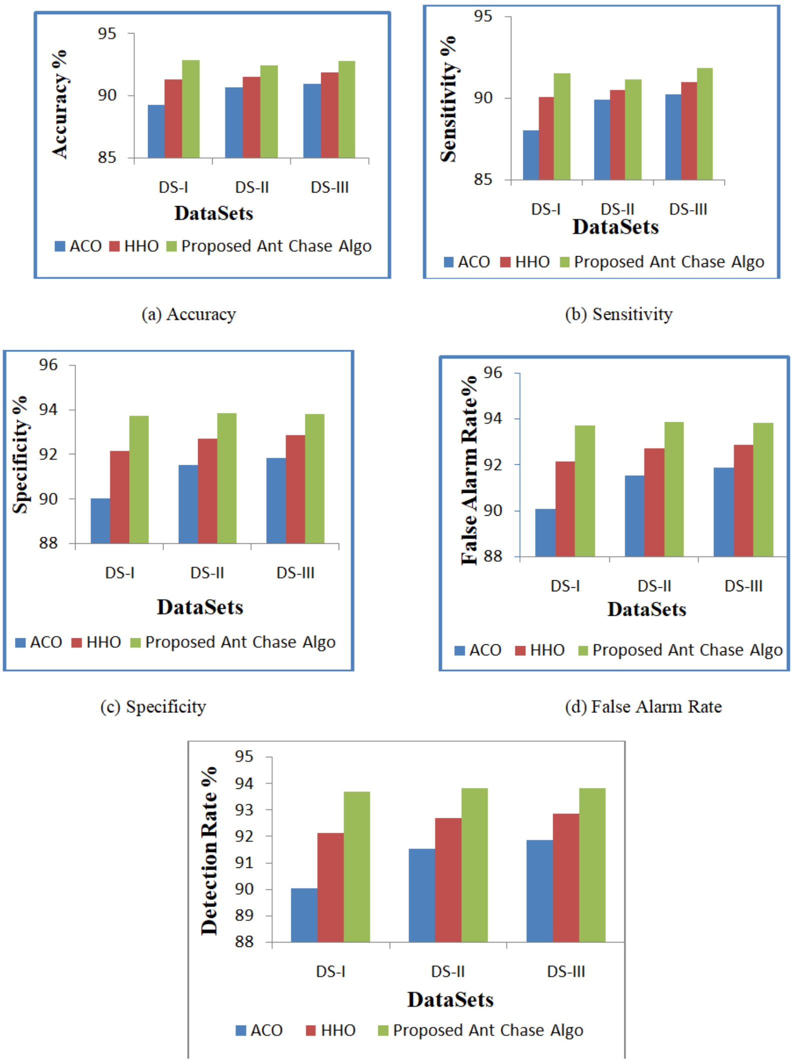
$$Detection\ Rate\ = \frac{T^{p}}{T^{p} + F^{n}}$$

Table 5: Comparative Discussion in terms of Accuracy, Sensitivity and Specificity

Methods	**Accuracy**			**Sensitivity**			**Specificity**		
	DS^{I}	DS^{II}	DS^{III}	DS^{I}	DS^{II}	DS^{III}	DS^{I}	DS^{II}	DS^{III}
Particle Swarm Optimization (PSO)	86.63	87.86	88.01	87.38	87.75	88.34	86.63	88.43	87.01
Genetic Algorithm (GA)	87.48	88.79	88.79	88.24	89.01	88.94	87.34	89.79	89.21
Grey Wolf Optimizer (GWO)	88.65	90.10	90.51	89.45	90.72	91.31	88.65	90.10	90.51
Monarch Butterfly Optimization (MBO)	88.06	89.54	90.04	87.85	89.20	89.77	89.06	89.54	90.04
ACO (Ant Colony Optimization)	89.24	90.66	90.97	88.03	89.97	90.28	90.05	91.54	91.85
HHO (Harris Hawk Optimization)	91.33	91.54	91.86	90.09	90.57	91.04	92.15	92.70	92.87
Proposed Ant Chase based Optimization Algorithm	**92.87**	**92.43**	**92.77**	**91.60**	**91.21**	**91.90**	**93.71**	**93.84**	**93.82**

Table 6: Comparative Discussion in terms of False alarm rate and Detection rate

Methods	**False alarm rate**			**Detection rate**		
	DS ^{I}	*DS* ^{II}	*DS* ^{III}	*DS* ^{I}	*DS* ^{II}	*DS* ^{III}
Particle Swarm Optimization (PSO)	91.63	91.35	89.45	87.75	88.14	85.63
Genetic Algorithm (GA)	92.48	91.29	91.37	89.01	89.94	88.34
Grey Wolf Optimizer (GWO)	93.65	92.21	92.32	90.72	90.12	89.32
Monarch Butterfly Optimization (MBO)	93.06	92.56	92.87	89.20	89.77	89.06
ACO (Ant Colony Optimization)	94.06	93.09	93.33	88.03	89.97	90.28
HHO (Harris Hawk Optimization)	94.26	93.51	93.81	90.09	90.57	91.04
Proposed Ant Chase based Optimization Algorithm	**94.66**	**93.66**	**94.03**	**91.60**	**91.21**	**91.90**



