A Collaborative Intrusion Detection System Using Machine Learning

Krisha Khandhar[1], Shreya Bachhav[2], Shrutika Badgujar[3], Neha Bagul[4]
[1]*Department of Information Technology, Nashik, India*
Savitribai Phule Pune University

***Abstract*—The design and implementation of a Collaborative Intrusion Detection System (CIDS) for precise and effective intrusion detection in a distributed system are presented in this study. The network, kernel, and application levels are where CIDS uses a variety of specialised detectors. In essence, CIDS combines the alarms from these detectors to produce a single intruder alarm. In comparison to separate detectors, this improves detection accuracy without noticeably degrading performance. The optimization algorithm is utilised to help those detectors find the attack faster, and graph-based detection is demonstrated to find the attack. The same is done using machine learning techniques, from feature selection and normalisation to categorization and attack detection.**

***Index Terms*— CIDS, Intrusion, Machine Learning, Optimization**

I. INTRODUCTION

Cybercriminals refine their attacks using a variety of techniques, tools, and methods as technology advances. The organization's network may be affected by these cyberattacks. Avoiding these cyberattacks is therefore urgently needed. Since all the data is kept on the network, its security cannot be jeopardized. IDS is a key network security tool that aids in identifying unusual and unauthorized access to secure networks. To safeguard computer infrastructures, intrusion detection systems (IDSs) are used. The two types of traditional IDSs are anomaly-based and misuse-based. False alarms (also known as false positives and false negatives) and missed alarms are the metrics used to assess an IDS. One of the most important network security tools, the IDS, aids in spotting unusual and unauthorized access to secure networks.

In IDS, classification, feature selection, and normalisation are all accomplished using various machine learning models and algorithms. The performance is optimised with the aid of these machine learning models. It is a mechanism used by IDS to identify attacks. Today, more machine learning approaches are being used to identify various cyberattacks. The best outcomes are achieved when machine learning is used, which improves accuracy and efficiency. The dataset used in the suggested system is KDDCUP99, and this model is adaptable and flexible enough to capture interdependencies. These characteristics can be categorical or continuous. Both supervised and unsupervised machine learning models can be utilised, depending on the availability of labels.

IDS uses a variety of algorithms, including both bio-inspired and non-bio-inspired algorithms. Lately, swarm-based, evolutionary, genetic, and other bio-inspired algorithms have become more popular.

IDS is developed using a hybrid approach that combines Particle Swarm Optimisation (PSO) and Support Vector Machine (SVM). Swarm intelligence algorithms outperform the competition. It is also possible to utilise a weighted local search method with simplified swarm optimization. A PSO-based K-means algorithm also produces superior outcomes. The most intelligent algorithm is PSO, which enhances global search. Kernel principal component analysis (KPCA), support vector machines, and genetic algorithms are all incorporated. The dimension and training time are decreased via SVM. The performance is enhanced via self-organized ant-based clustering.

II. RESEARCH MODEL AND HYPOTHESES DEVELOPMENT

In collaborative intrusion system project, the research model can be developed by considering the following components:

1. Research Objectives: The research objectives of the collaborative intrusion system project can be defined as follows:

a) To investigate the effectiveness of collaborative intrusion detection systems (CIDS) in detecting and preventing network security threats.

b) To identify the factors that contribute to the success or failure of CIDS.

c) To propose strategies for improving the performance of CIDS.

2. Research Variables: The research variables that can be considered for the collaborative intrusion system project include:

a) Independent Variables: The independent variables can be the various components of the CIDS, such as sensors, analyzers, classifiers, and decision-makers.

b) Dependent Variables: The dependent variables can be the effectiveness of the CIDS in detecting and preventing network security threats.

3. Hypotheses: Based on the research objectives and variables, the following hypotheses can be developed:

a) Hypothesis 1: There is a positive relationship between the components of the CIDS and the effectiveness of the system in detecting and preventing network security threats.

b) Hypothesis 2: The effectiveness of the CIDS is influenced by the level of collaboration among its components.

c) Hypothesis 3: The success of the CIDS depends on the accuracy of the classifiers and decision-makers.

d) Hypothesis 4: The performance of the CIDS can be improved by incorporating machine learning and artificial intelligence techniques.

III. FINDING AND DISCUSSIONS

This system will detect the attack type and then send the alerts to the networks which are in collaboration with us . As the attack is detected with the help of machine learning the detection is faster and accurate results are displayed to us.The detection is done using a dataset which is then normalized classified and finally the attack is detected with more accuracy and faster speed This system proves to be useful for organizations which require utmost security .

