International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

A Novel Hybrid Spotted Hyena-Swarm Optimization (HS-FFO) Framework for Effective Feature Selection in IOT Based Cloud Security Data

N. Sai Lohitha[1], M. Pounambal[2]
[1] School of Computer Science and Engineering,
VIT University, Vellore, TN, India
sailohitha.n@gmail.com
[2] Corresponding Author, School of Information Technology and Engineering,
VIT University, Vellore, TN, India
mpounambal@vit.ac.in

Abstract:
Internet of Things (IoT) has gained its major insight in terms of its deployment and applications. Since IoT exhibits more heterogeneous characteristics in transmitting the real time application data, these data are vulnerable to many security threats. To safeguard the data, machine and deep learning based security systems has been proposed. But this system suffers the computational burden that impedes threat detection capability. Hence the feature selection plays an important role in designing the complexity aware IoT systems to defend the security attacks in the system. This paper propose the novel ensemble of spotted hyena with firefly algorithm to choose the best features and minimise the redundant data features that can boost the detection system's computational effectiveness. Firstly, an effective firefly optimized feature correlation method is developed. Then, in order to enhance the exploration and search path, operators of fireflies are combined with Spotted Hyena to assist the swarms in leaving the regionally best solutions. The experimentation has been carried out using the different IoT cloud security datasets such as NSL-KDD-99 , UNSW and CIDCC -001 datasets and contrasted with ten cutting-edge feature extraction techniques, like PSO (particle swarm optimization), BAT, Firefly, ACO(Ant Colony Optimization), Improved PSO, CAT, RAT, Spotted Hyena, SHO and BOC(Bee-Colony Optimization) algorithms. Results demonstrates the proposed hybrid model has achieved the better feature selection mechanism with less convergence time and aids better for intelligent threat detection system with the high performance of detection.

Keywords: Internet of things (IoT), Security threats, Spotted Hyena, Firefly, Feature Selection, Data Redundancy.

I. Introduction:

In the real time application, role of IoT is very important in transmitting the heterogeneous data which includes health care, industrial automation, aerospace data and even agricultural data. These data are very vulnerable to many threats as the number of IoT increases exponentially. To safeguard the data, machine and deep learning algorithms based threat detection systems are designed and deployed [1]. To implement these systems efficiently, huge number of heterogeneous data are required. Due to the fact that IoT data frequently contains numerous redundant or irrelevant aspects, learning and detection performance are both significantly slowed down by these features. Selection algorithms for features are used as data processing techniques to choose the best subset from the overall feature set based on the specified criteria [2]. Feature selection can decrease the amount of the features by removing redundant ones, which speeds up learning and enhances the efficiency of intelligent systems [3].

The three categories of feature selection algorithms (FSAs) now in use are filter, wrapper, and embedding [4]. When comparing feature subsets, the filter and wrapper differ primarily in whether or not a classification algorithm is employed. Without using a classification algorithm, the filter approach picks subset of attributes as a pre-processing step. In general, it requires less computing than that of the wrapper and indeed the embedding. Consequently, it can quickly handle high-dimensional data. Due to the absence of a follow-up learning method, though, its performance frequently falls short of that of the wrapper and also the embedding [5-7]. The wrappers use a black box classifier or learning algorithm to grade feature subsets. This method is more effective compared to the filter method, but it requires more computing because a classifier is required to assess how well feature subsets do in classification. Additionally, the wrapper method requires strong search techniques to find the best feature subsets [8].

The hybrid technique, which combines the benefits of the filter and the wrapper in an efficient way, has strong

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

convergence when compared to the wrapper method and can provide a feature subset with excellent classification results [9]. The filter phase and even the wrapper stage are typically the two stages that make up a hybrid approach. By organising features based on various criteria, the filter stage aims to produce a smaller feature subset. The wrapper step is then utilised to employ certain local or global search algorithms to identify the best features from the decreased feature subset. Due to the aforementioned benefits, hybrid techniques are currently receiving increased attention from academics [10–12].

To design the hybrid combination, meta-heuristic algorithms can be applied for feature selection e.g., greedy algorithms based on heuristic search, exhaustive search optimization or particle swarm optimization [13-15]. Typically, these methods perform a transformation of the attributes space or a representation of the dataset into possible reduced patterns altering the original information of the data. However, these methods has drawbacks which are as follows

1. Ineffective feature selection criteria-based initialization technique that can lessen sensitivities towards the initial swarms
2. Most of the methods consider only the feature correlation but not redundant features in the data. Hence, there is a great probability of redundant data present in the features that can affect the performance of detection.

Considering the above mentioned problem, there arises the need for the hybrid feature selection method which combines advantages of wrapper along with filter method. In order to reduce the number of features and redundant features in the data space, this research suggests a novel hybrid Spotted Hyena and Firefly algorithm. Here is the paper's key contribution:

1. A new Spotted Hyena Based Swarm (FireFly) optimization technique is proposed which combines the merits of filter and wrapper method. The combination of the Spotted Hyena and Firefly algorithm increase the exploration space which in turn decreases the redundant features.
2. Proposing the two novel global seerach operators, based on the spotted hyena optimization that increase the searching space and the feature with the lowest relevance redundancy value is eliminated from the best feature selection.
3. Comprehensive experimentation has been carried out using the NSL-KDD datasets, UNSW and CIDCC-001 datasets and the proposed model's performance has been contrasted with that of another cutting-edge learning method.

The following is the way the paper is organised: Section 2 examines recent work on feature selection. Section 3 offers the suggested Feature selection algorithm. In Section 4, assessment is done by using it on real datasets and comparing it to various cutting-edge methods. Finally, Section 5 provides this paper's conclusion.

II. Related works:

In 2020, M. A. Elsayed and colleagues published PredictDeep, a methodology for anomaly detection and forecasting in massive data sets. It is entirely based on Graph Convolutional Networks (GCNs). This system was able to manage the complexity of clouds and delivered improved results in terms of the quick identification and forecasting of security breaches. The primary flaw in this approach, however, is that it fails to identify and categorise abnormalities in a variety of categories according to the modifications in system behaviour they bring about [16].

In a multi-user mobile edge-cloud computation offloading system using blockchain, D.C. Nguyen et al. (2021) looked at the challenges with both security and compute offloading. In order to increase offloading security, this framework offers a reliable access control system based on blockchain that can shield cloud resources from improper offloading practises. In order to achieve this, this framework created a double-dueling Q-network that is a sophisticated deep reinforcement learning algorithm. By reducing latency, energy use, and smart contract fees over the long run, this framework hopes to reduce overall system costs for all mobile devices. The disadvantage of this framework, however, is that as data volume grows, performance suffers [17].

Recurrent neural network-based deep learning techniques were examined by J. C. Kimmel et al. (2021) for their effectiveness in detecting attacks in cloud virtual machines. LSTMs and bidirectional RNNs were the main topics of this system (BIDIs). Depending on run-time, minor processes, and system elements like the CPU, memory, and disc consumption, these models gradually understand the behaviour of malware. The detection rates achieved by this framework are high. But is unable to continue to perform at the same level with heterogeneous data [18].

An advanced multilayered perceptron with recurrent neural network design that can understand the temporal context of multiple assaults was presented by G. Loukas et al. (2018). A mathematical model was created to identify, given network operating characteristics and deep learning model processing requirements, whether computation offloading is advantageous using detection delay as the criteria. Offloading reduces detection latency to a larger extent when the network is more dependable and when processing demands are higher.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

However, the increased communication overhead is this framework's main flaw [19].

Convolutional neural networks (CNN) and Grey Wolf optimization (GWO) were used to produce a hybrid data analysis model for network anomaly identification by S. Garg et al. (2019). Improved dropout functionality, explore, exploit, and initial population generation, GWO and CNN learning methodologies were upgraded to increase the model's capabilities. Regarding accuracy, false positives, and detection rate, this framework performs better. Its increased computational complexity, however, is a drawback of this approach [20].

