Abstract- The Internet of Things (IoT) enables smart settings, which help human pursuits. Although the IoT has increased economic opportunities and made numerous human conveniences possible, it has also made it easier for intruders or attackers to take advantage of the technology by either attacking it or avoiding it. Therefore, the primary concerns for IoT networks are security and privacy. For Internet of Things (IoT) networks, several intrusion detection systems (IDS) have been developed thus far using various optimization techniques methods. But as data dimensionality has increased, the search space has grown significantly, offering difficult problems for optimization techniques like swarm optimization using particles (PSO). To overcome these obstacles, this work suggests an approach for feature selection dubbed enhanced to increase the sticky binary dynamic sticky binary particle swarm optimization's searchability, and particle swarm optimization (IDSBPSO) was developed. It introduced a method for decrease of the dynamic search space and various dynamic parameters (SBPSO). To identify malicious data flow in IoT networks, an IDS was developed using this methodology. The IoTID20 and UNSW-NB15 IoT network datasets were used to assess the proposed model. It was found that, even with less features, IDSBPSO typically attained accuracy that was either higher or comparable. Moreover, as compared to traditional IDSBPSO and PSO-based feature selection methods dramatically reduced computational cost and prediction time.

Keywords: intrusion detection system, internet of things, anomaly detection

I. INTRODUCTION

Internet-based services have increased dramatically since the advent of the Internet [1]. As a result, it is simple to handle and administer a large number of the actual hardware or systems that are able to access the Internet from a distance. Then, client behavior may be observed and recorded, future choices can be forecasted, and helpful services can be offered [2]. Several industries, Employing the Internet of Things include smart manufacturing, smart supply chains, smart homes, smart cities, smart healthcare, and smart retail (IoT). Several potential IoT uses include used in daily life are shown in Figure 1. By solving concerns with living conditions, such a smart environment's objective is to increase the productivity and valuable. [3]. However, as the network has grown more interconnected and complex, maintaining network security has become increasingly challenging. Security flaws are viewed by hackers as an opportunity to find and use IoT network vulnerabilities. However, consumers and organizations alike may suffer large financial losses as a result of network security breaches. Consequently, it is crucial to design

![Real world IoT applications chart](image-url)
Signature-based intrusion detection systems (SIDS) and anomaly-based intrusion detection systems are the two basic categories of intrusion detection systems (AIDS), looking through matching patterns in network packets to a database of signatures or traditional SIDS techniques. AIDS uses a machine learning (ML) approach to instruct the model appropriate behavior. Network activity is contrasted after that, with such typical conduct. Systems that identify intrusions based on anomalies are thought of as a dynamic anomaly detection approach that employs behavior-based detection. Actually, compared to other plans, the AIDS strategy has garnered greater attention [5]. The primary advantage of AIDS is the capability to identify unknown or zero-day assaults. Since anomaly detection seems to be the most practical method, most studies opt for it [6, 7]. However, it is still difficult to create effective IDS for IoT devices for the following reasons:

II. DATASETS FOR CYBER SECURITY

Most of the datasets now in use are obsolete and may not be effective for understanding the behavior of contemporary cyber-attacks. Furthermore, nothing is known about the features and recurrence patterns of current attacks. Handling quality issues in datasets related to cyber security Cybersecurity datasets may be missing information, be uneven or noisy, or contain occurrences that are inconsistent with a specific security incident. These dataset concerns have an impact on the efficacy of ML-based models and the quality of the learning process [8].

III. THE PROPOSED MODEL

This section suggests a more advanced method for choosing features for the building of an IDS-BPSO-based effective for IoT networks, and precise IDS. Eberhart and Kennedy proposed the Particle swarm optimization (PSO), a population-based stochastic optimization approach, was first used in 1995. The PSO algorithm is regarded as efficient Computational has a simple feature coding because to its justification, limited number of parameters, and less time-consuming execution to choose and fix major feature issues That's PSO that was initially proposed was continuous (CPSO), and it was utilized to address many continuous problems. The fundamental disadvantage of PSO is that the local minimum will be reached by all further particles created by a particle that becomes stuck in an ideal local minimum, leading to incorrect solutions. therefore, prior Maintaining particle diversity is crucial for growing the network.

The PSO approach uses particles to depict conclusions drawn from the particle population in the relevant space. The term "swarm" is applied to describe this group. When d is Each swarm particle has a d-dimensional velocity vi regardless of the dataset's number of features and is represented by the vector xi = (x1,1, x1,2,..., x1,d) (vi,1 , vi,2,..., vi,d). PSO moves about in the search area at random and iteratively updates its velocity and position to discover relevant characteristics in order to maximize efficiency. Up to that point, the finest personal and worldwide fitness values or pbest and gbest, are used to adjust the particle's velocity and location at each iteration. The location and speed of the particles are updated in accordance with Additionally, w stands for c1 and c2 are acceleration factors that give weight to the updated velocity's cognitive and social components while the inertia factor gives weight to the prior velocity.