The project aim is to provide a secure and effective way to detect the attack and be secured even before it hits your network . This system will detect the attack type and then send the alerts to the networks which are in collaboration with us . As the attack is detected with the help of deep learning the detection is faster and accurate results are displayed to us . The detection is done using a dataset which is then normalized classified and finally the attack is detected with more accuracy and faster speed. The dataset use in this is KDDCUP99 and the feature selection algorithm we are using is the ant colony and cuttle fish algorithm, this helps to detect the attack more faster and accurate results are displayed, the SVM normalization is combined with this algorithm and the final result is then send to the IDS monitors and then to the other within the network. This system proves to be useful for organizations which require utmost security.

IV. SYSTEM OVERVIEW

Below is the architectural view of proposed system. It has two phases namely:
Collaborative Intrusion Detection System (CIDS)
Machine Learning (ML)

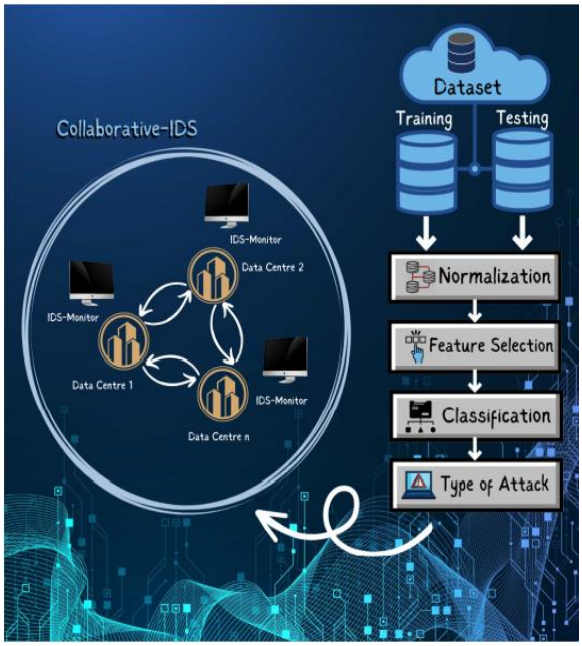

Fig 1 : System Architecture

Machine learning (ML) typically involves a series of stages, each of which is important in the development of a successful ML model. Here are the stages in an extended way:

1)Dataset

In the proposed system KDDCup99 dataset is used. A standard set of auditable data, including a wide range of simulated intrusions into a network environment, is contained in this database.

The KDDCup99 dataset is a widely used benchmark dataset for evaluating intrusion detection systems (IDS) and related cybersecurity research. The dataset was created by the US Department of Defense in 1999 to support research on intrusion detection systems.

The dataset contains a set of network traffic data that simulates a real-world network environment with various types of attack and normal traffic. It includes both TCP and UDP traffic, as well as a variety of attack types, such as DoS, probing, and user-to-root attacks.

The original KDD Cup 1999 dataset contains 41 attributes 34 continuous and 7 categorical but they have been condensed to just 4 attributes that are service, duration, src_bytes, and dst_bytes, as these are basic attributes with only "service" being categorical. The data is separated into subsets for http, smtp, ftp, ftp_data, and others using the 'service' atttribute.

2)Training and Testing

It is crucial to remember that the test data comes from a different probability distribution than the training data and contains particular attack types that weren't present in the training data. The task becomes more doable as a result. Some intrusion specialists contend that the majority of novel attacks are just modified versions of well-known ones, and that the "signature" of known assaults can be used to identify novel variants. There are a total of 24 training attack types in the datasets, while an additional 14 types are only present in the test data.

The train/test approach is a way to gauge how accurate your model is. Because you divide the data set into two sets—a training set and a testing set—this method is known as train/test.20% for testing, 80% for training. Train the model means create the model. Test the model means test the accuracy of the model.

```
from sklearn.model_selection import train_test_split
# Split dataset between training and testing (80/20 spl
it)
X_train, X_test, y_train, y_test = train_test_split(X, y
, test_size = 0.20, random_state = 42)
```

3)Normalization

A data preprocessing method called normalisation is used to convert the values of features in a dataset to a standard scale. This is done to make data modelling and analysis easier, as well as to lessen the effect of size differences on the precision of machine learning models.

