A Robust Network Intrusion Detection System Based on Machine-Learning Models with Early Classification

1st BHANUPRAKSHREDDY SURAKANTI
Department of CSE
Parul University
Vadodara, India
bhanuprakashreddysurakanti@gmail.com

2nd SANDEEPREDDY VADICHARLA
Department of CSE
Parul University
Vadodara, India
sandeepreddyvadicharla55@gmail.com

3rd SHARATHCHANDRA SAMINENI
Department of CSE
Parul University
Vadodara, India
sharathsamineni2003@gmail.com

4th DEVASENAN RAMU
Department of CSE
Parul University
Vadodara, India
devasena58948@gmail.com

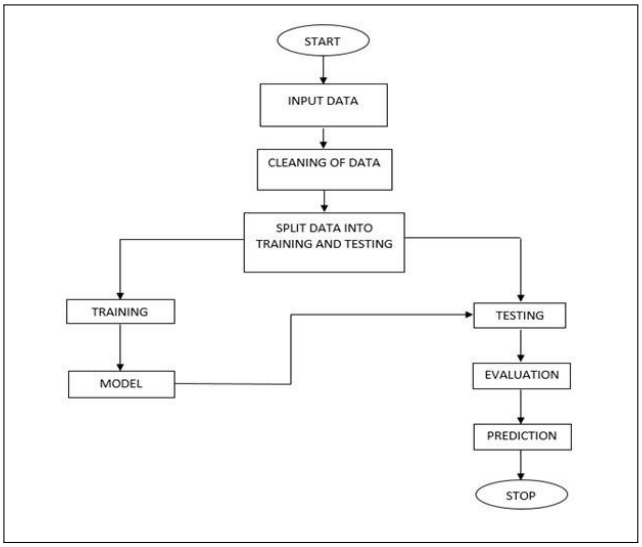
5th Asst.Proff PIRMOHAMMED KHAN
Department of CSE
Parul University
Vadodara, India
Sheikh.pirmohammad@gmail.com

***Abstract*—Abstract—Network Intrusion Detection Systems (NIDSs) using pattern coordinating have a lethal weakness in that they cannot detect new assaults since they as it were learn existing patterns and use them to identify those assaults. To fathom this issue, a machine learning-based NIDS (ML-NIDS) that identifies anomalies through ML algorithms by analyzing behaviors of conventions. However, the ML-NIDS learns the characteristics of assault trafiic based on preparing information, so it, as well, is unavoidably powerless to attacks that have not been learned, fair like pattern-matching machine learning. In this manner, in this ponder, by analyzing the characteristics of learning utilizing agent features, we appear that network intrusion exterior the scope of the learned information in the include space can bypass the ML-NIDS. To avoid this, planning the active session to be classi ed early, some time recently it goes exterior the detection range of the preparing dataset of the ML-NIDS, can effectively prevent bypassing the ML-NIDS. Different tests con rmed that the proposed strategy can identify interruption sessions early (before sessions end) signi cantly progressing the robustness of the existing ML-NIDS. The proposed approach can provide more strong and more precise classi cation with the same classi cation datasets compared to existing approaches, so we anticipate it will be utilized as one of attainable arrangements to overcome weakness and impediment of existing ML-NIDSs Index Terms—Network intrusion detection, early classification, robust classification, adversarial assault, machine-learning**

***Index Terms*—Network intrusion detection, early classification, robust classification, adversarial attack, machine-learning**

I. INTRODUCTION

In this modern world dominated by technology, it is important to recognize and handle network breaches as quickly and precisely as possible to preserve the functionality and health of the systems and networks. For this purpose, adedicated security device called the Network Intrusion Detection System (NIDS) was initiated[1] . These are widely used to monitor and analyze network traffic . So Traditional NIDSs are primarily



h

Fig. 1. Proposed ML-NIDS Framework

based on pattern matching mechanisms which are detecting only known attacks and failed to recognize new threats.[]

To solve this problem, various methods have been proposed and applied to the NIDS [3]. Recently Machine learning based network intrusion has received most attention which can improve alternative solutions for PM-NIDS . The ML-NIDS analyzes the characteristics of existing network intrusions using ML and detects the intrusions using overall behavioral characteristics.[5] More like PM-NIDS ,ML-NIDS are overly reliant on patterns which are present in the dataset that means thes ability of detection is directly influenced by quality of data used for training . This process creates a limitation for

ML-NIDS which fails to detect novel intrusions which are not present in the datset .

