RESEARCH ARTICLE OPEN ACCESS

Smart Intrusion Detection System Framework with Supervised Machine Learning Methods

Mrs. D. JYOTHI [1], S. MANASA [2]

[1] Assistant Professor,Computer Science & Engineering, Sree Vahini Institute Of Science & Technology, And Tiruvuru, N.T.R District - 521235, A.P-India.

[2] Pg Scholar Computer Science & Engineering, Sree Vahini Institute of Science & Technology, And Tiruvuru, N.T.R District - 521235, A.P-India.

ABSTRACT

In the rapidly evolving landscape of cyber security, the development of effective intrusion detection systems (IDS) is crucial to safeguarding sensitive information and critical infrastructures. This paper proposes a Smart Intrusion Detection System Framework that leverages the power of supervised machine learning methods to enhance the accuracy and efficiency of intrusion detection. The proposed framework integrates a diverse set of supervised machine learning algorithms, including but not limited to Decision Tree, Logistic regression, Random Forest and KNN, to analyse network traffic and identify patterns associated with malicious activities. This multi-algorithmic approach aims to mitigate the limitations of individual models and enhance the overall robustness of the intrusion detection system. The integration of supervised machine learning methods within the proposed framework offers a sophisticated and adaptive approach to intrusion detection, addressing the challenges posed by the ever-changing landscape of cyber threats. The framework's ability to learn from and adapt to new data makes it a valuable asset in enhancing the overall security posture of modern digital systems.

Keywords — Smart Intrusion Detection System, Sophisticated, Frame work, Cyber threats.

I. INTRODUCTION

Traditional Intrusion Detection Systems often struggle to keep up with the dynamic nature of cyber threats. The Smart IDS framework leverages the power of supervised machine learning to overcome these limitations, enabling the system to autonomously learn, adapt, and effectively distinguish between normal and malicious activities. In the ever-evolving landscape of cyber security, the need for robust Intrusion Detection Systems (IDS) is paramount. Traditional rule-based systems are proving insufficient to combat the sophistication of modern cyber threats. As a solution, Smart Intrusion Detection System frameworks leverage the power of Supervised Machine Learning (SML) methods to enhance the accuracy and efficiency of threat detection. This framework's main goal is to develop a dynamic and adaptive intrusion detection system that can quickly recognize and address different kinds of cyber threats. The system uses supervised machine learning techniques to learn from past data in order to precisely classify and predict possible intrusions.

II. KEY COMPONENTS OF THE FRAMEWORK

The critical parts of a Smart Intrusion Detection System (SIDS) coordinate different components to make a complete and viable safeguard against digital dangers. Here are the fundamental parts:

A. Information Assortment Module

Totals assorted datasets, including network traffic logs, framework logs, and application logs. Catches both ordinary and peculiar examples to give a thorough learning set to the framework.

B. Pre-processing and Feature Extraction

Pre-processes and cleans the raw data to get rid of noise and irrelevant data. Removes important highlights, for example, bundle size, recurrence, and timestamps for compelling model preparation.

C. Supervised Machine Learning Models

Uses deeply grounded calculations like Decision Tree, Logistic regression, Random Forest and KNN. Trains the models on marked datasets, recognizing typical and vindictive exercises.

D. Continuous Observing Module

Uses a system of continuous monitoring to look at the data streams that come in. Evaluates the behaviour in light of the patterns it has learned and sets off alerts for possible intrusions.

III. LITERATURE SURVEY

In recent years, machine learning-based intrusion detection systems (IDSs) have proven to be effective; especially, deep neural networks improve the detection rates of intrusion detection models. However, as models become more and more complex, people can hardly get the explanations behind their decisions.

TABLE I
LITERATURE SURVEY

Year	Paper Title	Machine learning Models	Limitations
2020[1]	An Explainable	Decision	This study offers

	Machine Learning Framework for Intrusion Detection Systems	tree, Bayes Classifier	valuable insight into the interpretability of the IDSs.
2021[2]	Explanation Framework for Intrusion Detection	local model-agnostic explanations (LIME)	theoretical framework for modular decision boundary
2022[3]	Explainable Artificial Intelligence for Intrusion Detection System	voting classifier , XAI algorithm LIME	No real-time data analysis and prediction performance evaluation
2023[4]	An explainable deep learning-enabled intrusion detection framework in IoT networks	Long Short-Term Memory (LSTM)	We demonstrate the proposed framework's ability to effectively enhance the interpretability of cyber defence systems in IoT networks.