In a cloud network, P. Abirami et al. (2022) showed how Deep Reinforcement Learning may be utilised to offload jobs while also identifying generalised attackers. For classification channels of Virtual Machine attacks, identity based piece - wise linear algorithms are used and this suggested system allows remote data monitoring techniques. Through the use of reinforcement learning, it can lead to better communication and lessen data secrecy.This framework's increased computational delay, however, is its only downside [21].

An method for Continuous Duelling Deep Q-Learning (C-DDQN) for cloud security was presented by Y. Tao et al. in 2022. The suggested Evolving Domain Adaptation Network and fine-tuning are the foundations upon which this architecture is achieved. Compared to not using transfer learning technologies, this framework exhibits superior convergence and learning capacity. However, this framework's increasing energy usage is its fundamental flaw [22].

An intrusion detection system using deep learning, cloud security was created by S. Hizal et al. in 2021 using convolutional NN along with recurrent NN. The cloud server cannot receive any detected or prohibited traffic under this approach. For a categorization of five classes, the suggested approach is 99.86% accurate. However, this framework's increased communication overhead is its main flaw [23].

A deep learning-based intrusion attack system to detect anomaly in three stages was described by C. Karri et al. in 2020. The system incorporates techniques from CNN, GANomaly, and K-means clustering. Network efficiency and automatic intrusion detection were both significantly increased by this system. The primary benefit of this framework is that it lowers computational complexity while not lowering overhead [24].

Stacking Contractive Auto Encoder (SCAE) technology was introduced by W. Wang et al. in 2022. The Support Vector Machine categorization method is the foundation around which this system is built. This framework allows for the automated learning of superior and more reliable low-dimensional characteristics from the unprocessed network traffic. The analytical overhead is greatly reduced by this framework. Greater detection performance is achieved using this strategy. However, this framework's main flaw is that it is unsuitable for environments that occur in real time [25].

Table : Summary of Literature Survey

Author's name	Proposed methodology	Merits	Demerits
M. A. Elsayed et al., (2020) [16]	GCNs	Security breaches that are promptly detected and predicted	However, does not reduce time overhead in a manner that is suitable for real-time environments
D. C. Nguyen et al., (2021) [17]	Mobile edge-cloud computation offloading system	Reduce system latency and energy consumption expenses over the long run.	When the amount of data increases, performance suffers.
J. C. Kimmel et al., (2021) [18]	Long Short Term Memory RNNs (LSTMs) and Bidirectional RNNs (BIDIs).	Obtains high detection rates	Heterogeneous data is not handled
G. Loukas et al., (2018) [19]	MLP and RNN	The detection latency is shorter	A higher cost of transmission
S. Garg et al., (2019) [20]	GWO and CNN.	A high rate of detection and precision	A more sophisticated algorithm
P. Abirami et al., (2022) [21]	Deep Reinforcement Learning	Reduces data confidentiality	Higher computational latency

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
Article Received: 12 October 2022 Revised: 07 November 2022 Accepted: 01 December 2022

Y. Tao et al. (2022) [22]	continuous duelling deep Q-learning	Rapidly converge	A rise in energy usage
S. Hizal et al.,(2021) [23]	K-means clustering, GANomaly and CNN algorithms	Simplifying the computations	Nonetheless, fails to decrease time overhead,
C. Karri et al., [24]	GANomaly and CNN algorithms	Simplifying the computations	However fails to reduce time overhead
W. Wang et al., (2022) [25]	Stacked Contractive Autoencoder and SVM	Lower analytical overhead	Unsuitable for real-time environments.

III. Proposed Technique:

The workings of the Spotted Hyena [26], Firefly algorithm [27], and Hybrid Feature selection technique are discussed in this section.

3.1Spotted Hyena Optimization :

Spotted Hyena Optimizer (SHO)

Both wet and dry habitats are home to spotty carnivorous hyenas. Swarms of spotted hyena, who are clever creatures, may attack individual wild animals like zebras and other species of wild beest. Building trust with the populace is the cornerstone of the hyena group's connections. The bulk of explorers are therefore made up of them.

3.2.1.(a)Enclosing the Target Phase

Spotted Hyena can very intelligently locate their preys. The Hyena being close to the prey is always termed as best candidate. The search space has to be spotted to achieve best search space.

The mathematical formulation of the behaviour is

$$\vec{D}_{sh} = |\vec{C}.\vec{P}_q(x) - \vec{P}(x)| \quad (6)$$

$$\vec{q}(x+1) = \vec{P}_q(x) - \vec{F}.\vec{D}_{sh} \quad (7)$$

Where $\vec{D}_{sh}$ is Hyena's separation from its intended prey, x shows the ongoing cycle, $\vec{E}, \vec{F}$ are vector coefficients since the population is dynamic in nature. P is the target prey's location vector and q is the hyena's position vector, || is the absolute value and . corresponds to the vector multiplication.

The vector $\vec{E}, \vec{F}$ coefficients are determined as

$$\vec{E} = 2.r\vec{d}_1 \quad (8)$$

$$\vec{F} = 2\vec{f}.r\vec{d}_2 - \vec{f} \quad (9)$$

$$\vec{f} = 5 - (cycle * \frac{5}{cycle}) \quad (10)$$

Where the $cycle = 1,2,3..cycle$

The value of the $\vec{f}$ function linearly decreases from 5 to 0 during the cycle of updating the best hyena.

$r\vec{d}_1, r\vec{d}_2$ are the arbitrary vectors that fall inside the range [0, 1].

Using the estimated values of the $\vec{E}$ and $\vec{F}$ vector, the location of the target prey is updated.

3.2.1.(b) Hunting Phase

The optimization algorithm for establishing the ideal solution space is found by analytically locating the behaviour of the spotted hyena.

$$\vec{D}_{sh} = |\vec{E}.\vec{P}_f - \vec{P}_k| \quad (11)$$

$$\vec{P}_k = \vec{P}_f - \vec{F}.\vec{D}_{sh} \quad (12)$$

$$\vec{C}_f = \vec{P}_k + \vec{P}_{k+1} + \cdots + \vec{P}_{k+N} \quad (13)$$

$\vec{P}_f$ be the finest spotted hyena's location

$\vec{P}_k$ the location of other hyenas in the search area

N is population size of hyenas as a whole

$$N = CountVal(\vec{P}_f, \vec{P}_{f+1}, \vec{P}_{f+2}, \ldots \vec{P}_{f+M}) \quad (14)$$

M is a carefully dispersed vector of values between [0.5, 1]. $CountVal$ is number of potential solutions, $\vec{C}_f$ is a group of solutions that come close to being the best option..

3.2.1.(c)Attacking the Target

Prior to the intended prey being assaulted, the vector $\vec{f}$ can be lowered to zero.

The spotted hyena group takes the prey at| $|F| < 1$.

This may be stated numerically as

$$\vec{P}(x+1) = \frac{\vec{C}_f}{N} \quad (15)$$

As more search spaces are explored, the location of the vector $\vec{P}(x+1)$ shifts until the optimal answer is found.

3.2.1.(d) Search for Target

A global searching mechanism is used by the spotted hyenas to look for their prey when clustered together. The algorithm is ended when the requirement of $|F| \geq 1$ is met.

3.2.1(e) ADVANTAGES OF SHO ALGORITHMS

1. High exploration and exploitation rates in comparison to other meta-heuristic algorithms as the Grey Wolf Optimizer, Genetic Algorithms, and even particle swarm optimization [41].
2. Less time-consuming.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

In Figure 3, the SHO's operational mechanism is shown.