Despite of this limitation, research in the field overlooks for this aspect. Instead, most studies focus on ways for bypassing ML-NIDS by changing features in the data. Recently some have come up with enhancing the robustness of the training datasets using Generative Adversarial Networks (GANs) or other deep learning techniques. However, these strong matches do not directly study the fundamental issue of the system dependence on the training dataset, leaving a gap in understanding how this dependency affects the detection capabilities.

So this paper aims to address this gap by probing the characteristics of ML-NIDS reliance on trainig dataset . The approach which we proposed enhances the intrusion detection performance without materially increasing the dataset size.

The main commitment of this study are : we showed that ML-NIDS is susceptible for detecting intrusions when slight changes are made to the behavior of the attack, that we added extra data packets.Moreover the effect of this dependency can differ depending on the Secondly we started a method that refines the detection of intrusions by adjusting the packet count. This method ensures that even short or long sessions, which traditional ML-NIDS may struggles to identify can now be detected with by improving accuracy. Finally, that the proposed method was lightweight which is enough to be implemented into the current NIDS-ML , rather than needed expensive,high-performance hardware and it is economically viable .

II. Literature Review

A. Summary of Previous Studies

Various methodologies have been studied by researchers to develop an efficient ML-based NIDS. These studies show several concerns for improving detection accuracy, minimizing false positives, and updating the strengths of network threats into practice.[6]

B. Machine Learning in NIDS:

The application of Machine Learning in NIDS is rapidly changing how cyber attacks are identified and dealt with.[7] For instance, there has been some work done towards the design of Intrusion Detection Systems that work with Machine Learning Algorithms. This was done by Yang et al using an NIDS model where actions are classified with the aid of deep learning. While the model had its weaknesses such as being prone to adversarial attacks, it also exhibited some promising patterns for detection. In another research, Zhang et al has proposed a new approach, which is extremely modern, by merging SVM and Deep Neural Networks. Even though this model managed to reduce the false positive detection rate, it required a vast amount of computational resources.[8]

C. Unsupervised and Deep Learning Approaches:

With the unsolved attack multi-faceted problems, researchers are adapting the use of deep learning techniques to promote enhanced usage of machine learning techniques. Autoencoders and Variational Autoencoders (VAE) - these systems uncover baseline traffic patterns and capture network anomalies by reconstructing them.

Recurrent Neural Networks (RNNs) And Long Short Term Memory (LSTM) - These techniques allows for the capturing of network traffic for a specific period of time, enabling the real-time detection of threats.

Generative Adversarial Networks (GANs)[9] - Traffic of the generated attack model is injected into the neural network to increase robustness of model

Park et al. Proposed an anomaly detection system based on generative adversarial networks which enhanced generaliztion to other unknown threats[10]. Likewise, Wang et al.[11] advanced a Packet Bytes Based CNN (PBCNN), which outperformed traditional models[11] in accuracy for packet level classification.

D. Challenges in Network Intrusion Detection:

The Problems of Network Intrusion Detection Systems: With more sophisticated cyber threats, there are more problem areas for machine learning, or ML, to tackle.

Sustaining High False Positives: Maintaining High, Yet Misguided Alerts: Elevated computing data inferences dismisses the accuracy of the information on hand, and in turn, floods the system with alerts about potential threats.

Analyzing Ecrypted Traffic: With the possibility of analyzing encrypted network traffic data no longer there, older models become much less effective..

Adversarial Evasion: The omnipresent innovativeness of cyber terrorists enable them to bypass detection rates by using sample data while still obfuscating their real information.

