Optimization Techniques for Intrusion Detection System

By

Juhi Saha
(10900220030)

Pritom Saha
(10900220046)

Under the Guidance of

Dr. Partha Ghosh

- Introduction
- Objectives
- System Configuration
- Dataset
- IDS using Feature Selection based on Modified Firefly Algorithm
- IDS using Feature Selection based on Grasshopper Optimisation Algorithm
- Comparative Study
- Conclusions & Future Work
- References

Introduction

➢ Intrusion Detection System:

- Intrusion Detection System (IDS) is a software application that monitors the network traffic to detect any malicious activity and alerts the administrator when discovered.
- A machine learning based IDS involves extensive computational resources. So, Feature Selection (FS) technique is used to reduce the dataset.

➢ Optimization Techniques:

- Optimization is a procedure of getting the best possible result for a given condition or constraint.
- A FS (Feature Selection) technique finds the optimal reduced feature subset. So, FS falls under the category of the optimization problem.
- Nature-inspired metaheuristic algorithms mimic biological or physical phenomena.
- Some of the popular nature-inspired metaheuristics algorithms are Firefly Algorithm (FA) modeled on flashing behaviour of fireflies,Grasshopper Optimization Algorithm for the metaheuristic behaviour of grasshopper, Particle Swarm Optimization(PSO) inspired by the social behavior of birds or fish etc.
- These algorithms are highly efficient if they are perfectly poised between exploration and exploitation.

Objectives

- The main objective of this project is to develop effective Intrusion Detection System (IDS) models to protect the network and devices from attacks.
- All the features of a dataset are not of equal importance. To obtain the most useful features from the complete dataset, Feature Selection methods through Optimization Techniques have been proposed in this project.
- The datasets which are used for training and testing purposes are very large in size. For this reason, they require a good amount of computing resources and very long time. The reduced feature sets obtained by the Feature Selection techniques can reduce the computation power and time considerably.

System Configuration

- The proposed model is built on Python 3.9
- scikit-learn version 0.20.2 library, developed by David Cournapeau, is used for implementing the machine learning techniques.
- The experiments are executed in an Intel® Core™ i5 processor@2.20GHz system with 8GB RAM running Windows 11 OS.
- The proposed IDSs is deployed on a cloud simulating framework, CloudSim.

Dataset

- In this project, the proposed optimization technique is applied on NSL-KDD dataset developed by Mahbod Tavallaee et al. in 2009.
- NSL-KDD dataset contains four components – KDD Train+, KDD Test +, 20% KDD Training and KDD Test-21.
- KDD Train+ (1,25,973 instances) is used to train the model and KDD Test+ (22,544 instances) is used for testing the model.
- Each record of the datasets has 41 features along with a class label – either normal or attack.
- The attacks are categorized into four categories – DoS (Denial of Service), Probe, R2L (Remote to Local) and U2R (User to Root).
- The features of NSL-KDD dataset can be classified into four types – basic features, content features, time-based traffic features, host-based traffic features.

IDS using Feature Selection based on Modified Firefly Algorithm

- A method to develop a proficient IDS is described in this project.
- In modified FA , optimize solution is more feasible apart from not being trapped under the local maxima.
- Our study enhances the global search capability by integrating the Genetic Algorithm's mutation operator.
- The modification is applied to the NSL-KDD dataset for obtaining an optimal feature subset.
- The Genetic Algorithm(GA) mutation operator will help the FA by facilitating the random movement of the best firefly in the population towards finding a new position.
- The proposed method's effectiveness is evaluated using classifiers like NN, DT, RF, and KNN.

