International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

GBJOF: Gradient Boosting Integrated with Jaya Algorithm to Optimize the Features in Malware Analysis

D. Sandhya Rani[1], G Bala Krishna[2], M Radha[3], Sirisha K L S[4], V Pradeep Kumar[5], Meduri Anupama[6]
[1]Professor,
Department of CSE,
CVR College of Engineering, Ibrahimpatnam, Rangareddy, India
d.sandhyarani@cvr.ac.in
[2]Professor,
Department of CSE,
CVR College of Engineering, Ibrahimpatnam, Rangareddy, India
g.balakrishna@cvr.ac.in
[3]Associate Professor,
Department of CSE,
CVR College of Engineering, Ibrahimpatnam, Rangareddy, India
Marepalli.radha@gmail.com
[4]Assistant Professor,
Department of CSE,
Keshav Memorial Institute of Technology, Narayanguda, Hyderabad, India
klssirisha@kmit.in
[5]Associate Professor,
Department of CSE,
B V Raju Institute of Technology, Narsapur, Medak, India
pradeepkumar.v@bvrit.ac.in
[6]Associate Professor,
Department of CSE,
Maturi Venkata Subba Rao(MVSR) Engineering College, Hyderabad, India
Anupama_cse@mvsrec.edu.in

Abstract— Malware analysis is used to identify suspicious file transferring in the network. It can be identified efficiently by using the reverse engineering hybrid approach. Implementing a hybrid approach depends on the feature selection because the dataset contains static and dynamic parameters. The given dataset contains 85 attributes with 10 different class labels. Since it has high dimensional and multi-classification data, existing approaches of ML could be more efficient in reducing the features. The model combines the enhanced JAYA genetic algorithm with a gradient boosting technique to identify the efficiency and a smaller number of features. Many existing approaches for feature selection either implement correlation analysis or wrapper techniques. The major disadvantages of these issues are that they are facing fitting problems with a very small number of features. With the Usage of the genetic approach, this paper has achieved 95% accuracy with 12 features, approximately 7% greater than ML approaches.

Keywords- Genetic Approaches, JAYA algorithm, Feature Selection, Accuracy, Dimensionality Reduction.

I. INTRODUCTION

Feature selection is a critical step in machine learning that involves selecting a subset of features (or variables) that is most relevant for predicting the target variable. The operation of feature selection may be automated using genetic algorithms, a sort of optimization approach [26]. Here are some needs for feature selection using genetic algorithms: Improved predictive accuracy: The accuracy of a machine learning method's prediction can be increased by utilizing genetic algorithms to choose the most pertinent features to the current challenge. By reducing the number of features, the model becomes less complex and can perform better. Reduced computational complexity: Feature selection can also help reduce the computational complexity of the model by eliminating irrelevant or redundant features [29]. This approach can result in faster training and inference times, making the model more

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

practical for real-world applications. Better interpretability: The resulting model may be easier to interpret and understand with fewer features. This model can be important for applications where knowing which features drive the model's predictions is important. Robustness to noisy data: Feature selection using genetic algorithms can help improve the robustness of the model to noisy data [25]. By selecting only the most relevant features, the model can be less susceptible to overfitting and perform better on new, unseen data. The categorization of feature selection techniques is presented in Figure 1.

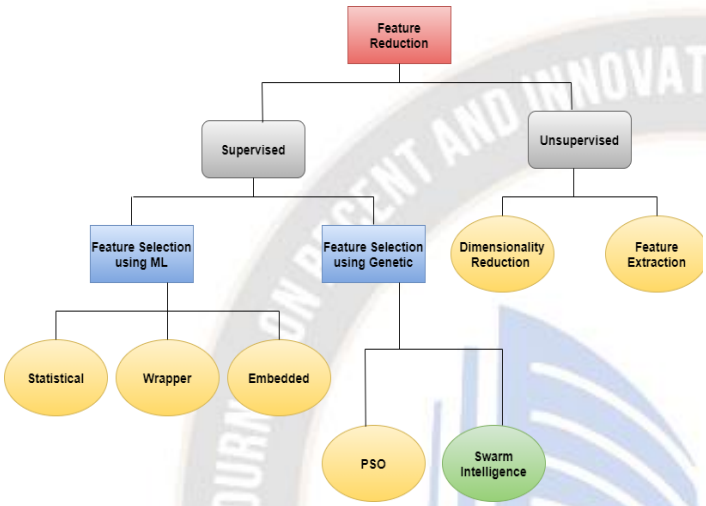

Figure 1: Categorization of Feature Reduction Techniques

There are, however, some significant distinctions between the selection and extraction of features. A fraction of the original characteristics are chosen through feature selection depending on their relevance to the target attribute [11]. Many techniques can do this, including correlation, mutual information, and statistical tests. Feature selection aims to find the most informative characteristics of the current situation and eliminate the unnecessary or redundant ones. Feature selection does not modify the original features but instead chooses a subset of them to be used in a machine-learning model.

On the other hand, dimensionality reduction is a technique that transforms the original information while fitting as many of the original characteristics into a lower-dimensional phase [12]. PCA and t-SNE are two techniques that may be used to reduce the number of dimensions. Dimensionality reduction aims to reduce the number of features while minimizing information loss. When there are more characteristics than observations—a situation known as the "curse of dimensionality—feature reduction can solve the problem [20]. The main difference is that FS considers the subset of real features, whereas DR transfers the real features into the low-dimensionality phase.