Algorithm 1	**Pseudocode of Spotted Hyena Optimizer**
Input	P_i be the population of spotted hyenas $(i = 1,2,..n)$
Output	best search direction
1	set the vector's parameters. f, E, F and N
2	Check each search path's fitness function
3	P_f = the ideal search route
4	C_f = the group of solutions that is very near to the ideal solution
5	**while** (N, No.of iterations) **do**
6	**for** each individual search path **do**
7	depending on Eq, adjust the current position (10)
8	**end for**
9	update the values of f, E, F and N
10	Ensure that the search route does not surpass the threshold
11	compute the fitness function for each search path once again.
12	update the value of P_h if the solution outperforms earlier searches and produces the best outcome
13	update the cluster solution C_f with respect to the new P_f value.
14	$x = x + 1$
15	**end while**
16	return P_f
17	**End**

Figure 3 Spotted Hyena Optimization Algorithm Pseudocode

3.2 Fire Fly Swarm Optimization :

The swarm intelligence algorithm family, including the Firefly algorithm, is thought to exist. During the summer evenings, fireflies, sometimes known as lightning bugs, are frequently seen flashing their lights in the sky. A mating partner may be attracted by a firefly's flashing habit, or a firefly may use it to hide from predators. The fact that the strength of the light I get from a firefly reduces as it moves further away from a brighter one is another key property of fireflies. As the distance grows, the air also influences the light intensity by absorbing it. Because of this, the fitness value and the value of light intensity are closely related. The complexity of fireflies' natural behaviours, however, prompts three presumptions to be made in order to create the algorithm's basic operating principle. According to the following assumptions:

1. Regardless of their sex, all fireflies were thought to be unisex and attracted to one another.
2. As the distance between two fireflies grows, attractiveness generally decreases and is inversely correlated with brightness.
3. Using the objective function's workable solutions, the brightness or light intensity is calculated.

The suppositions make it abundantly evident that the intensity of light I(r) of fireflies is inversely linked to the distance r since it diminishes as distance rises and again because light also gets absorbed when it travels through the air. As a measure of light absorption, the symbol y is employed. Equation (4) displays the variation in firefly light intensity I(r) as a function of distance r.

$$I(r) = I_0 e^{-yr2} \quad (7)$$

where I_0 is the attractiveness parameter and the starting value of the source's intensity β can be defined in two different ways as shown in

$$\beta(r) = \beta_0 e^{-yr2} \quad (8)$$

β_0 is denoted as attractive parameters at the starting distance of 0.

The behavioural rule for computing firefly positions are given in the equation below

$$x_{i+1} = x_i + \beta(r(i,j))(xj - xi) + AE \quad (9)$$

Where "A is the randomization factor and E is the random number vector and both the factors are derived from the Gaussian distribution. x_i is the i^{th} position of the firefly and x_{i+1} second term represents the value of attraction".

3.4 Proposed Hybrid HS-FFO Operator:

The primary flaw in the old-fashioned firefly and SHO method was discovered to be the trapping mechanism at local minima. In order to address the trapping issues, this research proposes a novel hybrid model that combines the SHO and firefly algorithms. As the primary engines for feature selection in this instance, we have used the binary SHO optimization and firefly method. The primary operator in HS-FFO is a hybrid operator that combines the property that attracts firefly with the exploration stages of the SHO algorithm, including the enclosing prey and hunting method. The two goal functions of decreased features and high classification accuracy are present in the multi-objective optimization function of feature selection, which is thought to provide the highest performance. The fitness function is therefore provided as follows for each iteration.

$$\text{Fitness Function} = Y\alpha(A) + \beta\left(\frac{F}{N}\right) \quad (14)$$

Where $Y\alpha(A)$ is categorization accuracy as a function, S is the multiple linearities of the chosen feature vectors and N is the total number of characteristics. μ is the primary role that signifies accurate categorization and β is Feature subsets' length. Figure 5 provides the pseudo code that reflects the proposed model's operational procedures.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

Pseudo-Code for the Proposed HS-FFO Feature Selection Method

```
1   Set the SHO Population be Xi={x1, x, x1,, x1,..... xn} -Feature vectors
2   Using Eqn, the prey fitness function is determined for each search (13)
3   Xbest_Value is the global best position
4   While(t<max_iter is the maximum iteration)
5   For each search mechanism
6   Update D,, and p,co-efficient vectors U and C
7   If (p<0.5)
8   If(U<1)
9   Current position is updated using the Eqn(7)
10  Elseif (U>1)
11  The current search position is updated along with the random search prey agent (XRandom_Position)
12  End if
13  Else if (p>0.5)
14  Current search position is updated by using the equation (13)
15  endif
16  End
17  An exploration agent's analysis if it ventures outside of the search space
18  Equation is used to determine the fitness function for the random search position (14)
19  If (best)
20  Update Xbest_Value,
21  If (Xbest_Value is not equal to fitness function
22  Update the position using the eqn(6)
23  Else
24  return Xbest_Value
25  End
26  End
```

Problems with entrapment are eliminated by the aforementioned hybrid operator (See Line number 22). The HS-FFO feature selector has an extremely basic form, according to the suggested method. The performance of classifier models with the least optimal feature selection was not compromised in the development of the proposed method to include the fitness function. Comparing this approach to other ones, the computational complexity and time of calculation have been decreased.

IV. Experimental Setup

Real-time experiments are conducted in the manner described in Section. Different data traces from about two months were gathered and used for additional analysis. In Tab. 4, the specifics of the data traces that were gathered throughout the experiment are provided.

The suggested model has been tested and trained using 32,273 records of information that were gathered over the course of two months. In order to develop the study, Python-Tensorflow 1.3 was used. The version has a 2.4 GHz i7 processor, a 2TB hard drive, 16GB of RAM, and a 2GB NVIDIA Geoforce.

Table 4: The overall amount of data traces that were captured throughout the experiment

Sl. No.	Details of the information	No Traces were recorded per day	Total Traces
01	No. of Normal data	19,600	
02	No. of DoS/DDoS data	17,225	
03	No. of Sybil data	15,400	62,273
04	No. of Wormhole data	5225	
05	No. Probe data	1223	
06	No. of RPL	1200	
07	Number of	2400	

In order to confirm the effectiveness of the suggested model, additional universal benchmarks including UNSW-NB15 [28], NSLKDD [29], and CIDDS-001 [30] are utilised to test the model under various situation. These data sets include real-time traffic subject to various attacks: The CIDDS-001 characteristics span an area of 20,000 DoS records and 80,000 records on average. The UNSW-NB15 benchmarks consist of regular and attack data and comprise 49 features, one class label, and over 1,75,000 examplesEach instance of legitimate traffic in NSL-KDD has 37,000 label characteristics, compared to 45,332 for instances of malicious traffic.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

4.1 ***Performance indices***

In terms of precision, recall, specificity, and f-score, this section assessed the proposed model's feature selection performance. The performance metrics are measured using the following formulae

$$Accuracy: \frac{TP+TN*(100)}{Total\ Testing\ Samples} \quad (9)$$

$$Precision = \frac{TP}{TP+TN} \quad (10)$$

$$Recall = \frac{FP}{TP+TN} \quad (11)$$

$$Specificity = \frac{FN}{FN+TP} \quad (12)$$

TP → True positive values, TN→ True Negative values, FP→False positive and FN→ False Negative values.

4.2.1 RESULTS AND DISCUSSION

Using the real-time datasets, we developed HS-FFO models for the best feature selection and carried out 10 iterations, as shown below. A ratio of 80% training and 20% testing was used for the test datasets. Additionally, the model's hyperparameters were tweaked through the HS-FFO procedure to produce the optimum accuracy outcomes. The output batch size is set to 50, the learning rate is 0.0001, and the epochs are optimised to 100.