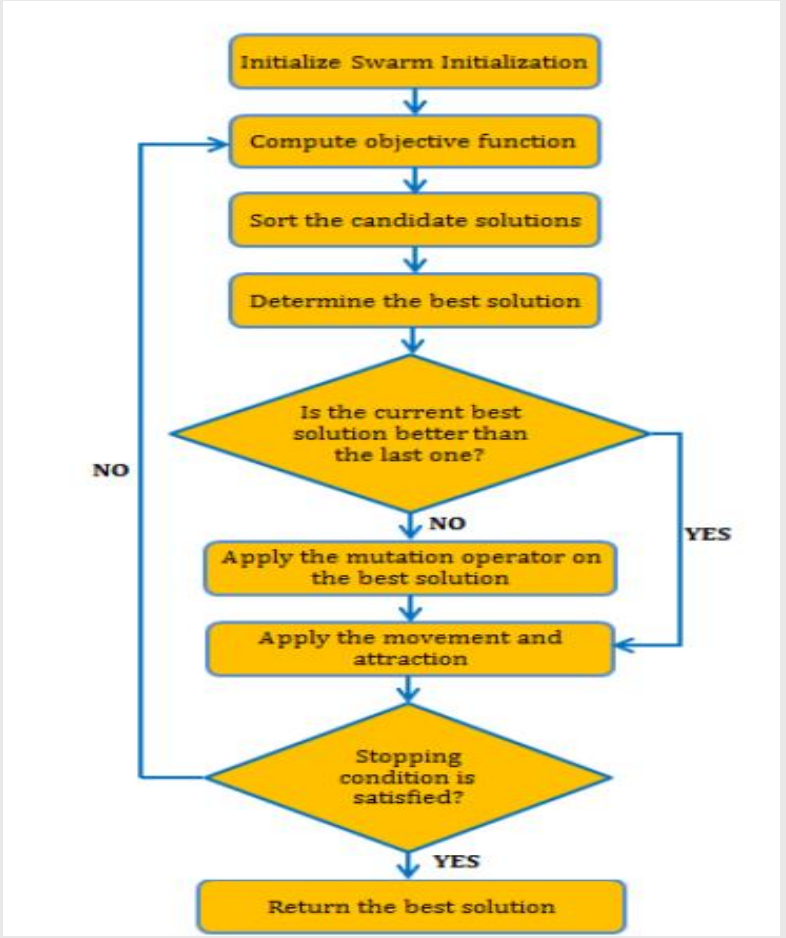

Figure 1. flowchart of the proposed model

Feature Selection Algorithm

- **Swarm Initialization:** This step involves random initialization of each firefly in the population. This initialization is done in a continuous domain using a uniform distribution as shown by Eq. (1).

$$x_i = (UB - LB) \times Rand + LB \tag{1}$$

- **Fitness Function Calculation:** The fitness function aids in guiding the search via assigning a quality value to any potential solution. The objective function relies on both the accuracy and number of features for the evaluation of the solutions using Eq. (2).

$$\min f(x_i) = \sum_{i=1}^{n} (x_i - Accuracy)^2 \tag{2}$$

The accuracy is obtained using DT classifier, which can be calculated based on the probability of the features.

- **Distance calculation:** The distance between f_i and f_j is expressed as f_{ij} and is described using Eq. (3).

$$f_{ij} = \left\| X_i - X_j \right\| = \sqrt{\Sigma_{d=1}^{D} \left(X_{id} - X_{jd} \right)^2} \tag{3}$$

where X_{id} is the position of firefly i. The Euclidean distance has been used to calculate the distance between two fireflies. In the suggested work, D represents the total number of features of a firefly.

- **Attractiveness**: For each firefly, the attractiveness β can be calculated by using Eq. (4).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{4}$$

where r symbolizes the distance between two fireflies and β_0 symbolize the attractiveness at r = 0 which is assumed as 1.

- **Updating Position of Fireflies:** The fireflies in the population move to other fireflies with greater light intensity based on Eq. (5).

$$X_{new} = X_{old} + \beta \times \left(X_j - X_i\right) + \alpha(Rand - 0.5) \tag{5}$$

Where X_{new} is the updated position, X_{old} is the previous position, the second part of the equation contains the attractiveness between the position of F_i and F_j, the third section symbolizes the random movement – α is the randomization parameter, while *Rand* is a uniformly distributed random number ranging from 0 to 1. Hence, the expression ($Rand$ − 0.5) ranges from -0.5 to 0.5 to accommodate both positive and negative changes.

- **Real Value to Binary Value Conversion:** Once the firefly gets updated on its position, the value of a dimension is a real value. But there is a need to transform the real value into binary i.e., 0 and 1 to determine which features are to be selected. So, a probabilistic rule based on a hyperbolic tangent sigmoid transfer function is applied to each dimension of the position vector. The formula is given in Eq. (6).

$$x_{id} = \begin{cases} 1, if\ \ rand < S(x_{id}) \\ O\ ,\ otherwise \end{cases} \tag{6}$$

Where, $$s(x_{id}) = tanh(|x_{id}|) = \frac{exp(2*|x_{id}|)-1}{exp(2*|x_{id}|)+1} \quad (7)$$

This way the process continues updating the fireflies and the k-best fireflies are taken out every time. Finally, after the maximum iteration is reached the firefly with the best fitness value is opted as the optimal feature subset. The experiment with the proposed model shows that it has performed better than the proposed model for the work of feature selection has done better than the full feature set of NSL-KDD in terms of both decreasing the processing time and the memory requirement.