1.1. Embedded Features Selection in Machine Learning

A feature selection approach used in machine learning termed embedded feature selection includes inserting feature selection within the model training procedure. In other words, the model-training process itself incorporates the feature selection method. Embedded feature selection techniques typically use a model-specific criterion to evaluate the importance of each feature during the training process [13]. The importance of each feature is then used to rank the features, and a subset of the most important features is selected for the final model. Examples of embedded feature selection techniques include LASSO (least absolute shrinkage and selection operator) and elastic net regularization. Embedded feature selection has some advantages over other feature selection techniques. One of the key advantages is that it can identify the features most relevant to the specific model being used. This technique is used because the feature selection process is interwoven into the training phase, allowing the design to acquire the most crucial characteristics for the job [19]. Additionally, embedded feature selection can be computationally efficient, as it does not require a separate feature selection step outside the model training process.

1.2. Wrapper for Feature Selection Using Machine Learning

Wrapper FS is a feature selection technique in machine learning that involves selecting features based on their performance on a specific machine learning algorithm. In other words, the feature selection process is wrapped around the machine learning algorithm being used. Wrapper feature selection techniques typically involve a search algorithm that evaluates different subsets of features and measures their performance using a machine learning algorithm [28]. The search algorithm can be exhaustive, evaluating every possible subset of features, or it can be heuristic, evaluating a smaller subset of features based on some criteria. RFE operates by repeatedly deleting the least significant features and reassessing the model's performance until the required amount of features is obtained. At each iteration, the model is trained on the remaining features, and the importance of each feature is evaluated using a performance metric such as accuracy or mean squared error. Wrapper feature selection has some advantages over other feature selection techniques [23]. One of the key advantages is that it can identify the most relevant features for a specific machine learning algorithm because the feature selection process is integrated with the machine learning algorithm, which allows the wrapper technique to capture any interactions between the features and the algorithm.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

1.3. Statistics in Feature Selection Using Machine Learning

Statistics-based FS is a feature selection technique in machine learning that involves selecting features based on their statistical properties. This approach involves analysing the statistical properties of each feature in the dataset and selecting the most informative features for the task at hand [14]. One common example of a statistics-based feature selection technique is correlation analysis. Obtaining the correlation coefficient between each characteristic and its specific value is a step in correlation analysis. High correlation coefficient features are considered more relevant and more likely to be chosen for the final model [15]. Additionally, features with a high correlation coefficient between them may be redundant, and one of them can be removed to reduce dimensionality. Another example of a statistics-based feature selection technique is mutual information [30]. Mutual information measures the dependence between two random variables and is used to evaluate the relevance of each feature to the target scope. Mutually informative score features are considered more relevant and, therefore, more likely to be chosen for the final version. Statistics-based feature selection has some advantages over other feature selection techniques [16]. One of the key advantages is that it is simple and computationally efficient, as it does not require training a model on the data

II. LITERATURE SURVEY:

Ihab Shahadat et al. [1] Have proposed a machine-learning technique against malware-affected Android phones. The dataset contains 1k files of malware and benign files in a ratio of 8:2. Nine variants of malware families were identified using the proposed method. Feature extraction values are retrieved from the heuristic strategy. This study used the random forest method based on a tree-based structure. Subsequently, cross-validation was performed to achieve classification data with high efficiency. Four validation techniques were used for the evaluation metrics. Six machine-learning methods were chosen to compare the proposed method with different approaches. Here, the evaluation results are performed on two classifiers: binary & multi-classifications.

R. Srinivasan et al. [2] Focused on an issue regarding malware attacks which are highly affected Android phones. Malware attacks can be identified in different formats using various approaches. The working process of the proposed approach involves taking the APK files followed by the intermediate files, which are split and generated according to the vectors of the extracted features. On the other hand, training data is transferred to the database, and classification parallel machine learning models are also considered for malicious and benign applications. Further working on machine learning approaches, the data is Interlaken for pre-processing, and the extraction and comparison of features are performed with these data. Now, these data are trained and tested, their classification approach is derived, and the results are retrieved. An accuracy of 90% was achieved by using this approach.

Muhammad Shoaib Akhtar et al [3] focused on malware attacks performed on the Android phone where information can be misused. In general, multiple attacks are performed on social media through different platforms, and malware is one of the major types of attacks. The method involves the training and detection phases. In the training phase, the sample set was analysed for feature extraction. These features are generated based on new data from the training detectors. In the detection phase, the samples were unknown, and the analysis was considered a sample. Subsequently, the features are extracted to generate new feature vectors. Therefore, these values are connected to the training detectors where the data travels to detect malware and differentiate between malware and benign. The dataset was collected from the Canadian Institute, where almost 17k data were available, with 51 varieties of malware. Six different approaches were chosen to evaluate the work, in which 89% accuracy was achieved with a decision tree.

Olaniyi Abiodun Ayeni [4] found an Android phone issue affecting users' data and security. Using machine learning techniques, malware can be detected, and, to some extent, the data can be protected. Here, three distinct machine-learning methods were used. The main objective of designing this method is to provide a proper framework for security using the supervised ml method. The data were collected from the Kaggle database with almost 1lakh rows and 35 columns, with 21 different malware features. The malware is generated in a two-phase signature and behaviour. The authors explained four different types of malware. The work starts with data intake, followed by transformation and machine learning training models. Random forest attained an elevated accuracy of 89.90%.