IV. PRAPOSED ARCHITECTURE

The proposed system utilizes Shapley Added substance Clarifications (SHAP)[5], and consolidates nearby and worldwide clarifications to work on the understanding of IDSs. The neighbourhood clarifications give the reasons why the model goes with specific choices on the particular info. The relationships between feature values and various types of attacks are presented in the global explanations, which provide the essential features extracted from IDSs. The system is on working on the interpretability of the IDS. Accordingly, notwithstanding the IDS's expectations, neighbourhood clarifications and worldwide clarifications are created to further develop the security specialists' confidence in the IDS. There are two strategies for worldwide clarifications in this introduced structure.

The main technique can break down the significant elements of the IDS. The subsequent technique presents connections between the worth of a component and the effect on the forecast. The neighbourhood clarification makes sense of the result of IDS. This technique additionally gives the significance of information highlights for the IDS forecast. The network protection specialists can approve the IDSs choices by utilizing neighbourhood and worldwide translations. By consolidating the over two strategies, it can at last assistance security specialists to have a superior comprehension of IDSs. Besides, one-versus all classifier and multiclass classifier are utilized in this structure. Subsequently, by looking at the distinctions between these two classifiers, then make the expectations from these IDS more obvious.

The proposed smart intrusion detection system (IDS) is thought to be a good way to keep the network safe and protect it from outside threats.

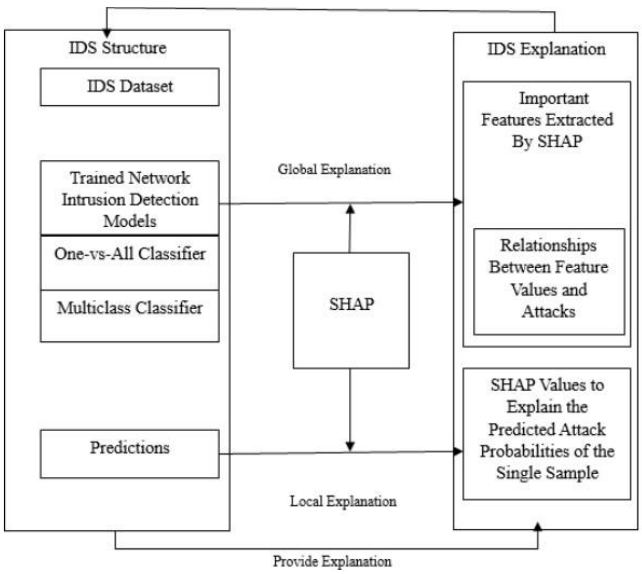

Fig. 1 Proposed model for Smart Intrusion Detection System

Notwithstanding, the current IDS frequently has a lower recognition rate under new assaults and has a high above while working with review information, and subsequently. AI strategies have been broadly applied in interruption location. In order to solve the classification problem of pattern recognition and intrusion identification, our proposed approach incorporates Decision Tree, Logistic regression, Random Forest, and KNN as learning methods.

V. SUPERVISED MACHINE LEARNING MODELS FRAME WORK FOR SIDS

Building a Smart Intrusion Detection System (IDS)[6] involves the use of supervised machine learning models to classify network activity as normal or malicious. Here are some commonly used algorithms and approaches for developing a supervised machine learning-based IDS:

A. Decision Trees

Decision trees [7] are natural and can be utilized to display the dynamic interaction for grouping network traffic. Highlights, for example, source and objective IP addresses ports, convention types, and bundle sizes can be utilized for choice tree development.

The decisions or the test are performed on the basis of features of the given dataset. o It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for

Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub trees.

The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the S into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

1) Attribute Selection Measures: While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are: Information Gain & Gini Index.

Information Gain and Gini Index are two metrics used in the context of decision trees for feature selection and node splitting. Both are employed in the ID3 (Iterative Dichotomiser 3), C4.5, and CART (Classification and Regression Trees) algorithms, which are popular for building decision trees.