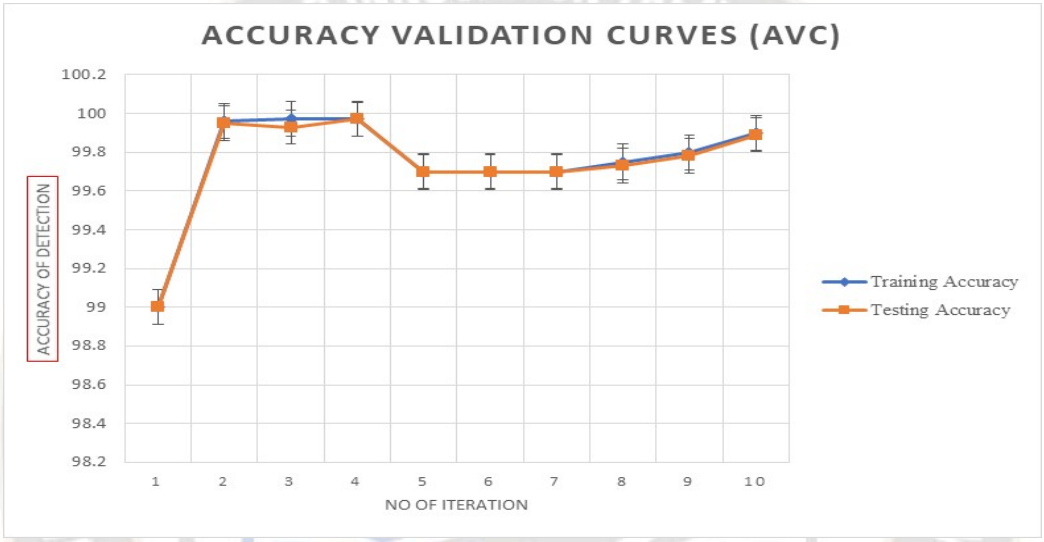

Figure 5: Using real-time datasets at 50 epochs, the suggested HS-FFO models' accuracy validation mechanism.

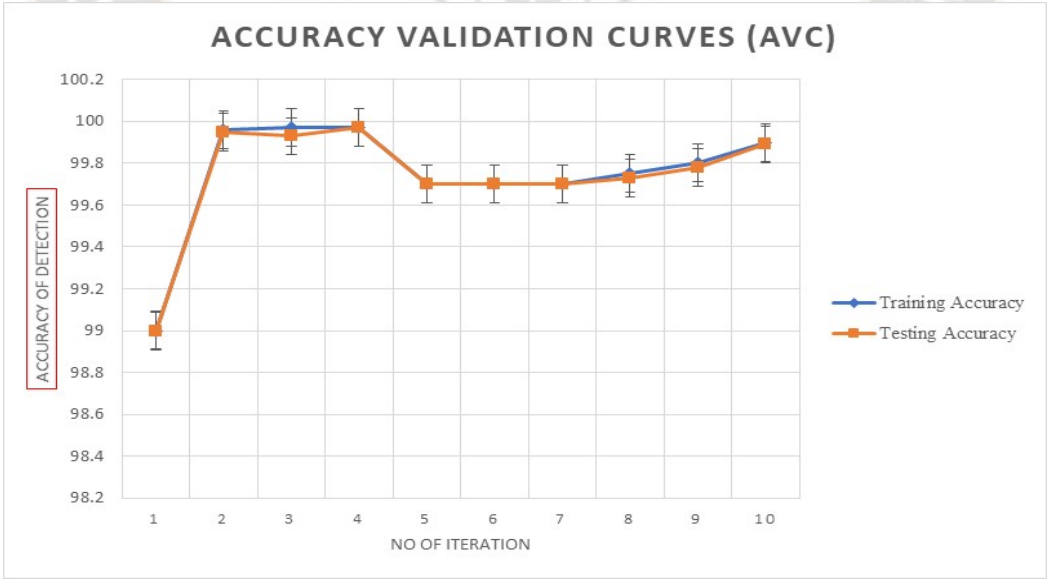

Figure 6: The real-time datasets are used as an accuracy validation mechanism for the suggested HS-FFO models to identify dos assaults at 50 epochs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

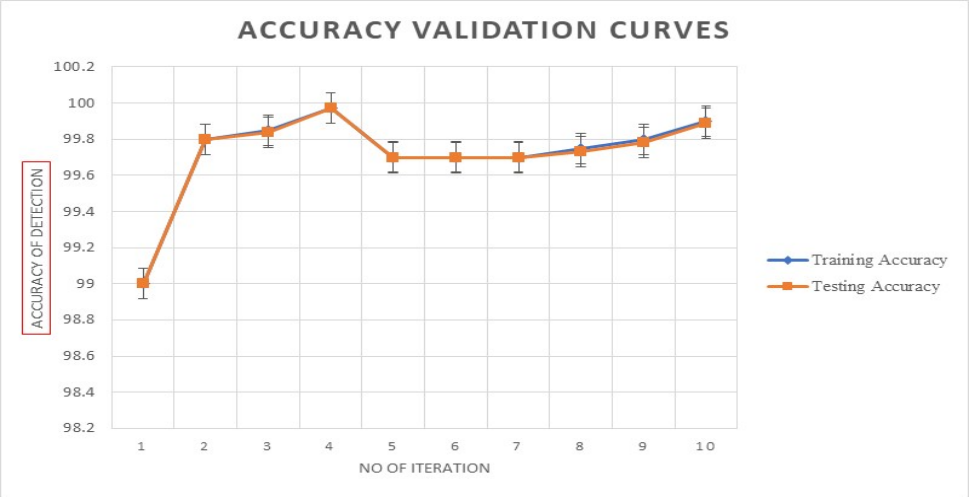

Figure 7: Real-time datasets are used as a precision validation mechanism for the developed HS-FFO models to identify DDoS attacks at 50 epochs.

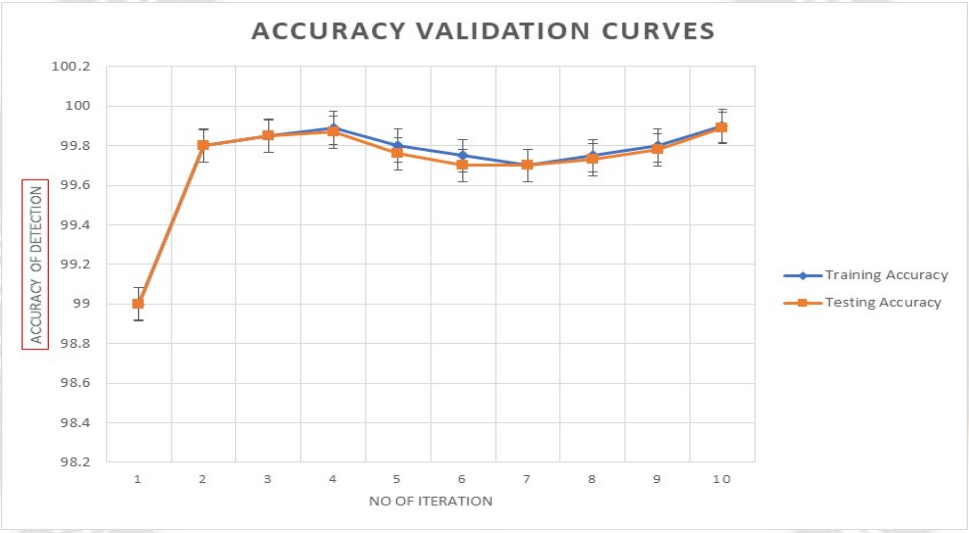

Figure 8: Using real-time datasets to validate the proposed HS-FFO models' accuracy in detecting Sybil assaults at 50 epochs