Tao Feng et al. [5] Focused on identifying malware on Android phones which is highly influenced the user's data for corruption etc. Malware can generally be identified in three ways: objective future and machine-learning methods. The objective is to categorize detection and similarities. In the future, the extraction of PE features should be performed. In machine-learning approaches, any algorithm can be performed when the free-trained model is ready. The proposed framework takes the data and directly transfers it to the training and testing phase, where the tested dataset is created from the training and testing phases. Here, the retrieved data are transformative, and some related features are collected; therefore, a machine-learning model is developed. Twelve machine learning algorithms were chosen to evaluate the proposed method. Among the random

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

forests, SGD, extra-tree classifier, and GNB had the highest accuracy of 100%.

In [6], Nighat Usman et al. Proposed a revolutionary cyber security method based on machine learning and big data forensics to detect rogue IP addresses, or Internet Protocol (IP), before the malware may cause harm. Reputation systems rely on multiple antiviral, ML, and blocked IPs. Due to their high administration costs, these techniques do not effectively classify zero-day attacks and must be regularly updated. Machine learning algorithms like Support Vector Machine, DT, MBK, and Naive Bayes are applied to the dataset obtained from Cuckoo. DT performs best when classifying the unidentified malware samples. There were 3 examples in the analysis. The author suggests a real-time reporting framework integrating many analysis tools and methodologies into a unified architecture. The system integrates manual, stationary, and dynamic analysis to automatically report on the actions taken by the zero-day attack. The suggested approach has the potential to provide fruitful outcomes. The researchers propose that the rules will set out an alarm when the structure detects any IP address with a poor reputation. The researchers will then be able to offer upgraded policies that are simple to embed in firewall or anti-virus engines.

In [7], Firoz Khan et al. Used machine learning to study a digital analysis of a DNA engine for ransomware detection. The researchers propose DNAact-Ran, a machine-learning technique that examines ransomware's digital DNA to identify it. Using the active learning approach, it generates the digital DNA Order for a few properties and classifies each instance as either good ware or ransomware. The authors compared the DNA act-Ran classifier against Naive Bayes, Decisions Stump, and AdaBoost. First, DNAact-Ran selects the most significant features of the pre-processed data using the MOGWO and Binary Cuckoo search techniques. A digital DNA sequencing technology that employs Machine Learning to foresee and detect ransomware has been developed by researchers. The test proves active learning algorithms are more efficient at swiftly identifying ransomware. The analysis involved 1524 records.

In [8], Jinsoo Hwang et al. Explored machine learning and dynamic analysis methods for two-stage ransomware detection. They are detecting ransomware using machine learning and demonstrating its applicability to different malware kinds. By creating Markov systems for both malicious and benign programs, the researchers first concentrate on the sequential properties of Windows APIs. The researchers recommend a mixed, two-stage detection technique with a significant emphasis on reducing the false negative rate of errors and a little focus on reducing false favourable error rates. The characteristics include ransomware note strings, suspicious file operations, API call patterns or frequency, system keys, file extensions, entropy adjustments to files, Master File Table alterations, etc. The authors will investigate a detection technique utilizing a Windows API call sequence, the typical feature selection in a Markov model, and other machine-learning detection techniques. The researchers tested the two-stage mixed detection approach. The authors have 1139 normal samples and 1909 samples of ransomware. To get the API sequences called during the API Attach Process To Job Object, the researchers ran the samples above inside Cuckoo SandBox. Table 1 presents the overall merit analysis of the previous works.

Table 1: Analysis of the Existing Systems

Author	**Algorithm**	**Merits**	**Demerits**	**Accuracy**
Ihab Shahadat et al	Decision Tree	I can identify almost nine malware families.	If the dataset is unbalanced, working will be tough.	78.2%
R. Srinivasan et al.	Genetic approach	The performance is high.	The data set needs to be larger.	90%
Muhammad Shoaib Akhtar et al.	SVM and NN	The proposed can identify 51 varieties of malware.	The tested data percentage needs to be increased.	89%
Olaniyi Abiodun Ayeni	Random Forest	The security permission is perfectly utilized.	The performances with the supervised method have high accuracy.	89.90%
Tao Feng et al.	RF, SGD, Extra trees and GNB	Among the 7 approaches, 4 have achieved the highest accuracy.	The data set is significantly less.	100%
Nighat Usman	IP Rep. - FDA	Multi-problem identification, the cuckoo binary box, and a decision	More defined detection of IP should be processed and can be	93.5%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

		tree are used for classification.	embedded with firewall mechanisms.	
Firoz Khan	DNAact-Ran	MOGWO and BCS identify the relevant features, use active learning, and reduce costs.	Only partially proves that active learner involvement will impact the system results.	87.9%
Jinsoo Hwang	Markov model and Random Forest model	Min-max scaling is employed and evaluated using 10-fold cross-validation.	Some attacks can't be identified, not adaptable	87.44%

III. PROPOSED METHODOLOGY:

The proposed model starts with pre-processing the data. The dataset contains 50000 records with 85 attributes. Many attributes are categorical data, so the proposed data initially transforms the data into numerical after checking the null validation rules. In the second step, the model has to reduce the dimensionality of the features. Since it is supervised data, it applies genetic approaches with a gradient boosting algorithm as shown in Figure 2.