Information Gain is a measure of the reduction in uncertainty or entropy achieved by splitting a dataset based on a particular attribute. It is used to decide the order in which attributes are tested at each node of a decision tree. For a dataset D, and an attribute A, the Information Gain (IG) is calculated as follows:

$$IG\,(D, A)=H(D)-\sum_{v\in Values(A)}|Dv/D|\cdot H(D_v) \text{ ------(1)}$$

Where $H(D)$ is the entropy of dataset D, $Values(A)$ are the unique values of attribute A, $|Dv|$ is the size of the subset of D for which attribute A has value v, and $H(Dv)$ is the entropy of that subset. Higher Information Gain indicates a better attribute for splitting the data because it maximally reduces uncertainty.

B. Logistic regression

A smart intrusion detection system (IDS) can be built using logistic regression [8]. With regards to arrange security, an IDS means to distinguish and answer malevolent exercises or unapproved admittance to a PC framework or organization. Calculated relapse, when applied to this issue, can assist with arranging network traffic or occasions into typical (non-meddlesome) or nosy classes.

Here is a general way to deal with involving strategic relapse for a shrewd interruption location framework:

In logistic regression, the model predicts the probability of a binary outcome. In the context of a smart intrusion detection system, let's assume we are predicting the probability of an intrusion (event denoted as 1) based on various features. The logistic regression model can be represented as follows:

Assuming you have n features $X_1, X_2,\ldots, X_n$, and the logistic regression equation can be written as:

$$\text{Log}\,(P\,(\text{Intrusion=1})\,/\,1-P\,(\text{Intrusion=1}))=b_0+b_1X_1+b_2X_2+\ldots+b_nX_n \quad \text{----------(2)}$$

Here:

P (Intrusion=1) is the probability of an intrusion occurring.

b_0 is the intercept term.

$b_1,b_2,\ldots,b_n$ are the coefficients associated with each feature $X_1,X_2,\ldots,X_n$.

The logarithm is the natural logarithm.

The logistic function (sigmoid function) is used to transform the right side of the equation to the range [0, 1]:

$$P\,(\text{Intrusion=1})=1/1+e^{-(b0+b1X1+b2X2+\ldots+bnXn)} \quad \text{----------(3)}$$

This formula calculates the probability of an intrusion event based on the weighted sum of the features, transformed by the sigmoid function. Once the model is trained using historical data, you can use it to predict the probability of intrusion for new instances. Typically, a decision threshold is applied (often 0.5) to classify instances into the two categories (intrusion or non-intrusion).

The coefficients $b_0, b_1,\ldots,b_n$ are estimated during the model training process using techniques like Maximum Likelihood Estimation. The goal is to find the coefficients that maximize the likelihood of observing the given historical data. Keep in mind that this is a basic explanation, and in a practical setting, you would need to consider aspects like data preprocessing, feature scaling, regularization, and model evaluation using metrics such as accuracy, precision, recall, and F1 score.

C. Random Forest

Random Forest [9] is an ensemble learning method that builds multiple decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. In the context of a smart intrusion detection system, you would be dealing with a classification problem where the outcome is binary (intrusion or non-intrusion).

Formula-based Representation: The prediction process involves aggregating the outputs of individual trees. Let's assume we have T trees in the Random Forest.

For a given instance with features $X_1, X_2,\ldots,X_n$, the probability of intrusion can be expressed as:

$$P\,(\text{Intrusion} = 1) = 1\,/\,1+e^{-(1/T\sum_{i=1}^{T}\text{logit}\,(\,\text{Tree}_i\,)}$$

----------(4).

Here:

Logit (Treei) represents the log-odds predicted by the *i*-th decision tree.

The sum is taken over all trees in the Random Forest.

The sigmoid function ($1/(1+e^{-x})$) is used to map the aggregated log-odds to a probability between 0 and 1.

For each tree *i*, the log-odds prediction can be obtained from the decision tree's structure and the values of the input features.

This formula gives you the probability of intrusion based on the aggregated predictions of all the decision trees in the Random Forest. The final classification is usually determined by applying a threshold (e.g., 0.5) to the predicted probability.

In practice, libraries like scikit-learn in Python make it easy to implement Random Forests without directly dealing with the underlying formula. The ensemble approach of Random Forests is powerful for handling complex relationships in data and tends to be robust against overfitting.

D.K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple and effective algorithm for classification, including its application in intrusion detection systems. In the context of smart intrusion detection, KNN [10] can be used to classify network traffic or events based on their similarity to historical data.