Resource Constraints: Getting thrown into a real life scenario requires a complete replacement of the ML based NIDS, which elevates the problems in computation.

E. Other Solutions:

The inquiry is focused toward mixed methods, but the comprehensiveness of results from ML-based NIDS is rather poor. NIDS based on federated learning was suggested by Wang et al.[3] that might significantly improve the security of distributed networks with a guarantee of privacy. However, this model came with issues – as pointed out by Sing et al.[4] – its heterogeneity of datasets creates many challenges. Singh, et al.[4] showed how robust reinforcement learning detection policies enable on-the-fly decision-making, but fail to consider deployment in real-time.

F. NIDS's Early Detection Classification

Real Time Packet Inspection: Network packets are checked at several different stages of the network using ML models.

Threshold Based Classification: When an anomaly is detected at the beginning of packet capture, the session is marked of further inspection.

Adaptive Learning: The understanding and classifying criteria is improved with active traffic monitoring.

With these changes, our approach allows for enhanced responsiveness to cyber threats with no compromise to network security.

III. Problem Formulation

Network intrusion detection system(NIDS) is a key element of in todays world security systems, but normal methods such as signature-based detection or anomaly-based detection are often faced limitations in dealing with new or evolving cyber threats. These kind of systems are struggling to detect zero-day attacks, suffering from high false positive rates, and often demanding. In specific, the failure to detect intrusions at an early stage can lead to overall damage, as attackers can bypass the detection systems before their activities are fully noticed. In this conditions machine learning(ML) based network intrusion detection system have obtained attention beacuse of the ability to adapt and learn from data to new attacks. The detection of threats in the incipient phase will have a tremendous impact on the quickened response time and minimal damage caused by these intrusions. Consequently, this study outlines the major problem to be addressed:

How can we use machine learning algorithms to both detect network intrusion with greater accuracy and then do it early enough so that the damage can be minimized, false positives reduced, and operational efficiency improved?

For that we proposed a solution which involves in voting ensemble method by combining different machine learning algorithms like Random forest , XGBOOST ADABOOST, Decesion tree to gain the best accuracy and good performance in detection.

1. Adaptability: Use of machine learning models can allow the system can easily emerge attack stratergies by continuously learning and improving its detection capabilities without need of manual updates.
2. Reduced False Positives : This system mainly focuses on reducing false positives to improve system performance.
3. Early Threat detection : This Early classification enables the detection of an attack at it early stages and reducing and mitigating the threats to the systems.
4. Proactive defence: This system gives a proactive defence by analyzing new threats rather than not responding to attacks before they damage the systems.

This mechanism strengthens the overall security of an organization.

IV. System Design

A. System Architecture

The system conforms to a pipeline with a specific sequence of model creation[12], testing, training of data, and output creation. User input in terms of text is processed through the system in terms of a trained model of a machine algorithm. For training, a dataset is taken, and subsequently, testing, meaningful output is generated through the system. With this, effective processing and proper prediction is assured.[13]