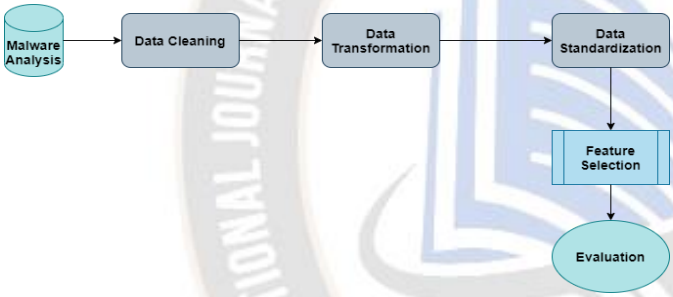

Figure 2: Block Diagram for Feature Selection in Embedded Jaya Approach

Genetic algorithms can be useful in feature selection because they can efficiently search through a large number of potential feature combinations to identify the exact subset of features that maximizes the performance of a machine learning model. Feature selection is an essential step in the machine learning pipeline that involves choosing a subdivision of features which is the most related tasks hand. This model can decrease data size, improve interpretability, and reduce overfitting. However, searching through all possible subsets of features can be computationally expensive and time-consuming, especially for high-dimensional data. This is where genetic algorithms come in - they can efficiently search through many possible feature combinations and find a good subset of features that maximizes the model's performance.

Similarly, in a genetic algorithm for feature selection, a population of potential feature subsets has evolved over multiple generations through mutation, crossover, and selection. Only the fittest individuals are retained and used to generate the next generation. Genetic algorithms can effectively and efficiently select features, especially for high-dimensional data where exhaustively searching through all possible feature subsets is not feasible.

3.1. Swarm intelligence algorithms

A class of computing algorithms known as swarm intelligence algorithms is modelled after the group behaviour of social creatures like ants, bees, and birds. These algorithms use animal behaviour to address challenging optimization and search issues. The basic idea behind swarm intelligence algorithms is that a group of individuals (agents) work together to find a solution to a problem. Each agent in the swarm interacts with other agents and the environment to find the best possible solution. The classification of the swarm intelligence algorithm is presented in Figure 3.

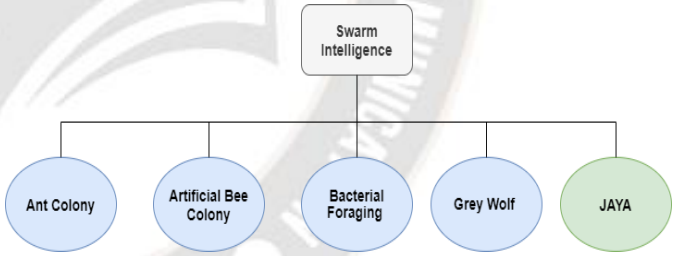

Figure 3: Categories of Swarm Genetic Approaches

3.2. Types of swarm intelligence algorithms

Swarm intelligence algorithms come in various forms, each drawing inspiration from the behaviour of various social animals. A few of the most typical varieties are listed below:

- ACO or Ant Colony Optimization ACO algorithms, are used to resolve optimization issues. The foraging behaviour of ants influenced these algorithms. The system determines the shortest route between two places. The algorithm mimics the pheromone trail made by ants.
- Artificial Bee Colony (ABC): ABC algorithms are used to resolve optimization issues and are modelled after the behaviour of honey bees. The algorithm

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

mimics how bees forage by looking for nectar sites and coordinating with one another to identify the optimum source.

- Bacterial Foraging Optimization (BFO): Inspired by the behaviour of bacterial colonies, BFO algorithms are used to solve optimization problems. The algorithm simulates the foraging behaviour of bacteria, searching for nutrients and adapting to their environment.
- Grey Wolf Optimizer (GWO): Inspired by the social hierarchy of wolf packs, GWO algorithms are used to optimize a function. The algorithm simulates the hunting behaviour of wolf packs, where each wolf adjusts its position based on the position of the alpha, beta, and delta wolves in the pack. These algorithms have been successfully applied in various fields, including optimization, data mining, machine learning, robotics, and image processing.
- Jaya Optimization Algorithm (JOA) is motivated by interpersonal collaboration. The algorithm uses a simple mechanism of improving solutions through individual cooperation to solve optimization problems.

The proposed research enhances the JOA algorithm's fundamental steps as follows:

Initialization: Establish a random starting population of n possibilities in the solution space. Determine fitness: Analyze the population fitness of each solution.

A population update Update the number using the given equation for each iteration t:

For each pair of solutions i and j in the population:

a. If the fitness of i is better than the fitness of j, update the jth solution using the below equation:

$$A(b,t+1) = A(b,t) + r * (A(i,t) - A(b,t))$$

b. If the fitness of b is better than the fitness of c, update the cth solution using the following equation:

$$A(c,t+1) = A(c,t) + r * (A(b,t) - A(c,t))$$

c. If the fitness of c is equal to the fitness of b, update the cth and bth solutions using the following equation:

$$A(c,t+1) = A(c,t) + r * (A(b,t) - A(c,t))$$

$$A(b,t+1) = A(b,t) + r * (A(c,t) – A(b,t))$$

where r is a random number between 0 and 1.

Boundary handling: Check if the new positions of the solutions are within the surrounding searching space. If a solution is out of bounds, randomly reposition it within the bounds.

Update the parameter: Update the parameter r using the following equation:

$$r(t+1) = r(t) * alpha$$

where alpha is a parameter that controls the rate of convergence.

Up until a stopping requirement is fulfilled, repeat steps 2 through 5.

The mathematical working of the JAYA algorithm is discussed in the below section.