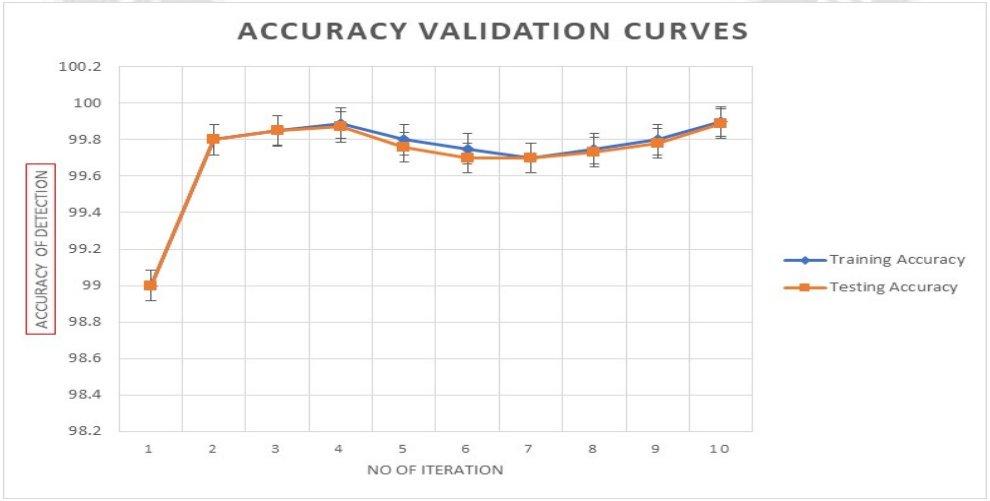

Figure 9: Real-time datasets are used as a reliability validation system for the presented HS-FFO models to identify wormhole assaults at 50 epochs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

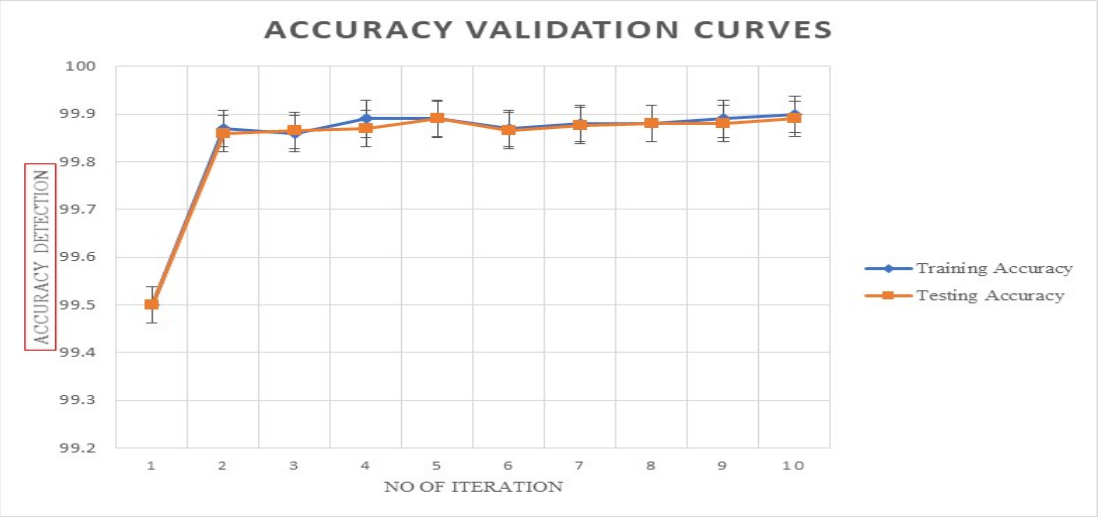

Figure 10: Real-time datasets are used as an accuracy endorsement mechanism for the developed HS-FFO models to identify RPL and URP assaults at 50 epochs.

The attribute selection accuracy (%) for the suggested learning models utilising the real-time testbeds is shown in Figs. 5 to 10. The overall average feature extraction accuracy for identifying the various network attack categories is 99.89%. Additionally, with the robustness of the learning network, optimised 50 epochs showed greater accuracy, nearly equal to 99.89%. Additionally, we calculated the additional performance measures, which are shown in Tab. 5 below:

Table 5: Feature selection performance of the proposed model under different assaults

Sl.no	Types of assaults	Precision	Recall	Sensitivity	F-Score
01	DoS Attacks	0.9986	0.9978	1.00	0.9815
02	DDoS Attacks	0.998	0.998	0.994	0.9814
03	Sybil Attacks	0.9987	0.9988	0.978	0.9786
04	Wormhole attacks	0.9986	0.9992	0.989	0.956
05	Probe Attacks	0.9987	0.978	0.982	0.966
06	RPL	0.988	0.976	0.978	0.9672
07	URP	0.986	0.967	0.977	0.9745

The effectiveness of the proposed model's feature selection under various attacks is shown in Tab. 5. The suggested model successfully detects assaults with good precision, recall, sensitivity, & f-score. Using a real-time situation, the optimised learning model has demonstrated its effectiveness in feature selection. The proposed model was trained using several benchmarks, as indicated in section 4.1, and performance metrics were calculated for a variety of scenarios.

Table 6: Feature selection performance of the recommended model utilizing the UNSW-NB15 benchmarks

Sl. no	Benchmarks Used	Attack types	Performance Metrics				
			Acy (%)	Pcn (%)	Rll (%)	Spy (%)	F-score (%)
01	UNSW-NB15	Dos	99.88	99.96	99.91	99.91	99.25
		DDos	99.90	99.83	99.91	99.43	99.65
		RPL	99.95	99.82	99.32	99.50	99.23
		URL	99.89	99.35	99.4	99.73	99.26

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

Table 7: Feature selection performance of the proposed model recommended model utilizing the NSL-KDD datasets benchmarks

Sl.no	Benchmarks Used	Attack types	Performance Metrics				
			Acy (%)	Pcn (%)	Rll (%)	Spy (%)	F-score (%)
01	NSL-KDD DATASETS 99	Dos	99.88	99.96	99.91	99.91	98.25
		DDos	99.90	99.83	99.91	99.43	99.65
		RPL	99.95	99.82	99.32	99.50	99.23
		URL	99.89	99.35	99.4	99.73	99.26

Table 8: Feature selection performance of the recommended model utilizing the CIDCC 001 benchmarks

Sl.no	Benchmarks Used	Attack types	Performance Metrics				
			Acy (%)	Pcn (%)	Rll (%)	Spy (%)	F-score (%)
01	CIDCC Datasets	Dos	99.88	99.96	99.91	99.91	99.25
		DDos	99.90	99.83	99.91	99.43	99.65
		RPL	99.95	99.82	99.32	99.50	99.23
		URL	99.89	99.35	99.4	99.73	99.26

With the use of the benchmark datasets, the evaluation criteria of the suggested algorithm were calculated and are shown in Tabs. 6 through 8. Performance measures like accuracy, precision, recall, sensitivity, and F-score have shown a very steady efficacy, ranging from 99% to 100% in feature selection against various kinds of assaults. Tab. 6 displays the performance metrics of the suggested algorithms utilising UNSW-NB15 Benchmarks. Similarly, Tab. 7 shows the performance indicators of the suggested method in feature selection utilising NSL-KDD datasets 99, where accuracy varies from 99.88% to 99.90%, precision ranges from 99.82% to 99.96%, recall ranges from 99.32% to 99.91%, sensitivity ranges from 99.43% to 99.91%, and F-score ranges from 98.23% to 99.65%. In addition, Tab. 8 displays a very good result in feature extraction within feature selection against multiple categories of attacks for the proposed model utilising the CIDCC datasets 2017. The outcome of the experiment demonstrates that the combination of the Firefly algorithm and the Spotted Hyena optimizer has demonstrated good performance in lowering the significant false alarm frequency.