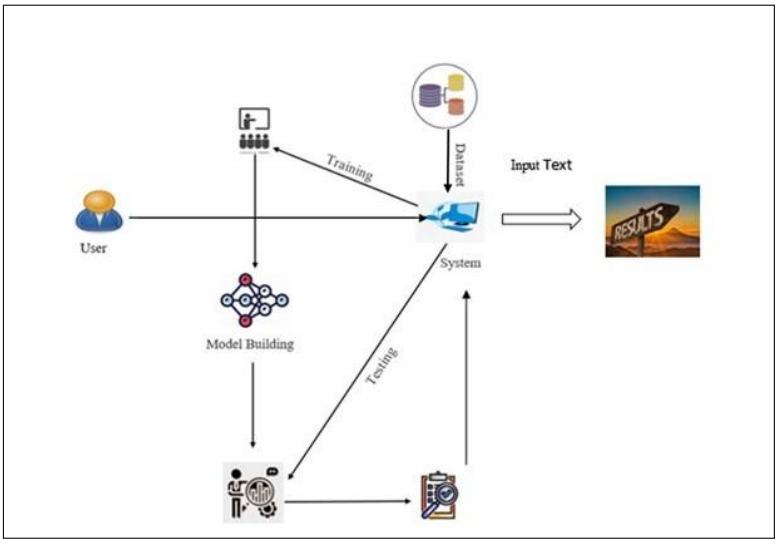


Fig. 2. System Architecture

B. UML Diagrams

1. Use Case Diagram

The use case diagram which is part of Unified Modeling Language (UML) is a behavioral diagram which is developed and created from a use-case analysis. Its primary purpose is to represent visually the fuctions of the system from the viewpoint of actors. It shows the purpose of these actors as they use cases which shows any dependencies between them. so the primary goal of an use-case diagram is to display which functions of system is performed for each actor and how they carry out.

2. Activity Diagram

Main purpose of Activity diagrams are used to represent workflows of sequential order,supporting concurrencyand choices . Within the context of UML,activity diagrams shows the opeartional and businesses workflow of items in a system.As of following These diagrams also shows the flow of control of components in various systems.

3. Component Diagram

The component diagram which is commonly known as a UML component diagram that shows the organization and interconnection of the physical components within the system. Basically these diagrams are primarily used to model implementation of a system and and verify that all necessary functions are included in the plan of development. when it comes to physical implementation component diagram plays a crucial role.

V. Modules

A. System:

1 .Store Dataset

The system allows users to upload and store the dataset containing network traffic data. This dataset serves as the base

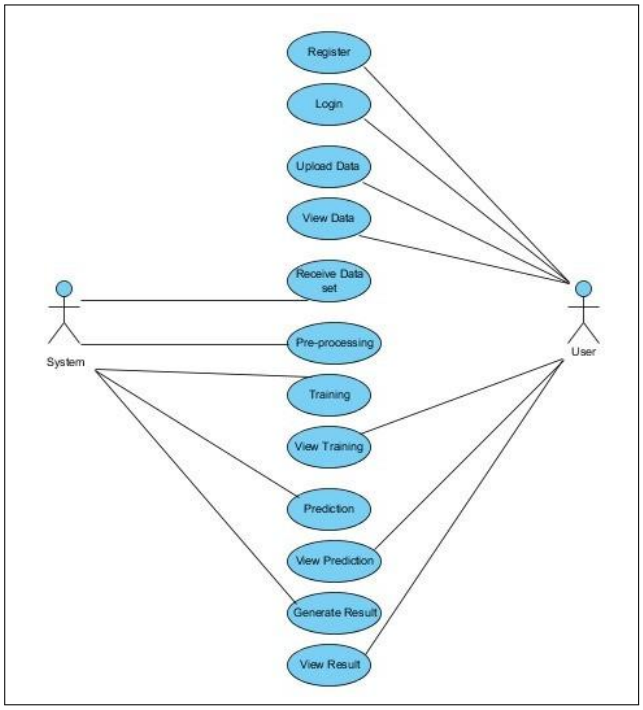


Fig. 3. Use Case Diagram

for training and testing machine learning models.

2 .Model Training

Machine learning models require data for training. In this module, the system takes the user-provided dataset and feeds it into the selected model. The model learns from the dataset to recognize patterns and anomalies in network traffic.

3 .Model Prediction

Model Prediction The system classifies and predicts the type of network traffic: normal or harmful based on the trained dataset and machine learning model. This classification is based on the information obtained from training.

4 . Data Splitting

With the objective of improving the model's robustness and accuracy, the system splits the dataset into two parts: a training set and a testing set. This split allows evaluation of the model using data that was not available during training which ensures that the model's performance is measured beyond the training phase.

B. User :

1.Registration

The users can easily register by providing the required information which can be utilized by users in subsequent registrations.