A scaling technique called normalisation shifts and rescales values so that they fall between the ranges of 0 and 1. Additionally called Min-Max scaling.

In this stage of project the Min-Max scaling is used to transform the features by scaling each feature in the range of 0 and 1.

```
from sklearn.preprocessing import MinMaxScaler
min_max_sc = MinMaxScaler() # Transform features
 by scaling each feature (ranfge = (0,1))
X = min_max_sc.fit_transform(X)
```

4)Feature Selection

Ant Colony Optimization (ACO) and Cuttlefish Algorithm (CA) are both metaheuristic algorithms that can be used for feature selection in machine learning tasks. The KDDCup99 dataset is a widely-used dataset for intrusion detection and contains a large number of features, making it a good candidate for feature selection.

Here's a general overview of how ACO and CA can be used for feature selection on the KDDCup99 dataset:

1. Data preprocessing: The KDDCup99 dataset contains a large number of features (41 continuous and 14 categorical) and instances (over 4 million). Therefore, it is important to preprocess the data to remove any redundant or irrelevant features, as well as normalize the continuous features.

2. Feature ranking: Both ACO and CA use a ranking scheme to evaluate the quality of each feature. In ACO, this is done by using the pheromone trail of the ants to measure the importance of each feature. In CA,

the fitness of each feature is evaluated based on its distance to the centroid of the dataset.
3. Feature selection: After ranking the features, a subset of the top-ranked features is selected. This can be done by setting a threshold or using a greedy algorithm to select the top k features.

4. Model training and evaluation: Finally, the selected features are used to train a machine learning model (such as a decision tree or neural network) and its performance is evaluated using cross-validation or a holdout set.

5. Classification
A classification algorithm based on training data is a supervised learning technique used to classify new distinct observations.

Here in this project three classification algorithms are executed that are as follows:
1. Naive Bayes: This is a simple probabilistic algorithm that assumes that the features are independent. It is fast and easy to implement, but may not perform well on complex data.

2. Decision Trees: This algorithm creates a tree structure to classify the data. It is easy to understand and interpret, but can be prone to overfitting.

3. Random Forests: This is an ensemble algorithm that uses multiple decision trees to make predictions. It is more robust than decision trees and less prone to overfitting.

Among the above algorithms the Random Forests gives the better accuracy and performance while classifying the attack.
After classification the type of attack will be detected which is then sent as an alert to all the systems which are in collaborative manner with each other. After the attack is detected, the attack information will go to targeted IDS monitor. Through this monitor the alert will be sent to all IDS monitors and this is how the CIDS will work.

V. DIAGRAMATIC REPRESENTATION

A. Data Flow diagram

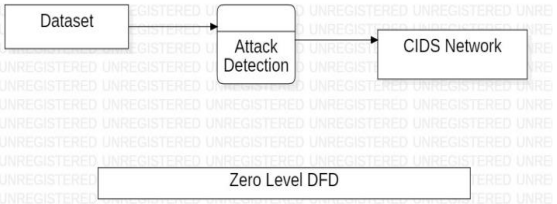

Fig 2: Zero Level DFD

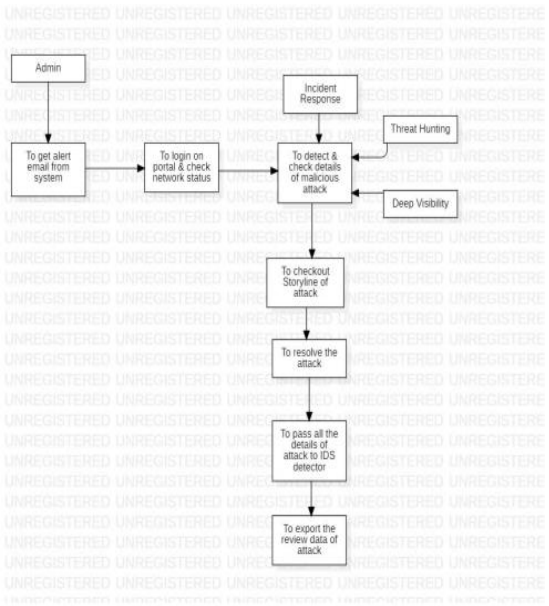

Fig:3 One Level DFD

The above data flow diagram gives the diagramatic view of exactly how the details of the attack detection will be showcased on the portal. It also provides information about the outputs and inputs of each entity and the process.