Formula-based Representation: Assume you have a dataset with instances represented by features $X_1, X_2, \ldots, X_n$. The distance metric used can be Euclidean distance, Manhattan distance, or others, depending on the specific requirements.

Euclidean Distance Formula:

- The Euclidean distance between two instances *A* and *B* in *n*-dimensional space is calculated as:

$$\text{Distance}(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \text{--(5)}$$

Where $A_i$and B_i are the values of feature X_i for instances *A* and *B*.

Prediction:

For a new instance X_{new}, find the *k* instances from the training set with the smallest Euclidean distances to X_{new}.

Voting:

Assign X_{new} to the majority class among its *k* nearest neighbors.Here's a general representation of the classification decision:

P (Intrusion=1|X_{new}) = $1/k\sum_{i=1}^{K}$ Label(X_i) ---- (6)

Where:

P (Intrusion=1|X_{new}) is the probability of intrusion for the new instance. Label (X_i) is the class label (intrusion or non-intrusion) of the *i*-th nearest neighbor.

The final decision is typically binary, based on a threshold. For example, if the majority class among the *k* nearest neighbors is intrusion for a given X_{new}, then P (Intrusion=1|X_{new}) >Threshold, and the new instance is classified as an intrusion. It's important to choose an appropriate value of *k* and the right distance metric based on the characteristics of your dataset and the problem at hand.

VI. CONCLUSION

Proactive cyber security is exemplified by a Smart Intrusion Detection System Framework with Supervised Machine Learning Methods. This framework attempts to give enterprises an adaptable and effective way to protect against an ever-expanding range of cyber threats by utilizing machine learning. Continuous improvement and modification guarantee the system's resilience to changing security threats. The deployment of a Smart Intrusion Detection System (IDS) framework integrated with supervised machine learning methods represents a cutting-edge and effective approach to bolstering network security. Here are the key takeaways and reflections on the framework.

REFERENCES

[1] M. Wang, K. Zheng, Y. Yang and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems," in *IEEE Access*, vol. 8, pp. 73127-73141, 2020, doi: 10.1109/ACCESS.2020.2988359

[2] Burkart, Nadia, Maximilian Franz, and Marco F. Huber. "Explanation framework for intrusion detection." *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2020*. Springer Berlin Heidelberg, 2021

[3] Patil, S.; Varadarajan, V.; Mazhar, S.M.; Sahibzada, A.; Ahmed, N.; Sinha, O.; Kumar, S.; Shaw, K.; Kotecha, K. Explainable Artificial Intelligence for Intrusion Detection System. *Electronics* **2022**, *11*, 3079. https://doi.org/10.3390/electronics11193079

[4] Marwa Keshk, Nickolaos Koroniotis, Nam Pham, Nour Moustafa, Benjamin Turnbull, Albert Y. Zomaya,An explainable deep learning-enabled intrusion detection framework in IoT networks,Information Sciences,Volume 639,2023,119000,ISSN 0020-0255, https://doi.org/10.1016/j.ins.2023.119000.

[5] Yasunobu Nohara, Koutarou Matsumoto, Hidehisa Soejima, Naoki Nakashima,Explanation of machine learning models using shapley additive explanation and application for real data in hospital,Computer Methods and Programs in Biomedicine,Volume 214,2022,106584,ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2021.106584

[6] Elrawy, M., Awad, A. & Hamed, H. Intrusion detection systems for IoT-based smart environments: a survey. *J Cloud Comp* **7**, 21 (2018). https://doi.org/10.1186/s13677-018-0123-6

[7] Song YY, Lu Y. Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry. 2015 Apr 25;27(2):130-5. doi: 10.11919/j.issn.1002-0829.215044. PMID: 26120265; PMCID: PMC4466856.

[8] Wright, R. E. (1995). Logistic regression. In L. G. Grimm & P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics* (pp. 217–244). American Psychological Association.

[9] Biau, G., Scornet, E. A random forest guided tour. *TEST* **25**, 197–227 (2016). https://doi.org/10.1007/s11749-016-0481-7

[10] Kramer, O. (2013). K-Nearest Neighbors. In: Dimensionality Reduction with Unsupervised Nearest Neighbors. Intelligent Systems Reference Library, vol 51. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2