4.2.2 Compare and contrast

We have contrasted the current deep learning algorithms including PSO(particle swarm optimization) [32], BAT [33], Firefly [34], ACO(Ant Colony Optimization) [35], Improved PSO [36], CAT [37], RAT [38],SHO [39] and BOC(Bee-Colony Optimization) [40] algorithms for the better feature selection performance under different datasets.

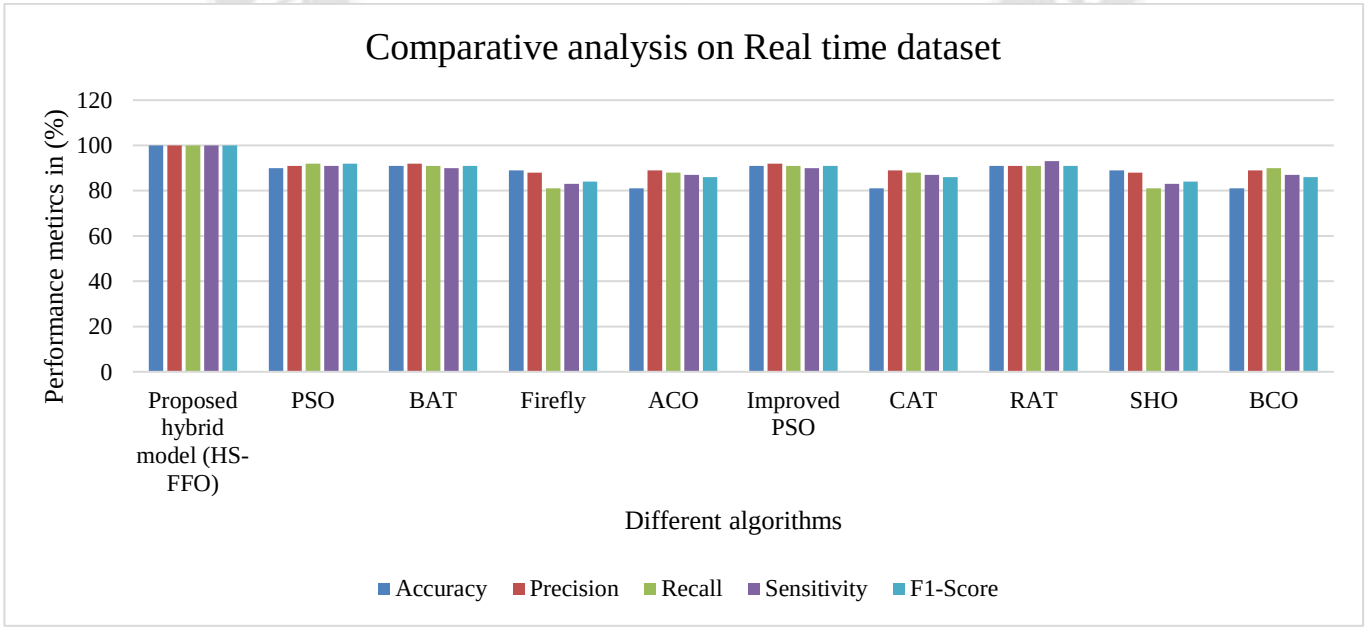

Figure 11: Proposed framework's Feature selection performance comparison among different algorithms using the real-time datasets

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
Article Received: 12 October 2022 Revised: 07 November 2022 Accepted: 01 December 2022

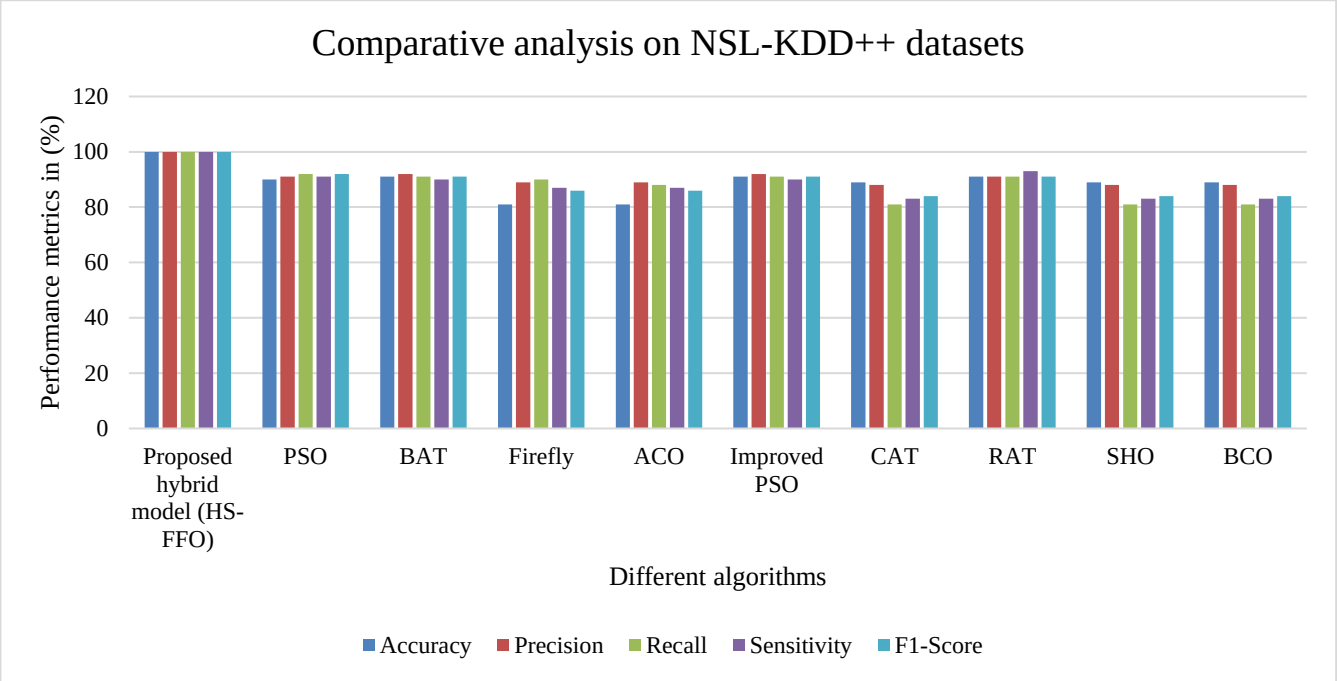

Figure 12: Proposed framework's Feature selection performance comparison using the NSL-KDD++ datasets