Let us consider there are two variables X, Y

Population size be 5

So the range will be [-5, 5]

Upper limit (U) = 5

Lower limit (L) = -5

Since there are two variables, let the fitness function by minimizing

$$f(X, Y)= X^2 – XY + Y^2 + 2X + 4Y + 3$$

At the 0^{th} iteration, the values of X and Y are calculated using the below formula.

$$X = L + rand. * (U – L)$$

Where L = Lower limit, U = Upper limit, and. = random number in [0,1]

By using the above formula and assuming a random number, the model got the values as shown in Table 2

Table 2: 0^{th} iteration values of the particles

Particle Number	X	Y	f (X, Y)
1.	-4.58	3.99	
2.	-0.02	4.17	
3.	-0.80	4.32	
4.	1.73	2.65	
5.	1.00	-2.60	

Let us now calculate minimizing fitness function for each particle using the function and presents the computed values in Table 3.

$$f(X, Y)= X^2 – XY + Y^2 + 2X + 4Y + 3$$

1^{st} particle when X = -4.58, Y = 3.99

$$f(-4.58, 3.99) = (-4.58)^2 – (-4.58)(3.99) + (3.99)^2 + 2(-4.58) + 4(3.99) + 3 = 64.97$$

2^{nd} particle when X = -0.02, Y = 4.17

$$f(-0.02, 4.17) = (-0.02)^2 – (-0.02)(4.17) + (4.17)^2 + 2(-0.02) + 4(4.17) + 3 = 37.11$$

3^{rd} particle when X = -0.80 and Y = 4.32

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

$f(-0.80, 4.32) = (-0.80)^2 – (-0.80)(4.32) + (4.32)^2 + 2(-0.80) + 4(4.32) + 3 = 41.43$

4^{th} particle when X = 1.73 and Y = 2.65

$f(1.73, 2.65) = (1.73)^2 – (1.73)(2.65) + (2.65)^2 + 2(1.73) + 4(2.65) + 3 = 22.49$

5^{th} particle when X = 1, Y = -2.60

$f(1, -2.60) = 1^2 – (1)(-2.60) + (-2.60)^2 + 2(1) + 4(-2.60) + 3 = 4.96$

Table 3: Fitness values of each particle

Particle Number	X	Y	f (X, Y)
1.	-4.58	3.99	64.97
2.	-0.02	4.17	37.11
3.	-0.80	4.32	41.43
4.	1.73	2.65	22.49
5.	1.00	-2.60	4.96

Now the proposed model performs the 1^{st} iteration to update the fitness values to identify the essential particles of the given objective function.

Let $r_1 = 0.34$ and $r_{2 = 0.65}$

X_best (minimum f(X , Y) = particle 5 has minimum [1.00 -2.60]

X_worst (maximum f(X, Y) = particle 1 has maximum [-4.58 3.99]

$X_{new} = X_{j,k} + r_1(X\_best - |X_{j,k}|) – r_2(X\_worst - |X_{j,k}|)$

After finding the new solutions, we find the fitness function using the formula.

Now perform greedy operation among the new f(X, Y) and old f(X, Y)

If f(X_new , Y_new) < f(X, Y)

Then update the X and Y values; otherwise, the previous are left.

1^{st} particle

$X = -4.58 + 0.34(1.00 -| -4.58|) – 0.65 (-4.58 - |-4.58|) = 3.27$

$Y = 3.99 + 0.34(-2.60 -| 3.99|) – 0.65 (3.99 - |3.99|) = 1.74$

$X_{new} = [3.27 \quad 1.74]$

$\rightarrow f(3.27, 1.74)= (3.27)^2 – (3.27)(1.74) + (1.74)^2 + 2(3.27) + 4(1.74) + 3 = 24.53$

f (X_new , Y_new) < f(X, Y)

24.23 < 64.97 True, so update the X, Y and f (X, Y)

As shown in the above section, the JAYA algorithm updates the values of each particle and presents the new information in Table 4 after the 1^{st} iteration is completed.

Table 4: 1^{st} iteration value

Particle number	X	Y	f (X, Y)
1.	3.27	1.74	24.53
2.	3.30	1.98	25.79
3.	2.62	2.18	22.86
4.	1.73	2.65	22.49
5.	1.00	-1.73	0.81

Many optimization issues, including engineering design, image processing, and machine learning, have been solved using the JOA method. The method is simple to construct and utilize since it only needs a small number of parameters. The workflow of the JAYA algorithm is presented in Figure 4.