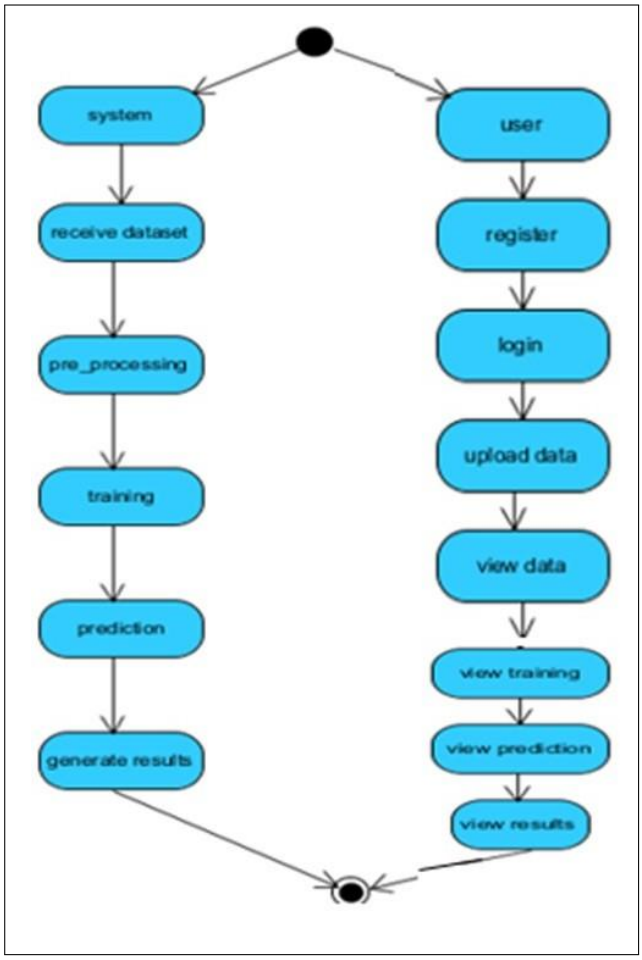


Fig. 4. Activity Diagram

2.Login

registered user will be required to enter his/her credentials and an appropriate password for the various system functionalities. The issued task is how to choose an appropriate model to get correct results in the right way.

3.Load Dataset

The users are allowed to upload the specific dataset which they want to work on, providing them the needed freedom and flexibility in their intrusion detection strategies.

4.View Dataset

This module enables users to look at their uploaded dataset. This capability is especially relevant for understanding the nature of the data and inspecting what they wish to work on.

5.Select Model

Readers may pick a certain model from a number of dreadful choices that captures the features specific to their dataset for detecting an intrusion. Choosing an appropriate model to accomplish the task is of great importance for getting correct results.

6. View Results

Users can view system generated model prediction results regarding whether there is an attack on the network or not. These results are information that can be used in making security decisions.

VI. METHODOLOGY

We developed our approach that enhances ML-NIDS by introducing Early Classification Framework to detect intrusions before they bypass security measures .

A. Feature Selection:

Knowing that feature selection Play an important in identifying most relevant network traffic characterstics that are contributing to detection of intrusions.

- **Statical Extraction :** This feature part includes such as packet size, port number, received packets, flow duration and protocol frequency to differentiate between normal and attack traffic.
- **Deep learning-Based Extraction** This Feature includes deep neural networks like ANN and CNN to enhance anomaly detection.

B. Algorithm Selection

The following supervised machine learning classifiers were used for detecting and classification:

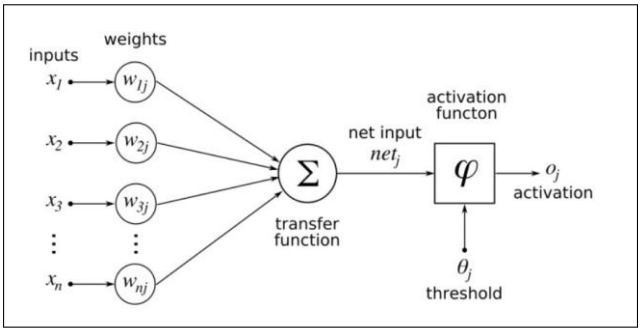
1. Decision Tree : Decision tree which is also known as choice tree plays an important role in various applications. It is relatable to a person making a decision and it is of great significance in machine learning as far as the tasks of classification and regression are concerned. In decision analysis, choice trees are a methodical computer aided way of representing all decisions together with their reasoning. As the name suggests, they have a plant like segmented appearance for making choices. They are common in data science, particularly in information mining in formulating the techniques of how to achieve certain goals. Real life datasets are different and every feature is one of the sets of information that are analyzed to make decisions. Decision trees are overly simple yet effective. Their strength relates to an individual's ability to absorb different sets of information. This type of approach is well known to assist learning on decision trees. For example, a classification tree is created for classification tasks such as finding out if the passengers in the data set survived or did not survive. On the other hand, regression trees are used to forecast the value of a variable for instance prices of houses. So, how is a decision tree made? To develop the decision tree, the relevant features that are most significant are picked out and conditions that would help in splitting the data are created. Furthermore, one of the most critical features of tree construction is being able to know when it is time to stop further splitting. As trees grow naturally, there are pruning techniques that are used in order to improve their structure and prune them for accuracy. Decision trees are one of the most relevant methods in business intelligence, as they enable classification and forecasting in an orderly manner. Now, let us look at some popular methods of their construction.

2. Random Forest : .Arbitrary Woodland : A arbitrary woodland is a learning algorithm for classification and regression tasks. It functions by ensemble learning using multiple classifiers to solve difficult problems in a more efficient way. In predicting outcomes, a random forest model contains multiple decision trees that work together to provide the most accurate forecast. The Random Forest algorithm builds a forest by means of bagging, also known as bootstrap aggregating. An ensemble method called bagging is used to increase the performance of predictive models by training several models on different portions of the data. The random forest algorithm relies on predicting using several decision trees. The predictions of these so called trees are summed up in order to get a final answer. For every tree in a forest, there needs to be a considerable amount of information which needs to be processed in order to differentiate one from the other. Random Tree is less critical. A decision tree on the other hand works by determining root nodes and determining all other splits and submodules based on the predetermined structure. Random forests do not implement this criterion. Instead, they take twigs which are picked at random. The criteria of bagging, which means splitting the dataset and training multiple models simultaneously instead of training on a single set is implemented in a random forests algorithm. Through features and observations, one is able to build better predictions. Each tree in a random forest processes information from a different subset of data therefore result in one final answer is determined. These predictions are then calibrated by voting, and the end result is the winner or average of votes.

3 . XGBoost :XGBoost or extreme gradient boosting is among the most powerful and sophisticated machine learning libraries for performing gradient boosting as it tries to parallelize the process of tree boosting. This process works best in multi-dimensional data science problems as it helps enhance models as well as solve the problems at hand effectively and swiftly. Boosting is a sequential ensemble type learning technique where multiple weak classifiers are combined to achieve high accuracy. Unlike bagging methods that aim to reduce variance in models, boosting attempts to resolve both bias and variance and thus, yields better results. One more reason why Xgboost is unique is the parallel and distributed processing, which makes it faster than many boosting algorithms. Hence, it is widely used for many processes because of its advanced performance in optimizing the models of gradient boosting.

4 . ANN : AAn Artificial Neural Network (ANN) is a neural network model, with lots or very few layers of processing units called neurons. Each of these layers is connected to another layer by nodes known as layers (input, hidden, output). These nodes take a particular set of data, some specific structures depending on certain parameters or values, transforming that data into an inner weighting system. To process data in a particular specified manner, these structures and shapes are needed to enable the unit to provide a certain amount of information as output. An ANN is then used to compute details from a presented dataset by providing a specific desired

output. When ordering or categorizing information, people need rules, restrictions and processes, similarly they set rules to avoid challenges like hurdles in summing up or processing the entire data set. To control data output, ANN sets rules referred to as backpropagation; this system assists the unit in processing where using labeled training data reduce the generalization errors by adjusting neural connections known as neurons storing memories, which aids in providing the desired outcome. To put this into an example, initially like every other neural network model at first undergoes an active and learning stage where they try to distinguish patterns which can be visual, audio or text, the network uses supervised learning patterns or a classification algorithm. During this supervised pattern its real yield is produced with what it was meant to produce—the yield that was wanted. The difference between both outcomes is balanced by the fact of using most reduced conceivable blunder.