Mutation: This operator is used for the best firefly as described earlier. The best firefly maintains its position in the original FA, and this slows down the search process and makes the algorithm more prone to local optima entrapment. However, in the proposed GA-FA variant, the GA mutation operator is employed to update the position of the best firefly via random exchange of the features and variables. In order to execute this operator, a few steps should be performed. At first, an even random number representing how many features are required to be swapped inside the best firefly is set. Then, half of the number is swapped with the other half. The mutation operator enhances the searching process of fireflies and decreases the chances of the algorithm getting stuck in the search space.

Assumptions: Idealizing firefly flashing characteristics, the Firefly Algorithm (FA) is formulated with three simplified rules:

1. Unisex Fireflies: All fireflies are considered unisex, eliminating sex-specific attraction. Any firefly is attracted to others regardless of sex.

2. Brightness-based Attraction: Attractiveness is directly proportional to brightness. When comparing two flashing fireflies, the less bright one moves towards the brighter one. Attractiveness and brightness decrease with distance. If no brighter firefly is nearby, movement is random.

3. Objective Function Influence on Brightness: Brightness is influenced by the landscape of the objective function. For maximization problems, brightness is proportional to the objective function value. Different forms of brightness can be defined by fitness functions in genetic algorithms.

Pseudo Code For Modified Firefly Optimization Techniques

- **Input:** KDD Train+ Dataset
- **Output:** Feature Score of each feature for each decision class
- **Steps:**

Initialize parameters
Objective function f(x), x = (x_1, ..., x_n)
Generate initial population of fireflies x_i (i = 1, 2, ..., n)
Calculate fitness value of each firefly
Set light intensity I_i at x_i
Set light absorption coefficient
while (t <MaxGeneration)
    for i = 1 : n
        for j = 1 : i
            if (I_j > I_i),
                Move firefly i towards j in d-dimension;
            end if
            Update attractiveness
            Evaluate new solutions and update light intensity
        end for j
    end for i
    Rank the fireflies and find the current best
end while
Postprocess results and visualization

Result

- To evaluate the proposed model, NSL-KDD dataset has been used.
- Only 26 out of a total of 41 features are obtained through the modified FA algorithm. The selected features are 1, 2, 3, 4, 5, 8, 11, 13, 17, 18, 19, 21, 22, 23, 25, 26, 27, 30, 31, 33, 34, 35, 36, 37, 40, 41
- Neural Network(NN), Random Forest(RF), Decision Tree(DT) and K-Nearest Neighbour(KNN) classifiers are applied to evaluate the optimal feature subset.
- In case of Neural Network(NN), the maximum accuracy is achieved for the reduced feature set.

Method	NN	RF	DT	KNN
FA(26)	77.5%	78.9124%	81.0060%	78.0696%
All (41)	77.0272%	77.4796%	78.7083%	77.6082%

Table 2. Classification Accuracy of Different Classifiers on Different Feature Sets