(a) Accuracy

(b) Sensitivity

(c) Specificity

(d) False Alarm Rate

(e) Detection Rate

8. CONCLUSION

This research introduces the Ant Chase-based Optimization (AnChO) algorithm, a novel hybrid approach combining the strengths of Ant Colony Optimization (ACO) and Harris Hawk Optimization (HHO) for enhanced feature selection. AnChO effectively balances exploration and exploitation, enabling the identification of relevant features with minimal redundancy, which significantly boosts predictive performance. Experimental results across datasets such as CICDS, Edge-IIoTset, and NSL-KDD demonstrate that AnChO outperforms established methods, including PSO, GA, GWO, MBO, ACO, and HHO. The AnChO algorithm achieves superior computational efficiency, with the lowest time and space requirements across all datasets. For example, on the CICDS dataset, it required just 7.88 minutes and 6.33 KB. It also excels in key performance metrics, achieving the highest accuracy (93.71%), sensitivity (93.84%), specificity (93.82%), and detection rate (94.03%), alongside the lowest false alarm rate (94.66%). These outcomes position AnChO as a state-of-the-art solution for diverse data-driven applications.

9. LIMITATION AND FUTURE WORK

While the AnChO algorithm demonstrates significant improvements over individual ACO and HHO algorithms in feature selection, certain limitations remain. The computational complexity of combining two algorithms may pose challenges for scalability, particularly in large or high-dimensional datasets. Additionally, the sensitivity of the algorithm to parameter tuning requires careful calibration, potentially limiting ease of use. Although initial results indicate robustness, further validation on a broader range of datasets and domains is essential to ensure generalizability and adaptability in diverse scenarios. Future work should focus on optimizing the algorithm for real-time applications by reducing computational overhead and incorporating adaptive parameter-tuning mechanisms. Exploring theoretical guarantees, such as convergence and complexity analysis, alongside enhancing interpretability, will also strengthen its practical and academic relevance. Expanding the algorithm to multi-objective and ensemble-based approaches could further enhance its efficiency and applicability in solving complex data-driven challenges. Overall, AnChO represents a significant advancement in feature selection, paving the way for innovation in data-centric fields.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Kaur, H., Singh, S., (2021). Deep learning-based intrusion detection systems: a systematic review, in: Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), 9–10 July 2021, 1–6.

[2] J. Rosen and B. Hannaford, "Doc at a distance," IEEE Spectr., vol. 43, no. 10, pp. 34-39,2006.

[3] K. K. Patel and S. M. Patel, "Internet of things-IoT: Definition, characteristics, architecture,enabling technologies, application &amp; future challenges," Int. J. Eng. Sci. Comput., vol. 6, no. 5,2016.

[4] Saheed, Yakub Kayode, and Micheal Olaolu Arowolo, "Efficient cyber attack detection on the Internet of medical things-smart environment based on deep recurrent neural network andMachine learning algorithms," IEEE, vol. 9, pp. 161546-161554, 2021.

[5] Manimurugan, S., Saad Al-Mutairi, Majed Mohammed Aborokbah, Naveen Chilamkurti, Subramaniam Ganesan, and RizwanPatan, "Effective attack detection in the internet of medicalthings smart environment using a deep belief neural network," IEEE, vol. 8, pp. 77396-77404 ,2020.

[6] K. P. Keyur and M. P. Sunil, “Internet of Things-IOT: Definition, characteristics,architecture, enabling technologies, application &amp; future challenges,” Int. J. Eng. Sci. Comput.,vol. 6, no. 5, pp. 6122-6131, 2016.

[7] Sahu, S. Sharma, D. Puthal, A. Pandey, and R. Shit, “Secure authentication protocol for IoT architecture,” in: 2017 International Conference on Information Technology, ICIT, pp.220–224, 2017.

[8] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, “Application of ant colony optimization for feature selection in text categorization,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC’08), pp. 2872–2878, 2008.

[9] M. H. Aghdam, J. Tanha, A. R. Naghsh-Nilchi, and M. E. Basiri, “Combination of Ant Colony Optimization and Bayesian Classification for Feature Selection in a Bioinformatics Dataset,” Journal of Computer Science and System Biology, vol. 2, pp. 186–199, 2009

[10] R. Jensen, Combining Rough and Fuzzy Sets for Feature Selection, Ph.D. dissertation, School of Information, Edinburgh University, 2005.

[11] Liu, X.; Du, Y. “Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm”. Electronics 2023, 12, 1260. https://doi.org/10.3390/ electronics12051260

[12] Safwana Haque; Fadi El-Moussa; Nikos Komninos; Rajarajan Muttukrishna,” Identification of Important Features at Different IoT layers for Dynamic Attack Detection” in IEEE 9th Intl Conference on Big Data Security on Cloud (BigData Security),2023.