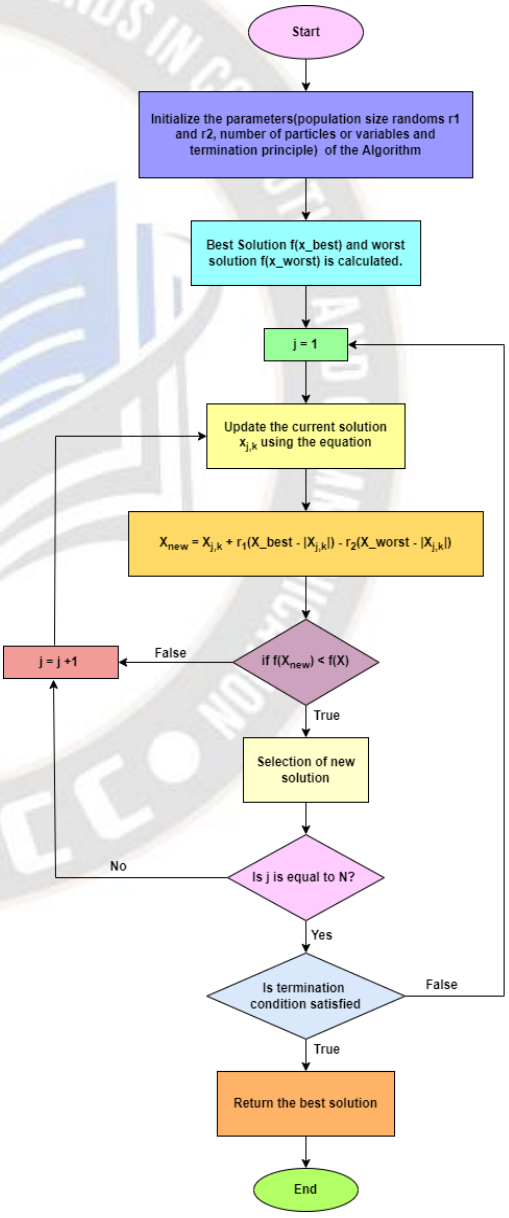

Figure 4: Working of the JAYA algorithm to update particle values

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

The JAYA algorithm is combined with a gradient-boosting algorithm to define the objective function. Most genetic algorithms define their objective function based on the KNN, but KNN is the laziest algorithm that slows the process with a huge amount of data. It also takes more iterations to compare every possible value in the dataset. The proposed dataset contains 50000 records, so to simplify the process, the model used gradient boosting instead of KNN. The model builds a tree sequentially by recovering the error parts of the previous step. Sequential learning slows the learning process, but it has the advantage of correcting the parameters then itself without waiting till the end. The key parameter of this algorithm is the learning rate. The proposed model identified the learning rate using the JAYA algorithm and made the model learn complex features at a more significant rate.

IV. RESULTS & DISCUSSION:

The proposed model transforms the data using label encoding. Label encoding assigns the 0 to n-1 values to every column, where n is the number of unique values. This data transformation process is expressed in table 5.

Table 5: Transformed Data After Pre-processing

Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Active Std
217	1	54819	37	443	6	1497414172	194	0.0
221	1	51023	39	443	6	1497414172	5	0.0
279	1	39805	62	443	6	1497414178	199542	0.0
279	1	39805	62	443	6	1497414178	254	0.0
212	11	443	1	36040	6	1497414179	2164751	0.0
	-	-	-	---	-		---	
128	43	443	1	34379	6	1497523080	96856	0.0
133	1	49669	111	80	6	1497523076	5631270	0.0
129	1	34380	111	443	6	1497522891	5717407	0.0
129	1	34380	111	443	6	1497523081	241	0.0
132	1	34392	111	443	6	1497522896	177084	0.0

Table 6 explains the correlation analysis between all the attributes present in the dataset. In the below calculations, most of the values are negative, which means that most of the attributes in the dataset are uncorrelated, i.e., means most of the features are retained as it is.

Table 6: Correlation Analysis over Attributes in the Dataset

Flow ID	Source IP	Source Port	Destination IP	Destination Port
Flow ID	1	0.02835	-0.097102	-0.055444
Source IP	0.02835	1	-0.785815	-0.372254
Source Port	-0.097102	-0.785815	1	0.477774
Destination IP	-0.055444	-0.372254	0.477774	1
Destination Port	0.031008	0.752186	-0.861457	-0.558114

Table 7 presents the selected features using the proposed model. It has got 12 features as essential out of 85.

Table 7: Feature Selected using the Proposed Model

Feature Number	Feature Name	Description
6	Protocol	Type of the protocol implemented in sharing the data
13	Forward packet length	Length of the packets transferring from source to destination (Data)
16	Backward packet length	Length of the packets transferring from destination to source (Acknowledgements)
21	Flow Packets	The nature of the packet flows
76	Push Flags	Priority Flag values to decide the order of flow
77	Mean Packet Length	The average packet length in both directions
78	Segmentation Size	The size of the packet partition
79	Active Hubs	Number of receiving active ends

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

80	Flag Count	Number of Flag values that are initialized during the packet transmission
81	Source Port	Source IP address
82	Destination Port	Destination IP address
83	Flow duration	Time spent in sending the packet

Table 8 presents the features selected by different approaches and the accuracy rate based on the selection. Out of the approaches utilized, the proposed algorithm, i.e., the JAYA algorithm, when defining the objective function using the gradient boosting approach, has fewer features with high accuracy.

Table 8: Feature Selection Using Different Approaches

S.No	Name	Number of Features Selected	Accuracy
1	Correlation	75	60.95
2	RFE	45	78.12
3	Embedded	40	80.40
4	PSO	32	89.93
5	Proposed	12	95

Figure 5 represents the visualization of existing approaches with the proposed one based on feature selection and accuracy. Y-axis presents the measuring values like the number of features in blue and accuracy in orange. X-axis marks the algorithms implemented in the comparison analysis.

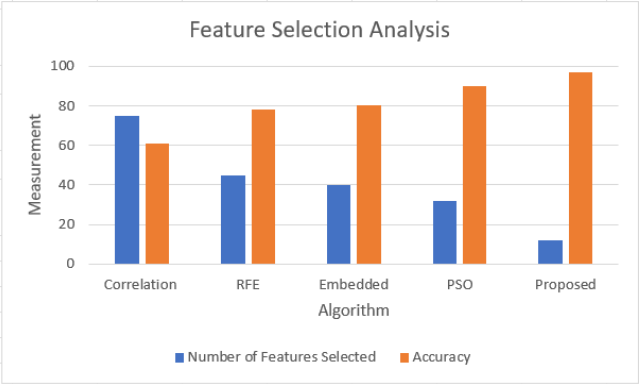

Figure 5: Feature Selection Analysis Report

Figure 6 represents the accuracy of different mechanisms studied in the literature survey and compares it with the proposed model to prove the system's efficiency. One of the existing approaches has the 100% (Random Forest integrated approach), but 100% accuracy is considered an overfitting problem. While the other approaches have less accuracy when compared to the proposed model. X-axis denotes algorithms used in the literature survey, and Y-axis denotes accuracy measurement.