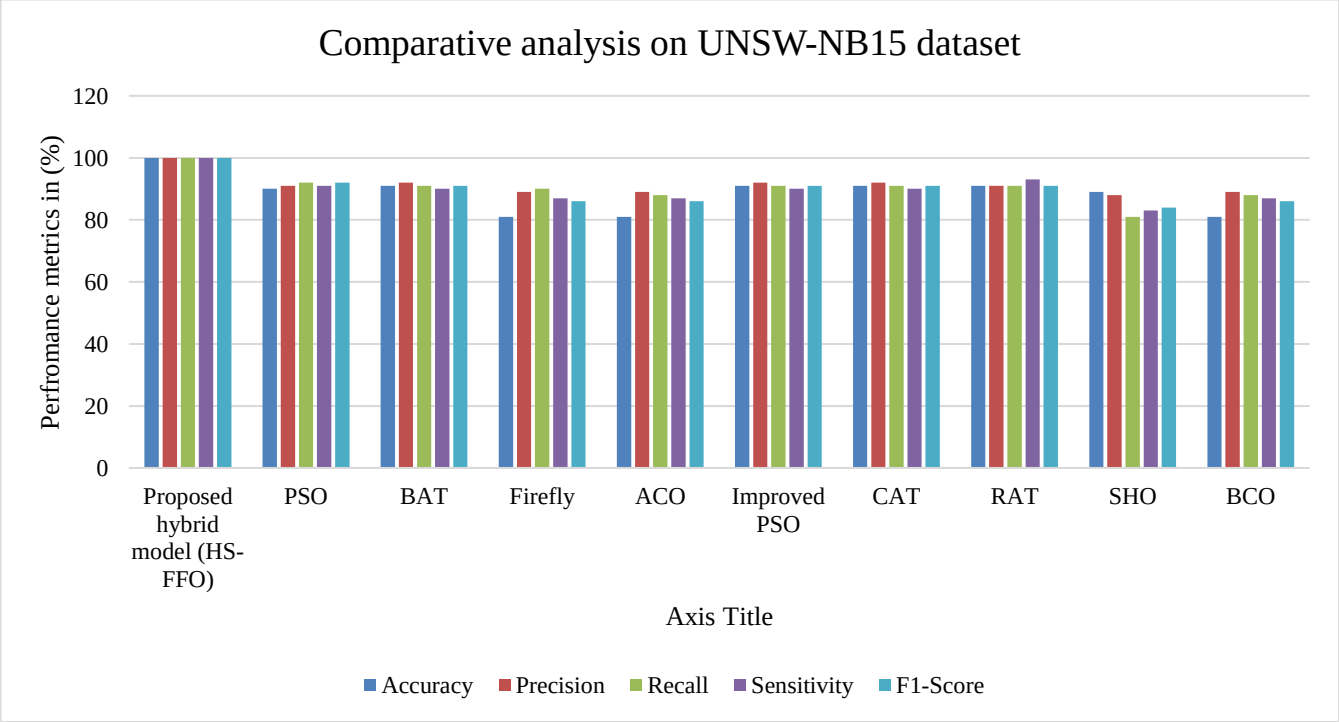

Figure 13: Proposed framework's Feature selection performance comparison with UNSW-NB15 benchmark dataset.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

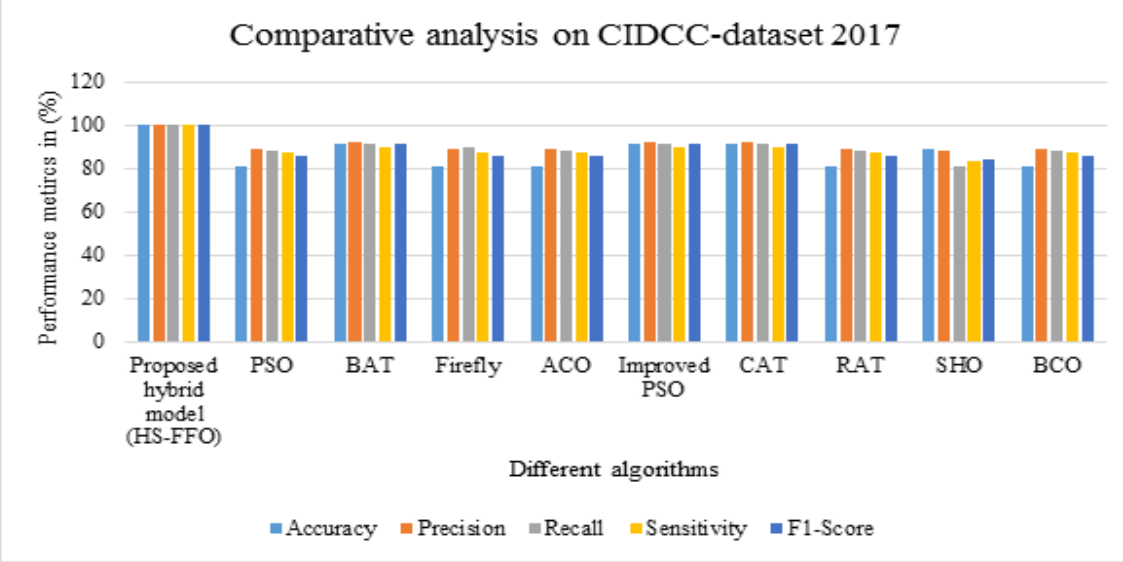

Figure 14: Proposed framework's Feature selection performance comparison using the CIDCC-dataset 2017

Figure 11 compares the performance characteristics of various hybrid deep learning algorithms for real-time datasets. Fig. 11 clearly shows that the suggested method performed better than the other learning models, with an accuracy of 99.95%. While other algorithms do even worse than the HS-FFO, it has a respectable accuracy of 99.97%. The SHO and Firefly algorithms' integration has demonstrated consistent performance on real-time test data, making it appropriate for use in real life scenarios. Figures 12 through 14 evaluate the effectiveness of deep learning models using the additional benchmark datasets. According to experimental findings, all hybrid models have accuracy ranging from 99.91% to 99.97%, precision from 99.92% to 99.98%, recall from 99.93% to 99.99%, and F-score from 99.93% to 99.99%. As a result, the proposed model outperforms existing deep learning models in terms of performance. The suggested HS-FFO model was discovered to have demonstrated stable outcomes with the actual data as well as with the various benchmarks. Many hybrid deep learning models, on the other hand, have demonstrated darker performance on real-world datasets but brighter over benchmarks. In light of the aforementioned findings, our suggested HS-FFO model has proven to be suitable for a real-time dataset employed in smart health care monitoring systems.

V. Conclusion

The hybrid spotted hyena-swarm optimization (HS-FFO) framework has been proposed in the paper. The integrated CPU linked up with esp8266 transceivers to gather the real-time datasets. Almost 60,000 pieces of data were gathered and put to use for training and testing. Furthermore, benchmark datasets including real-time environments have been used to compare the proposed technique to previous hybrid deep learning modelsThe findings demonstrate that the suggested algorithm has demonstrated consistent performance, as evidenced by its correctness of 99.95%, precision of 99.95%, recall of 99.93%, sensitivity of 99.94%, and f-score of 99.94% in the recommended model's feature selection achievement. The performance of the various hybrid deep learning approaches was exceptional when applied to benchmarks, but it significantly decreased when applied to real-time datasets. The results of the experimentation demonstrate that the suggested deep learning model performs better across a range of datasets and has enormous potential for use in cutting-edge health care monitoring solutions. Additionally, the proposed models must concentrate on recognising the uptick in attacks without sacrificing the reduction in energy utilization.

REFERENCES

[1]. Lim, S.Y.; Kiah, M.M.; Ang, T.F. Security Issues and Future Challenges of Cloud Service Authentication.*Polytech. Hung.* **2017**, *14*, 69–89.

[2]. Borylo, P.; Tornatore, M.; Jaglarz, P.; Shahriar, N.; Cholda, P.; Boutaba, R. Latency and energy-aware provisioning of network slices in cloud networks. *Comput. Commun.* **2020**, *157*, 1–19.

[3]. Carmo, M.; Dantas Silva, F.S.; Neto, A.V.; Corujo, D.; Aguiar, R. Network-Cloud Slicing Definitions for Wi-Fi Sharing Systems to Enhance 5G Ultra-Dense Network Capabilities. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 1–17.

[4]. Dang, L.M.; Piran, M.; Han, D.; Min, K.; Moon, H. A Survey on Internet of Things and Cloud Computing for healthcare. *Electronics* **2019**, *8*, 768.

[5]. Srinivasamurthy, S.; Liu, D. Survey on Cloud Computing Security. 2020. Available online:

https://www.semanticscholar.org/ (accessed on 19 July 2020).

[6]. Mathkunti, N. Cloud Computing: Security Issues. *Int. J. Comput. Commun. Eng.* **2014**, *3*, 259–263

[7]. Stefan, H.; Liakat, M. Cloud Computing Security Threats And Solutions. *J. Cloud Comput.* **2015**, *4*, 1.