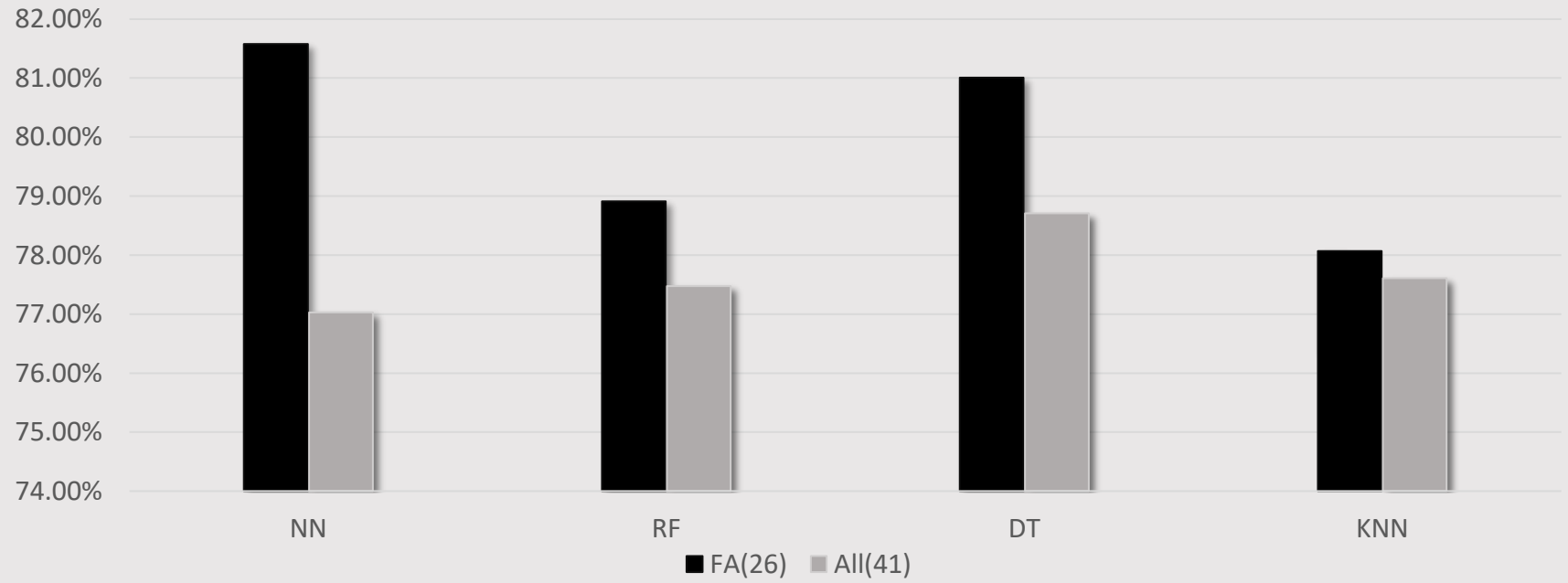

Figure 2. Comparison between Classification Accuracy of different Classifiers

IDS using Feature Selection based on Grasshopper Optimisation Algorithm

- Utilizes a nature-inspired phenomenon for developing an effective Intrusion Detection System (IDS).
- IDS needs to be trained using datasets before deployment.
- Not all dataset features are equally useful, necessitating data reduction techniques to retain important data.
- Grasshopper Optimization Algorithm (GOA) is applied to the NSL-KDD dataset, where each grasshopper represents a data point and each dimension represents a feature.
- Accuracy of a feature subset determined by the fitness value of individual grasshoppers.
- Obtained from the grasshopper with the highest fitness value.
- Various classifiers are employed to calculate fitness values.