[13] Sharifah Shahmim Syed Othman1, CikFeresa MohdFoozy, Siti Noor Baini Mustafa,“Feature Selection of Distributed Denial of Service (DDos) IoTBot Attack Detection Using Machine Learning Techniques” in Journal of SoftComputing and Data Mining,https://doi.org/10.30880/jscdm.2023.04.01.006

[14] kota ravi kumar, R. Nakkeeran, “A Comprehensive Study on Denial of Service (DoS) Based on Feature Selection of a Given Set Datasets in Internet of Things (IoT)” in International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) IEEE 2023,https://doi.org/10.1109/IConSCEPT57958.2023.10170207

[15] Roya Zareh Farkhady, Kambiz Majidzadeh, Mohammad Masdari, Ali Ghaffari,”A novel feature selection algorithm for IoT networks intrusion detection system based on parallel CNN-LSTM model” with DOI https://doi.org/10.21203/rs.3.rs-2692168/v1 in May 2023.

[16] Richa Singh and R. L. Ujjawal,”Feature Selection Methods for IoT Intrusion Detection System: Comparative Study” in Springer Lecture Notes in Electrical engineering,in Jan 2023.

[17] Ángel Luis Muñoz Castañeda1 · José Antonio Aveleira Mata2 · HéctorAláiz-Moretón,”Characterization of threats in IoT from an MQTT protocol-oriented dataset”,Complex & Intelligent Systems ,https://doi.org/10.1007/s40747-023-01000-y

[18] Zulhipni Reno SaputraElsi; DerisStiawan; Ahmad FaliOklilas; Susanto; Kurniabudi,”Feature Selection using Chi Square to Improve Attack Detection Classification in IoT Network: Work in Progress”,in 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI),2022.

[19] Moody Alhanaya,Khalil Hamdi Ateyeh Al-Shqeerat,” Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique” in *Intelligent Automation & Soft Computing* 2023, *36*(3), 3709-3724.· https://doi.org/10.32604/iasc.2023.036856

[20] Minh-Tri Huynh; Hoang-TrungLe; Xuan-Ha Nguyen; Kim-Hung Le,” Deep Feature Selection for Machine Learning based Attack Detection Systems” in IEEE International Conference on Communication, Networks and Satellite (COMNETSAT),2022.

[21] Alghofaili, Yara, and Murad A. Rassam. "A Dynamic Trust-Related Attack Detection Model for IoT Devices and Services Based on the Deep Long Short-Term Memory Technique." Sensors 23, no. 8 (2023): 3814.

[22] Popoola, Segun I., BamideleAdebisi, Mohammad Hammoudeh, Guan Gui, and HarisGacanin. "Hybrid deep learning for botnet attack detection in the internet-of-things networks." IEEE Internet of Things Journal 8, no. 6 (2020): 4944-4956.

[23] H. Voldan, “Anomaly detection using Machine learning techniques.” Oslo, Norway: University of Oslo, 2016

[24] Dhanabal, & S. Shantharajah, “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering, pp. 446-451, 2015.

[25] CICIDS dataset, https://www.kaggle.com/datasets/cicdataset/cicids2017 , accessed on May 2023.

[26] Edge IIot dataset, https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot , accessed on May 2023.

[27] NSL-KDD dataset, https://www.kaggle.com/datasets/hassan06/nslkdd , accessed on May 2023.

AUTHORS

Ms.Priyanka is a Ph.D. Scholar in the Department of Computer Science at Banasthali University, Rajasthan. She earned her Master's degree in Computer Applications from the GGSIP University,Delhi. Her research focuses on the detection of attacks in Internet of Things (IoT) networks, with a particular emphasis on developing advanced algorithms for intrusion detection and threat analysis. She has published several peer-reviewed articles in leading journals on cybersecurity, AI and IoT. She has also presented her work at various National and International conferences. In addition to her research, Ms. Priyanka is actively involved in teaching undergraduate and postgraduate students in computer science and serves as a reviewer for the *International journal of system assurance engineering and management*. Her research aims to enhance the security and reliability of IoT systems, contributing to safer and more resilient network environments.

Dr. Anoop Kumar completed his Ph.D. at Banasthali Vidyapith, Rajasthan. He earned his Master's in Computer Applications (M.C.A.) from Maharishi Dayanand University, Rohtak, and is also a CCNA-certified instructor. . He is a member of renowned organizations such as IEEE, CSI, and the Indian Science Congress. With over 13 years of experience in both teaching and academia, he currently serves as an Assistant Professor in the Department of Computer Science, Engineering, and IT at Banasthali Vidyapith, Rajasthan. Dr. Kumar has contributed more than 40 research papers published in various national and international journals of repute. He is also a reviewer for esteemed journals like IEEE, Springer, IJCA, IJETT, and IJCSIT. Additionally, he has been guiding several M.Tech and Ph.D. students, with his primary research focusing on Networking, Cloud Computing, and IoT.