There are as may be seen in, velocity has three components (1). The first element that demonstrates how the current direction has an impact. When everyone shares their best experiences, it helps to keep the swarm diverse because various particles typically have different momentums. Once a particle has reached the best position the swarm has so far found, the only thing that will enable it to continue seeking out better solutions is momentum. However, the other two are the social and cognitive components, which urge particles in the direction of their own and their neighbors' best experiences.

Binary PSO was created to address combinatorial issues, such as feature selection and job-shop scheduling. Instead of combining location and velocity in BPSO, in order to the chance of reaching the corresponding updated position values is calculated using velocity to obtain a new position, as seen in (3), when the quantity of iterations grows. At first, sk = 1 was established for k = 0. Adaptive SBPSO is an additional SBPSO version that is suggested to control a particle's capacity for exploration and exploitation. Ns, np, and ng are utilized in dynamic SBPSO to boost exploitation at the end while beginning to increase exploration. Traditionally, the evolutionary phase of A fixed-dimensional search space is used by a BPSO algorithm (where d is the quantity of unique traits.) Particle size optimization of the PSO algorithms or production count requires a lot of computing power when d is big. A search space reduction technique is helpful because it can lower the amount of computing power required for the PSO used.

Search space reduction strategy in Algorithm 1 Data: Particles

\[ AP \ best \ s \ P'_B \]

\[ = \{ p \ best \ 1, p \ best2, \ p \ best3, ..., p \ best \ v \}, \]

particles positions

\[ P, P = \{ X_k, X_k, X_k, ..., X_k \}, \]

\[ X_k = (x_k, x_k, x_k, ..., x_k) \]

where \( I = \{1,2,3, ..., V\} \), unmasked bit \( U_B = \{d_1, d_2, d_3, ..., d_W \} \)

\[ \text{Result: Updated } U_B \text{ and } P_P \text{ set} \]

For \( d \in U, \ not \ do \)

Since each component of this collection represents one unmasked bit, the mask in this algorithm is denoted by the notation U B, and the set P B represents the pbests of particles. By gathering data from each swarm's pbest, updated is the U B set. If a bit is not chosen by every pbest in the swarm, it is removed from U B. The set is updated continuously during stage of the algorithm's development. Some parts of Each time the mask-update method is used, U B are hidden utilized. A smaller search space is guaranteed by the mask-update method since only the bits in U B can change over time. It is then possible to rewrite the position-updating process as shown . The position of \( u \) value of 0 is assigned to that particle, signifying that it has been eliminated from the search space in accordance with the third condition, if \( d / U_B \), in order to save computation time and resources. The enhanced SBPSO techniques are depicted in grey blocks in Figure 2 to illustrate this improvement. Algorithm 2 illustrates the whole IDS-BPSO-based feature choice procedure. Every time, the mask is refreshed. K revisions in the proposed method, K
being the most iterations possible, in order to decrease the total number of features updated during an iteration.

Algorithm 2: Feature selection using a pseudocode method based on IDSBPSO

```
end
for particle = 1 to L do
  $p_i^{r+1}$ Update $i$: probability vector using Equation (6)
  $X_i^{r+1}$ Update $i$: position using Equation (11)
  $S_i^{r+1}$ Update $i$: stickiness parameter vector using Equation (7) evaluate the fitness value of $i$ using Equation (12);
  Update $p_{best}^{r+1}$ of $i$: particle using best fitness value;
  Update $g_{best}^{r+1}$ using $X_i^{r+1}$
end
k = k + 1;
$F_S$ Decode $g_{best}^{r+1}$;
```

Information: A set $F_S$ of selected features is determined by the training dataset's features, $K$ maximum generations, and $L$ maximum swarm sizes;

IV. IMPLEMENTATION AND RESULTS EVALUATION

This part covers the experimental setup, parameter setting, experimental findings, and evaluation metrics used to assess the performance of the suggested design. A review of the results from the concludes the proposed model.

A. Experimental Configuration

Performance of the suggested model was assessed using a Dell computer with an Intel (R) Core (TM) i7-6500U processor clocked at 2.50 GHz to 2.60 GHz. two cores, four It was evaluated. 16 GB of RAM and logical processors are used. Python was used to implement the feature selection and classification techniques (version 3.8). On the aforementioned system,Anaconda Navigator was set up for the experimental setup.