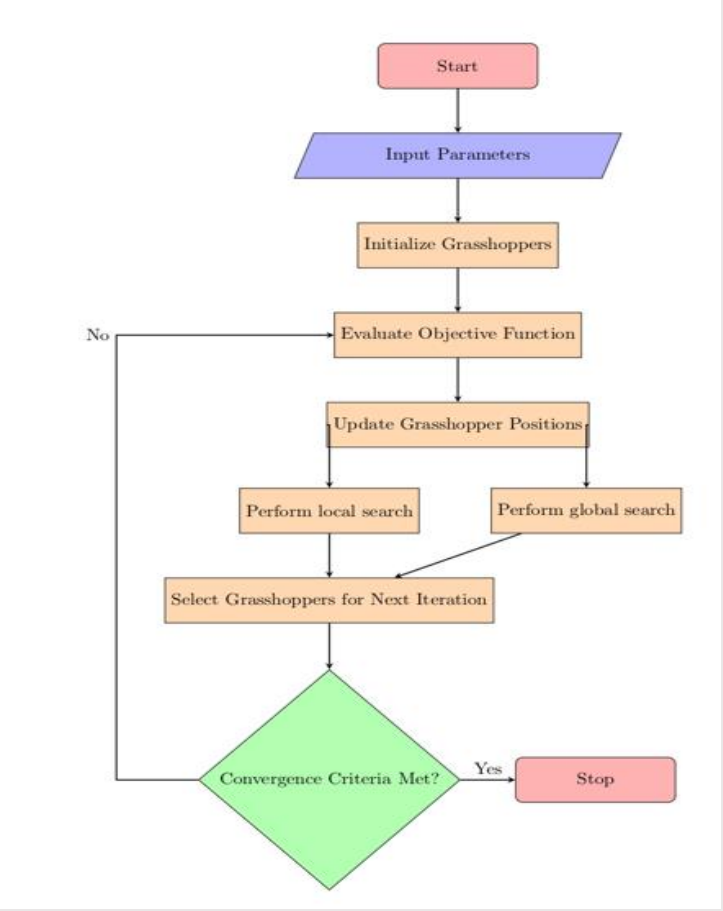

Figure 3. flowchart of the proposed model

Feature Selection Algorithm

- The Grasshopper Optimization Algorithm (GOA) is inspired by the swarming behavior of grasshoppers in nature. The algorithm mimics their behavior to find optimal solutions in a search space.
- **Initialization:** Initialize a population of grasshoppers randomly within the search space.

  Define parameters such as maximum iterations and convergence criteria.

$$x_i = (UB - LB) \times Rand + LB \qquad (1)$$

- **Fitness Function Calculation:** The fitness function aids in guiding the search via assigning a quality value to any potential solution. The objective function relies on both the accuracy and number of features for the evaluation of the solutions using Eq. (2).

$$\min f(x_i) = \sum_{i=1}^{n} (x_i - Accuracy)^2 \qquad (2)$$

The accuracy is obtained using DT classifier, which can be calculated based on the probability of the features. The predicted class is calculated based on the output of the maximum probabilities for all possible values. The next step after calculating the error rate, the light intensity for each firefly is calculated by using Eq. (3).

$$I(F_i) = \frac{1}{1+error^2} \qquad (3)$$

- **Update Grasshopper Positions**:

Update each grasshopper's position based on the local and global search strategies. The position update of each grasshopper is influenced by other grasshoppers

$$x_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})(x_j - x_i) + T_g$$

Where ;

- $s(d_{ij})$ is the social interaction function.
- $d_{ij}=|x_j - x_i|$ is the distance between grasshopper i and grasshopper j.
- T_g is the target position, guiding the global search.

- .**Social Interaction Function:**

$$s(d) = fe^{-d/l} - e^{-d}$$

Where;

- F is the intensity of attraction.
- L is the attractive length scale.

- **Convergence Check:**
  - Check if the convergence criteria are met. If yes, stop the algorithm; otherwise, continue to the next iteration.

- **Real Value to Binary Value Conversion:** Once the firefly gets updated on its position, the value of a dimension is a real value. But there is a need to transform the real value into binary i.e., 0 and 1 to determine which features are to be selected. So, a probabilistic rule based on a hyperbolic tangent sigmoid transfer function is applied to each dimension of the position vector. The formula is given in Eq. (7).

$$x_{id} = \begin{cases} 1, if\ \ rand < S(x_{id}) \\ 0\ ,\ otherwise \end{cases} \quad (7)$$

where, $$s(x_{id}) = tanh(|x_{id}|) = \frac{exp(2*|x_{id}|)-1}{exp(2*|x_{id}|)+1} \quad (8)$$

This way the process continues updating the fireflies and the k-best fireflies are taken out every time. Finally, after the maximum iteration is reached the firefly with the best fitness value is opted as the optimal feature subset. The experiment with the proposed model shows that it has performed better than the proposed model for the work of feature selection has done better than the full feature set of NSL-KDD in terms of both decreasing the processing time and the memory requirement.

Additional Notes:

- The social interaction function s(d) and the parameters f and l are crucial for balancing exploration and exploitation in the search space.

- The target position Tg can be updated dynamically based on the current best solutions to guide the search process effectively.

- Convergence criteria can include a maximum number of iterations, a target fitness value, or minimal changes in fitness over successive iterations.

Pseudo-Code of Grasshopper Optimization Algorithm

- **Input:** KDD Train+ Dataset
- **Output:** Optimal Feature Subset
- **Steps**:

Initialize parameters
N: Population size
MaxIter: Maximum iterations
Xmin, Xmax: Lower and upper bounds of the search space
ConvergenceCriteria: Criteria for convergence

Objective function f(X)

Initialize Grasshoppers
For each grasshopper i:
Initialize position Xi = random(Xmin, Xmax)

Evaluate Objective Function
For each grasshopper i:
Evaluate fitness f(Xi)