[8]. Palumbo, F.; Aceto, G.; Botta, A.; Ciuonzo, D.; Persico, V.; Pescapé, A. Characterizing Cloud-to-user Latency as perceived by AWS and Azure Users spread over the Globe. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, 7–11 December 2019; pp. 1–6.

[9]. Hussein, N.H.; Khalid, A. A survey of Cloud Computing Security challenges and solutions. *Int. J. Comput.Sci. Inf. Secur.* **2017**, *1*, 52–56.

[10]. Le Duc, T.; Leiva, R.G.; Casari, P.; Östberg, P.O. Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey. *ACM Comput. Surv.* **2019**, *52*, 1–39.

[11]. Callara, M.; Wira, P. User Behavior Analysis with Machine Learning Techniques in Cloud Computing Architectures. In Proceedings of the 2018 International Conference on Applied Smart Systems, Médéa, Algeria, 24–25 November 2018; pp. 1–6.

[12]. Khan, A.N.; Fan, M.Y.; Malik, A.; Memon, R.A. Learning from Privacy Preserved Encrypted Data on Cloud Through Supervised and Unsupervised Machine Learning. In Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies, Sindh, Pakistan, 29–30 January 2019; pp. 1–5.

[13]. Khilar, P.; Vijay, C.; Rakesh, S. Trust-Based Access Control in Cloud Computing Using Machine Learning. In *Cloud Computing for Geospatial Big Data Analytics*; Das, H., Barik, R., Dubey, H., Roy, D., Eds.; Springer: Cham, Switzerland, 2019; Volume 49, pp. 55–79.

[14]. Bhamare, D.; Salman, T.; Samaka, M.; Erbad, A.; Jain, R. Feasibility of Supervised Machine Learning for Cloud Security. In Proceedings of the International Conference on Information Science and Security, Jaipur, India, 16–20 December 2016; pp. 1–5.

[15]. Li, C.; Song, M.; Zhang, M.; Luo, Y. Effective replica management for improving reliability and availability in edge-cloud computing environment. *J. Parallel Distrib. Comput.* **2020**, *143*, 107–128.

[16]. M. A. Elsayed and M. Zulkernine, "PredictDeep: Security Analytics as a Service for Anomaly Detection and Prediction," in IEEE Access, vol. 8, pp. 45184-45197, 2020, doi: 10.1109/ACCESS.2020.2977325.

[17]. D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Secure Computation Offloading in Blockchain Based IoT Networks With Deep Reinforcement Learning," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3192-3208, 1 Oct.-Dec. 2021, doi: 10.1109/TNSE.2021.3106956.

[18]. J. C. Kimmel, A. D. Mcdole, M. Abdelsalam, M. Gupta and R. Sandhu, "Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure," in IEEE Access, vol. 9, pp. 68066-68080, 2021, doi: 10.1109/ACCESS.2021.3077498.

[19]. G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon and D. Gan, "Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning," in IEEE Access, vol. 6, pp. 3491-3508, 2018, doi: 10.1109/ACCESS.2017.2782159.

[20]. S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya and R. Ranjan, "A Hybrid Deep Learning-Based Model for Anomaly Detection in Cloud Datacenter Networks," in IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 924-935, Sept. 2019, doi: 10.1109/TNSM.2019.2927886.

[21]. P. Abirami, S. Vijay Bhanu and T. K. Thivakaran, "Crypto-Deep Reinforcement Learning Based Cloud Security For Trusted Communication," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 1-10, doi: 10.1109/ICSSIT53264.2022.9716429.

[22]. Y. Tao, J. Qiu and S. Lai, "A Hybrid Cloud and Edge Control Strategy for Demand Responses Using Deep Reinforcement Learning and Transfer Learning," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 56-71, 1 Jan.-March 2022, doi: 10.1109/TCC.2021.3117580.

[23]. S. Hizal, Ü. ÇAVUŞOĞLU and D. AKGÜN, "A new Deep Learning Based Intrusion Detection System for Cloud Security," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-4, doi: 10.1109/HORA52670.2021.9461285.

[24]. C. Karri and M. S. R. Naidu, "Deep Learning Algorithms for Secure Robot Face Recognition in Cloud Environments," 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2020, pp. 1021-1028, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00154.

[25]. W. Wang, X. Du, D. Shan, R. Qin and N. Wang, "Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1634-1646, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3001017.

[26]. Q. Lio, J. Li, Y. Zhou and L. Liao "Using spotted hyena optimizer for training feedforward neural networks," *Cognitive Systems Research*, vol. 65, pp. 1-16, 2021.

[27]. K. Sood, A. Jain and A. Verma, "A hybrid task scheduling approach using firefly algorithm and gravitational search algorithm," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 2997-3002, doi: 10.1109/ICECDS.2017.8390005.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5851
***Article Received**: 12 October 2022 **Revised**: 07 November 2022 **Accepted**: 01 December 2022*

[28]. I. Butun, S. D. Morgera and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266-282, 2014.

[29]. I. Andrea, C. Chrysostomou and G. Hadjichristofi, "Internet of Things: security vulnerabilities and challenges," in *Proceedings of IEEE Symposium on Computers and Communication*, pp. 180-187, 2015

[30]. M. A. Patel and M. M. Patel, "Wormhole attack detection in wireless sensor network," in *Proceedings of International Conference on Inventive Research in Computing Applications, pp. 269-274, 2018*.

[31]. W. F. de Silva, R. Spolon, R. S. Lobato, A. M. Júnior and M. A. C. Humber, "Particle Swarm Algorithm Parameters Analysis for Scheduling Virtual Machines in Cloud Computing," 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 2020, pp. 1-6, doi: 10.23919/CISTI49556.2020.9141021.

[32]. X. Li, N. Luo, D. Tang, Z. Zheng, Z. Qin and X. Gao, "BA-BNN: Detect LDoS Attacks in SDN Based on Bat Algorithm and BP Neural Network," 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 300-307, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00050.

[33]. Y. Liu, X. Wang, M. Cheng, J. Wang and Y. Zhang, "An Efficient Task Offloading Strategy in Cloud-Edge Computing Under Deadline Constraints," 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2020, pp. 661-667, doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00085.

[34]. Z. -G. Chen et al., "Multiobjective Cloud Workflow Scheduling: A Multiple Populations Ant Colony System Approach," in IEEE Transactions on Cybernetics, vol. 49, no. 8, pp. 2912-2926, Aug. 2019, doi: 10.1109/TCYB.2018.2832640.

[35]. Z. Zhou, F. Li, J. H. Abawajy and C. Gao, "Improved PSO Algorithm Integrated With Opposition-Based Learning and Tentative Perception in Networked Data Centres," in IEEE Access, vol. 8, pp. 55872-55880, 2020, doi: 10.1109/ACCESS.2020.2981972.

[36]. R. Pushpa and M. Siddappa, "Adaptive Hybrid Optimization Based Virtual Machine Placement in Cloud Computing," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 1-9, doi: 10.1109/ICSSIT53264.2022.9716298.

[37]. Z. Xu, X. Liang, M. He and H. Chen, "Multiple Adaptive Strategies-based Rat Swarm Optimizer," 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), 2021, pp. 159-163, doi: 10.1109/CCIS53392.2021.9754632.

[38]. Kaur, Amandeep & Dhiman, Gaurav & Garg, Meenakshi. (2021). Task Scheduling in Cloud Computing Using Spotted Hyena Optimizer. 10.4018/978-1-7998-5040-3.ch009.

[39]. Ulah, Arif & Mohd Nawi, Nazri & Uddin, Jamal & Baseer, Samad. (2019). Artificial Bee Colony algorithm used for load balancing in cloud computing: Review. IAES International Journal of Artificial Intelligence (IJ-AI). 8. 156-167. 10.11591/ijai.v8.i2.pp156-167.