B. Assessment Metrics

The following metrics can be used to assess accuracy (AC), recall (RC), precision (PR), and F1-score of the proposed ML model ($F_1S$) the PR and RC harmonic mean is the $F_1S$. Meanwhile, the following is how AC, PR, RC, and $F_1S$ are calculated:

$$AC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$PR = \frac{TP}{TP + FP}$$

$$RC = \frac{TP}{TP + TN}$$

$$F_1S = \frac{2 \times (PR \times RC)}{PR + RC}$$

where each component of the equations above has the following definition:

- True Positive (TP): When The expected and actual values are both positive, this is referred to as a "True Positive" (TP).
- True Negative (TN): A signifying value that is negative in both the actual and forecasted values.
- False Positive (FP): When a positive result is predicted by the model but the actual value is negative.
- False Negative (FN): A value that is positive even though the model projected it to be negative. Additionally, since the suggested model is for energy-constrained IoT devices, an assessment metric based on computing time was used to verify the model's efficacy.
V. CONCLUSIONS

For the purpose of choosing features in classification, an enhanced binary PSO technique termed IDSBPSO is put forth in this study. A dynamic method and a search space reduction technique to control how momentum, pbest, and gbest contribute to particle motion were both adopted for IDSBPSO in order to improve feature selection performance. This led to a balance between exploitation and exploration. The recommended technique is utilized to build IoT networks that might benefit from an anomaly-based intrusion detection solution since it requires less expensive computational resources. The parameters of precision, F1 score, detection rate, and accuracy were compared computation time. The usefulness and efficiency of IDSBPSO were proved by the outcomes of two IoT network datasets under experimentation. Most of the time, IDSBPSO performed better. PSO-based feature selection techniques can be compared by achieving greater or comparable precision with fewer features. IDSBPSO specifically, which is made for energy-constrained IoT devices, dramatically less time spent computing as compared to benchmark PSO-based feature selection algorithms.

Because a wrapper-based method requires heavy calculation, even though the suggested Despite the fact that the IDSBPSO method greatly decreased calculation time when compared to Even using the benchmark PSO algorithms, it was shown to be slow.

computation period Some attacks have poor accuracy in categorizing themselves. The suggested method performs better on Furthermore, it is less appropriate for smaller-dimensional datasets, because it loses some useful characteristics from them, lowering accuracy. Therefore, the authors will aim to improve in the future.

REFERENCES

[1] Gupta, A.; Tewari, B.B. Security, privacy and trust of different layers in Internet of Things framework. Future Gener. Comput. Syst. 2020, 108, 909–920.

[2] Tewar, A.; Gupta, B.B. A novel ECC-based lightweight authentication protocol for Internet of Things devices. Int. J. Higperformance Comput. Netw. 2019, 15, 106–120. [CrossRef]

[3] Thakare, A.; Lee, E.; Kumar, A.; Nikam, V.B.; Kim, Y.G. PARBAC: Priority attribute-based RBAC model for azure IoT cloud. IEEE Internet Things J. 2020, 7, 2890–2900. [CrossRef]

[4] Ferrag, M.A.; Maglaras, L.; Ahmim, A.; Derdour, M.; Janieke, H.; RDTIDS: Rules and decision tree-based intrusion detection system for internet-of-things networks. Future Internet 2020, 12, 44. [CrossRef]

[5] Almomani, I.; Qaddoura, R.; Habib, M.; Alshehri, S.; Al Khayer, A.; Aljarah, H.; Faris, H. Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data. IEEE Access 2021, 9, 57674–57691. [CrossRef]

[6] Karami, A.; Guerrero-Zapata, M. A fuzzy anomaly detection system based on hybrid PSO-K-means algorithm in content-centric networks. Neurocomputing 2015, 149, 1253–1269. [CrossRef]

[7] Amouri, A.; Alaparthi, V.T.; Morgera, S.D. A machine learning based intrusion detection system for mobile Internet of Things. Sensors 2020, 20, 461. [CrossRef]

[8] Sarker, I.H.; Kayes, A.S.M.; Badsha, S.; Alqahtani, H.; Watters, P. Cybersecurity data science: An overview from machine learning perspective. J. Big Data 2020, 7, 41. [CrossRef]

[9] Keservani, P.K.; Govil, M.C.; Pili, E.S.; Govil, P. A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. J. Reliab. Intell. Environ. 2021, 7, 3–21. [CrossRef]

[10] Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity 2021, 4, 18. [CrossRef]

[11] Pajouh, H.H.; Dastghaibyfard, G.H.; Hashemi, S. Two-tier network anomaly detection model: A machine learning approach. J. Intell. Inf. Syst. 2017, 48, 61–74. [CrossRef]

[12] Al-Qatif, M.; Lasheng, Y.; Al-Sahabi, K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access 2018, 6, 52843–52856. [CrossRef]

[13] Ghazy, R.A.; El-Rabaie, E.S.M.; Dessouky, M.I.; El-Fishawy, N.A.; El-Samie, F.E.A. Feature selection ranking and subset-based techniques with different classifiers for intrusion detection. Wirel. Pers. Commun. 2020, 111, 375–393. [CrossRef]

[14] Shukla, P. ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things. In Intelligent Systems Conference (IntelliSys); Springer: Berlin/Heidelberg, Germany, 2017; pp. 234–240.

[15] Ullah, A.; Javaid, N.; Samuel, O.; Imran, M.; Shoaih, M. CNN and GRU based deep neural network for electricity theft detection to secure smart grid. In International Wireless Communications and Mobile Computing (IWCMC); IEEE: Piscataway, NY, USA, 2020; pp. 1598–1602.