Repeat until Convergence Criteria Met:
t = 0
while (t < MaxIter and not ConvergenceCriteria):
Update Grasshopper Positions
For each grasshopper i:
Update position based on the social interaction with other grasshoppers and target position

Perform Local Search
Adjust position based on local information (e.g., best local solutions)

Perform Global Search
Adjust position based on global information (e.g., global best solution)

Select Grasshoppers for Next Iteration
Choose grasshoppers based on their updated positions and fitness values

Check Convergence Criteria
If convergence criteria are met, proceed to Termination

Increment t

Termination
End algorithm

Result

- To evaluate the proposed model, NSL-KDD dataset has been used.
- Evaluated using Neural Network (NN), Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN) classifiers.
- Only 26 out of a total of 41 features are obtained through the modified FA algorithm. The selected features are 1, 2, 3, 4, 5, 8, 11, 13, 17, 18, 19, 21, 22, 23, 25, 26, 27, 30, 31, 33, 34, 35, 36, 37, 40, 41
- In case of Random Forest (RF), the maximum accuracy is achieved for the reduced feature set.

Method	NN	RF	DT	KNN
Full Feature Set Accuracy	77.5%	80.2%	75.1%	78.9%
Reduced feature set Accuracy	84.3%	85.6%	81.2%	83.4%

Table 2. Classification Accuracy of Different Classifiers on Different Feature Sets

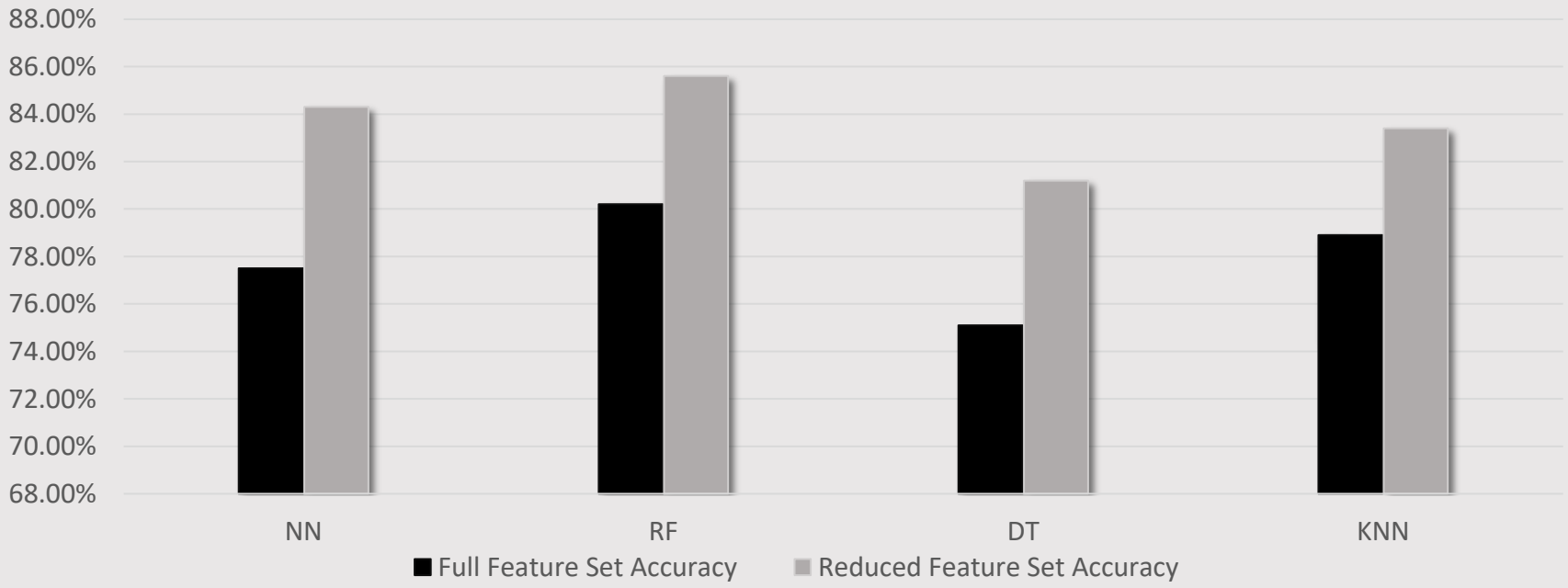

Figure 2. Comparison between Classification Accuracy of different Classifiers

Comparative Study

- The two proposed models in this project have been compared with the IDS models developed by other authors. Table 1 and table 2 depict the comparisons of train-test and 10-fold cross validation performances respectively.