B. Class Diagram

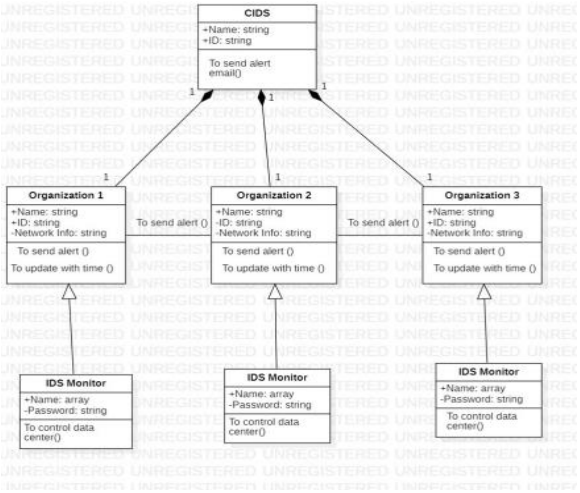

Fig:4 Class Diagram

Class diagram will represent each aspect or component of the system architecture in the form of class and will specify its working as the attribute of each class. Also it represents the dependency between different classes.

VI. CONCLUSION

In this paper, we have described the features and advantages of Collaborative Intrusion Detection System by studying various aspects related to it. Also the paper presents detailed information about the steps or the phases of Machine Learning which detect the attack. Each phase of Machine Learning plays an important role in the accurate attack detection. For the same, Optimization algorithm plays a crucial role. Hence, The optimization algorithms used are also briefly discussed and elaborated. The system architecture presented gives the visual representation which helps to understand the working of the CIDS as a whole. Also the Data flow diagrams and class diagram are also mentioned in order to get clear view of the entire process of the system.

REFERENCE

[1] Mehdi Hosseinzadeh Aghdam, Peyman Kabiri, et al. Feature selection for intrusion detection system using ant colony optimization. Int. J. Netw. Secur., 18(3):420–432, 2016.

[2] VR Balasaraswathi and M Sugumaran. A hybrid algorithm using ant colony optimisation and cuttle fish algorithm for feature selection of intrusion detection.

[3] L Dhanabal and SP Shantharajah. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering, 4(6):446–452, 2015.

4] Sheren Sadiq Hasan and Adel Sabry Eesa. Optimization algorithms for intrusion detection system: A review. 2020.

[5] Shailendra Sahu and Babu M Mehtre. Network intrusion detection system using j48 decision tree. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 2023– 2026. IEEE, 2015.

[6] Chibuzor John Ugochukwu, EO Bennett, and P Harcourt. An intrusion detection system using machine learning algorithm. LAP LAMBERT Academic Publishing, 2019.

[7] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max M¨uhlh¨auser, and Mathias Fischer. Taxonomy and survey of collaborative intrusion detection. ACM Computing Surveys (CSUR), 47(4):1–33, 2015.

[8] Sharmila Kishor Wagh, Vinod K Pachghare, and Satish R Kolhe. Survey on intrusion detection system using machine learning techniques. International Journal of Computer Applications, 78(16), 2013.

[9] Yu-Sung Wu, Bingrui Foo, Yongguo Mei, and Saurabh Bagchi. Collaborative intrusion detection system (cids): a framework for accurate and efficient ids. In 19th Annual Computer Security Applications Conference, 2003. Proceedings., pages 234–244. IEEE, 2003.

[10]M. A. Al-Qershi and A. T. Al-Khazraji, A Survey on Intrusion Detection Systems, published in the International Journal of Advanced Computer Science and Applications (IJACSA) , 2015.

[11] S. B. Nikam and S. N. Talbar, Intrusion Detection System: A Comprehensive Review, published in the International Journal of Computer Applications (IJCA),2013.

[12] A. D. Khan and S. Khan, Intrusion Detection System using Machine Learning Techniques: A Review, published in the International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) in 2016.

[13] M. S. Abdallah, R. E. Sabry, and M. I. Elhenawy, A collaborative intrusion detection system based on hybrid approaches, published in the Journal of Computer Science and Technology (JCST) in 2018.

[14] M. Wang, Y. Li, and S. Zeng, A collaborative intrusion detection system based on adaptive boosting and K-means clustering, published in the Journal of Network and Computer Applications (JNCA) in 2020.

[15] C. Zhang, Y. Li, and Y. Wang, A collaborative intrusion detection system based on genetic algorithms and support vector machines, published in the International Journal of Security and Communication Networks (IJSCN) in 2017.