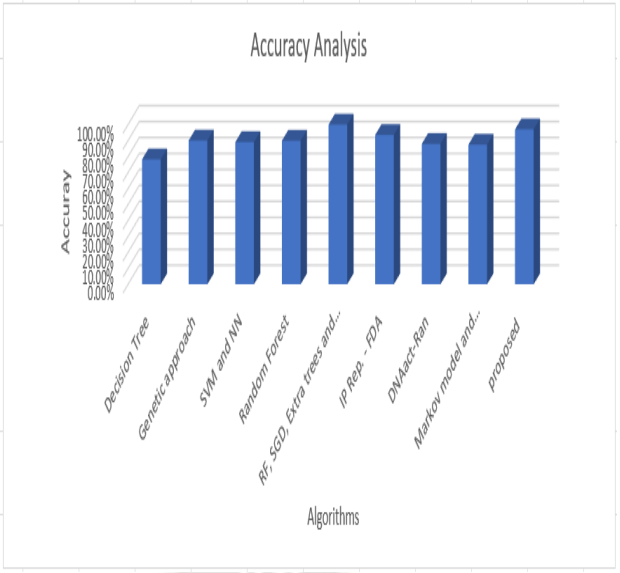

Figure 6: Analysis based on the accuracy

V. CONCLUSION:

Feature selection is the essential step in achieving the efficient accuracy of any ML model. The major focus of this paper is efficient feature selection using the combination of genetic and ML approaches because these approaches as an individual have their drawbacks. ML sometimes faces the problem of inefficient features because of the inappropriate parameters assigned to the techniques. The genetic algorithms may update particles wrongly because of choosing the inappropriate fitness function to reach the objective values. So, combining these techniques helps the model to adjust the particle values based on accuracy or error rate as the objective function. Since the goal is fixed, random assumptions are not required. The range of domain space is limited to several features. In future work, the model tunes the parameters of ML and integrates the genetic approach so that it can be generalized to similar datasets because malware analysis has different types of datasets available in the market. Generalization of the model helps the automation process to reduce the burden of developing the system from scratch.

REFERENCES

[1] Shhadat, I., Bataineh, B., Hayajneh, A., & Al-Sharif, Z. A. (2020). The Use of Machine Learning Techniques to Advance the Detection and Classification of Unknown Malware. Procedia Computer Science, 170, 917–922. https://doi.org/10.1016/j.procs.2020.03.110

[2] Srinivasan, R., Karpagam, S., Kavitha, M., & Kavitha, R. (2022). An Analysis of Machine Learning-Based Android Malware Detection Approaches. Journal of Physics: Conference Series, 2325(1). https://doi.org/10.1088/1742-6596/2325/1/012058

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

[3] Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry, 14(11). https://doi.org/10.3390/sym14112304

[4] Ayeni, O. A. (2022). A Supervised Machine Learning Algorithm for Detecting Malware. Journal of Internet Technology and Secured Transactions, 10(1), 764–769. https://doi.org/10.20533/jitst.2046.3723.2022.0094

[5] Akhtar, M. S., & Feng, T. (2023). Evaluation of Machine Learning Algorithms for Malware Detection. Sensors, 23(2). https://doi.org/10.3390/s23020946

[6] Usman, Nighat, Saeeda Usman, Fazlullah Khan, Mian Ahmad Jan, Ahthasham Sajid, Mamoun Alazab, and Paul Watters. "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics." Future Generation Computer Systems 118 (2021): 124-141.

[7] Khan, Firoz, Cornelius Ncube, Lakshmana Kumar Ramasamy, Seifedine Kadry, and Yunyoung Nam. "A digital DNA sequencing engine for ransomware detection using machine learning." IEEE Access 8 (2020): 119710-119719.

[8] Hwang, J., Kim, J., Lee, S., & Kim, K. (2020). Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques. Wireless Personal Communications. doi:10.1007/s11277-020-07166-9

[9] Mahindru, A., Sangal, A.L. MLDroid—a framework for Android malware detection using machine learning techniques. *Neural Comput & Applic* **33**, 5183–5240 (2021). https://doi.org/10.1007/s00521-020-05309-4

[10] Daoudi, N., Samhi, J., Kabore, A.K., Allix, K., Bissyandé, T.F., Klein, J. (2021). DEXRAY: A Simple, yet Effective Deep Learning Approach to Android Malware Detection Based on Image Representation of Bytecode. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (eds) Deployable Machine Learning for Security Defense. MLHat 2021. Communications in Computer and Information Science, vol 1482. Springer, Cham. https://doi.org/10.1007/978-3-030-87839-9_4

[11] Akhtar, M. S., & Feng, T. (2023). Evaluation of Machine Learning Algorithms for Malware Detection. Sensors, 23(2), 946. MDPI AG. Retrieved from http://dx.doi.org/10.3390/s23020946

[12] Shaukat, K., Luo, S., & Varadharajan, V. (2023). A novel deep learning-based approach for malware detection. In Engineering Applications of Artificial Intelligence (Vol. 122, p. 106030). Elsevier BV. https://doi.org/10.1016/j.engappai.2023.106030