Table 1. Comparison between train-test results of different IDS models

Algorithm	No. of Features	Classification Accuracy (%)	
		NN	DT
Information Gain [13]	13	76.663	79.724
Gain Ratio [13]	13	77.089	80.092
ReliefF [13]	13	80.487	81.175
EMFFS [13]	14	77.009	80.820
DMFSM [9]	33	73.550	81.938
TVCPSO-SVM [17]	17	80.935	81.095
TVCPSO-MCLP [17]	17	75.608	77.564
FSNIDS [15]	25	79.183	79.924
FA based IDS (proposed model 1)	**26**	**81.57**	**81.00**
GOA based IDS (proposed model 2)	**26**	**84.3**	**81.2**

Table 2. Comparison between cross-validation results of different IDS models

Authors	Detection Rate (%)	False Alarm Rate (%)
de la Hoz et al. [10]	93.40	14
Kang et al. [15]	99.10	1.2
Bamakan et al. [17]	97.03	0.87
Singh et al. [26]	97.67	1.74
Tavallaee et al. [20]	80.67	NA
Raman et al. [27]	97.14	0.83
Abd-Eldayem [28]	99.03	1.0
Gogoi et al. [29]	98.88	1.12
FA based IDS (proposed model 1)	**99.74**	**0.0115**
GOA based IDS (proposed model 2)	**99.57**	**0.174**

Conclusion & Future Work

- In this project, two IDS models using feature selection through an optimization techniques named modified Firefly Algorithm (modified FA) and Grasshopper Optimization Algorithm (GOA) are developed.
- The efficiency of both the feature selection techniques is evident from the classification phase described in the experimental results.
- For future work, the proposed feature selection techniques can be applied on other datasets to test its efficiency in different cases.
- Different combinations of classifiers can be tested to achieve better detection of intrusions.

Reference

[1] S. O. Al-Mamory, H. Zhang, "IDS alerts correlation using grammar-based approach", *Journal in Computer Virology*, 5, 2009. https://doi.org/10.1007/s11416-008-0103-3

[2] Y. Mehmood, M. A. Shibli, U. Habiba, R. Masood, "Intrusion Detection System in Cloud Computing: Challenges and Opportunities", *2013 2nd National Conference on Information Assurance (NCIA)*, 2013, 59-66. https://doi.org/10.1109/NCIA.2013.6725325

[3] P. Ghosh, M. Bardhan, N. R. Chowdhury, S. Phadikar, "IDS Using Reinforcement Learning Automata for Preserving Security in Cloud Environment", *International Journal of Information System Modeling and Design (IJISMD)*, 8(4), 21–37, 2017. https://doi.org/10.4018/IJISMD.2017100102

[4] P. Deshpande, S. C. Sharma, S. K. Peddoju, S. Junaid, "HIDS: A host-based intrusion detection system for cloud computing environment", *International Journal of System Assurance Engineering and Management*, 9, 567–576, 2018. https://doi.org/10.1007/s13198-014-0277-7

[5] N. Harale, B. B. Meshram, "Network Based Intrusion Detection and Prevention Systems: Attack Classification, Methodologies and Tools", *International Journal of Engineering and Science*, 6(5), 1–12, 2016. http://www.researchinventy.com/papers/v6i5/A60501012

[6] O. Depren, M. Topallar, E. Anarim, M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", *Expert Systems with Applications*, 29(4), 713–722, 2005. https://doi.org/10.1016/j.eswa.2005.05.002

[7] K. J. Mathai, K. Agnihotri, "Optimization techniques for feature selection in classification", *International Journal of Engineering Development and Research*, 5(3), 1167–1170, 2017. https://www.ijedr.org/papers/IJEDR1703165

[8] R. H. Abiyev, M. Tunay, "Optimization of High-Dimensional Functions through Hypercube Evaluation", *Computational Intelligence and Neuroscience*, 2015, 1–11, 2015. https://doi.org/10.1155/2015/967320

[9] K. Bajaj, A. Arora, "Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods", *International Journal of Computer Applications*, 76(1), 5–11, 2013. https://doi.org/10.5120/13209-0587