[13] Dhalaria, M., Gandotra, E., & Gupta, D. (2023). Comparative Analysis of Feature Selection Methods for Detection of Android Malware. In T. Kavitha, G. Senbagavalli, D. Koundal, Y. Guo, & D. Jain (Eds.), Convergence of Deep Learning and Internet of Things: Computing and Technology (pp. 263-284). IGI Global. https://doi.org/10.4018/978-1-6684-6275-1.ch013

[14] Chaganti, R., Ravi, V., & Pham, T. D. (2023). A multi-view feature fusion approach for effective malware classification using Deep Learning. In Journal of Information Security and Applications (Vol. 72, p. 103402). Elsevier BV. https://doi.org/10.1016/j.jisa.2022.103402

[15] Herrera-Silva, J. A., & Hernández-Álvarez, M. (2023). Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms. Sensors, 23(3), 1053. MDPI AG. Retrieved from http://dx.doi.org/10.3390/s23031053

[16] Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2022). A study on malicious software behaviour analysis and detection techniques: Taxonomy, current trends and challenges. In Future Generation Computer Systems (Vol. 130, pp. 1–18). Elsevier BV. https://doi.org/10.1016/j.future.2021.11.030

[17] Urooj, U., Al-rimy, B. A. S., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2021). Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Applied Sciences, 12(1), 172. MDPI AG. Retrieved from http://dx.doi.org/10.3390/app12010172

[18] Benjamin Jackson, Mark Johnson, Andrea Ricci, Piotr Wiśniewski, Laura Martínez. Intelligent Automation through the Integration of Machine Learning and Decision Science. Kuwait Journal of Machine Learning, 2(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/222

[19] Yadav, C. S., Singh, J., Yadav, A., Pattanayak, H. S., Kumar, R., Khan, A. A., Haq, M. A., et al. (2022). Malware Analysis in IoT & Android Systems with Defensive Mechanism. Electronics, 11(15), 2354. MDPI AG. Retrieved from http://dx.doi.org/10.3390/electronics11152354

[20] Rath, M. & Mishra, S. (2022). Advanced-Level Security in Network and Real-Time Applications Using Machine Learning Approaches. In I. Management Association (Ed.), Research Anthology on Machine Learning Techniques, Methods, and Applications (pp. 664-680). IGI Global. https://doi.org/10.4018/978-1-6684-6291-1.ch035

[21] Fahd Alhaidari, Nouran Abu Shaib, Maram Alsafi, Haneen Alharbi, Majd Alawami, Reem Aljindan, Atta-ur Rahman, Rachid Zagrouba, "ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques", Computational Intelligence and Neuroscience, vol. 2022, Article ID 1615528, 15 pages, 2022. https://doi.org/10.1155/2022/1615528

[22] Usman, N., Usman, S., Khan, F., Jan, M. A., Sajid, A., Alazab, M., & Watters, P. (2021). Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics. In Future Generation Computer Systems (Vol. 118, pp. 124–141). Elsevier BV. https://doi.org/10.1016/j.future.2021.01.004

[23] A. Abusnaina et al., "DL-FHMC: Deep Learning-Based Fine-Grained Hierarchical Learning Approach for Robust Malware Classification," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3432-3447, 1 Sept.-Oct. 2022, doi: 10.1109/TDSC.2021.3097296.

[24] Srinivasan, S., Vinayakumar, R., Arunachalam, A., Alazab, M., & Soman, K. (2020). DURLD: Malicious URL Detection Using Deep Learning-Based Character Level Representations. In Malware Analysis Using Artificial Intelligence and Deep Learning (pp. 535–554). Springer International Publishing. https://doi.org/10.1007/978-3-030-62582-5_21

[25] Suryanarayana, G., Prakash K, L., Mahesh, P.C.S. et al. Novel dynamic k-modes clustering of categorical and non categorical dataset with optimized genetic algorithm based feature selection. Multimed Tools Appl 81, 24399–24418 (2022). https://doi.org/10.1007/s11042-022-12126-5

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7235
***Article Received**: 30 April 2023 **Revised:** 14 June 2023 **Accepted:** 29 June 2023*

[26] Balakrishna, G., V. Radha, and K. Rao. "ESMP: EXPLORATORY SCALE FOR MALWARE PERCEPTION THROUGH API CALL SEQUENCE LEARNING." journal of theoretical & applied information technology 95.10 (2017).

[27] Daoudi, N., Samhi, J., Kabore, A. K., Allix, K., Bissyandé, T. F., & Klein, J. (2021). DexRay: A Simple, yet Effective Deep Learning Approach to Android Malware Detection Based on Image Representation of Bytecode. In Deployable Machine Learning for Security Defense (pp. 81–106). Springer International Publishing. https://doi.org/10.1007/978-3-030-87839-9_4

[28] Mahindru, A., Sangal, A. FSDroid:- A feature selection technique to detect malware from Android using Machine Learning Techniques. Multimed Tools Appl 80, 13271–13323 (2021). https://doi.org/10.1007/s11042-020-10367-w

[29] Hira, S., Bai, A. A Novel Map Reduced Based Parallel Feature Selection and Extreme Learning for Micro Array Cancer Data Classification. Wireless Pers Commun 123, 1483–1505 (2022). https://doi.org/10.1007/s11277-021-09196-3

[30] Agrawal, P., & Trivedi, B. (2020). Machine Learning Classifiers for Android Malware Detection. In Data Management, Analytics and Innovation (pp. 311–322). Springer Singapore. https://doi.org/10.1007/978-981-15-5616-6_22