[10] E. de la Hoz, A. Ortiz, J. Ortega, E. de la Hoz, "Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-linear Projection Techniques", *International Conference on Hybrid Artificial Intelligence Systems*, *Springer*, 103–111, 2013. https://doi.org/10.1007/978-3-642-40846-5_11

[11] Amrita, P. Ahmed, "A Hybrid-Based Feature Selection Approach for IDS", *Networks and Communications (NetCom2013), Lecture Notes in Electrical Engineering*, *Springer*, 284, 195–211, 2014. https://doi.org/10.1007/978-3-319-03692-2_16

Reference

[12] P. Ghosh, C. Debnath, D. Metia, R. Dutta, "An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment", *IOSR Journal of Computer Engineering (IOSR-JCE)*, 16(4), 16–26, 2014. http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue4/Version-7/D016471626

[13] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu, M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", Springer, *EURASIP Journal on Wireless Communications and Networking* 2016, *Springer*, 2016. https://doi.org/10.1186/s13638-016-0623-3

[15] S.-H. Kang, K. J. Kim, "A feature selection approach to find optimal feature subsets for the network intrusion detection system", *Cluster Computing*, 19, 325–333, 2016. https://doi.org/10.1007/s10586-015-0527-8

[16] M. S. Akhtar, D. Gupta, A. Ekbal, P. Bhattacharyya, "Feature Selection and Ensemble Construction: A Two-step Method for Aspect Based Sentiment Analysis", *Knowledge-Based Systems*, 125, 116–135, 2017. https://doi.org/10.1016/j.knosys.2017.03.020

[18] S. Mirjalili, "Knowledge-Based Systems Moth-Flame Optimization Algorithm: A Novel Nature-inspired Heuristic Paradigm", *Knowledge-Based Systems*, 89, 228–249, 2015. https://doi.org/10.1016/j.knosys.2015.07.006

[19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, 12, 2825–2830, 2011. https://jmlr.org/papers/volume12/pedregosa11a

[20] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[22] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, N. Yazdani, "Mutual information-based feature selection for intrusion detection systems", *Journal of Network and Computer Applications*, 34(4), 1184–1199, 2011. https://doi.org/10.1016/j.jnca.2011.01.002

[23] H. Uğuz, "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm", *Knowledge-Based Systems*, 24(7), 1024–1032, 2011. https://doi.org/10.1016/j.knosys.2011.04.014

[26] R. Singh, H. Kumar, R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine", *Expert Systems with Applications*, *Elsevier*, 42(22), 8609–8624, 2015. https://doi.org/10.1016/j.eswa.2015.07.015

[27] M. R. G. Raman, N. Somu, K. Kannan, R. Liscano, V. S. S. Sriram, "An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine", *Knowledge-Based Systems*, *Elsevier*, 134, 1–12, 2017. https://doi.org/10.1016/j.knosys.2017.07.005

Reference

[28] M. M. Abd-Eldayem, "A proposed HTTP service based IDS", *Egyptian Informatics Journal*, *Elsevier*, 15(1), 13–24, 2014. https://doi.org/10.1016/j.eij.2014.01.001

[29] R. Singh, H. Kumar, R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine", Expert Systems with Applications, 42(22), 8609–8624, 2015. https://doi.org/10.1016/j.eswa.2015.07.015

[30] P. Gogoi, M. H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, "Packet and Flow Based Network Intrusion Dataset", *International Conference on Contemporary Computing*, *Communications in Computer and Information Science*, *Springer*, 2012, 306, 322–334. https://doi.org/10.1007/978-3-642-32129-0_34

[31] S. Saremi, S. Mirjalili, A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application", Advances in Engineering Software, 105, 30-47, 2017. https://doi.org/10.1016/j.advengsoft.2017.01.004

[32] S. Bansal, K. Deep, "Enhanced Grasshopper Optimization Algorithm for Continuous Functions and Engineering Applications", Expert Systems with Applications, 149, 113322, 2020. https://doi.org/10.1016/j.eswa.2020.113322

[33] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization", Stochastic Algorithms: Foundations and Applications, SAGA 2009. Lecture Notes in Computer Science, vol 5792. Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-04944-6_14

[34] A. T. A. Rahman, A. R. Ramli, M. F. Miskon, "Firefly algorithm for solving manufacturing scheduling problems: A comprehensive survey", Applied Soft Computing, 100, 106922, 2021. https://doi.org/10.1016/j.asoc.2020.106922

THANK YOU