5 . CNN : Conventional ML models like Irregular timberland requore future designing , where specialists are physically extractingn features of arrange like information parcel estimate and stream length etc . Be that as it may CNNs can naturally design learning network and extraction of highlights from crude information without manual intervention. Following, we will apply a Pooling layer to our Convolutional layer, so that from each highlight outline we create a Pooled include outline as the primary reason of the pooling layer is to make beyond any doubt that we have spatial invariance in our images. It moreover makes a difference to decrease the measure of our pictures as well as avoid any kind of overfitting of our information. After that, we will flatten all of our pooled pictures into one long vector or column of all of these values, taken after by contributing these values into our artificial neural organize. t can be characterized as a standard neural network layer that gets an input from the going before layer followed by computing the course scores and comes about in a 1- Dimensional cluster that has the rise to estimate to that of the number of classes.Lastly, we will nourish it into the locally connected layer to accomplish the last yield. Here we utilized 2D-CNN for image based discovery where utilizing pooling layers , the pooling layer is to make beyond any doubt that we have spatial invariance in our images. It moreover makes a difference to decrease the estimate of our pictures as well as maintain a strategic distance from any kind of overfitting of our information. After that, we will flatten all of our pooled pictures into one long vector or column of all of these values, taken after by contributing these values into our counterfeit neural arrange. Finally, we will bolster it into the locally associated layer to accomplish the last output.In case some certain introduction edges are display at that point as it were a few individual neuronal cells get let go interior the brain such as a few neurons responds as and when they get uncovered to the vertical edges, however a few reacts when they are appeared to level or diagonal edges, which is nothing but the inspiration behind Convolutional Neural Systems. For the usage of CNN, we are going to utilize the Jupyter scratch pad. So, we will start with bringing in the libraries, information preprocessing followed by building a CNN, preparing the CNN and in conclusion, we will make a single expectation. All the steps will be carried out in the same way as we did in ANN, the as it were distinction is that presently we are not pre-processing the classic dataset, but some pictures, which is why the information preprocessing is different and will comprise of doing two steps, i.e., in the to begin with, we will pre-process the preparing set and at that point will pre-process the test set.

C. Early Classification Mechanism

It ensures that some potential intrusions are are flagged before the session is going to complete, which may reduce the risk of attacker successfully evading detection . we used this to reduce false negatives which makes the system more Robust. If the anomaly is detected in early packets then moves towards for further analysis. ML models classifys network packets at multtiple checkpoints. so , Early classification ensures that system becomes more robustness with threshold-based classification,this approach improves accuracy and speep-up response time and strengthen networks.

VII. Implementation and Results

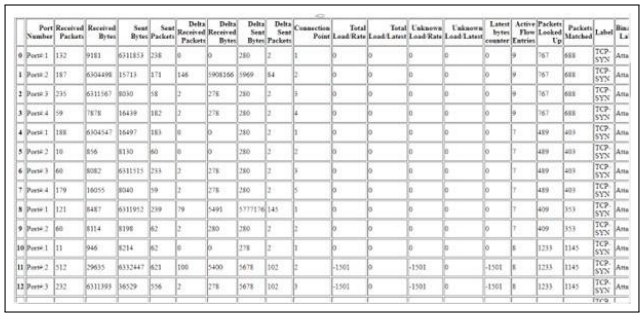
The proposed NIDS-ML system is developed to enhance and detect real-time intrusions using machine learning models This implementation process involves collecting Datasets, model Training, System Developement and Perfromance evaluation. This system used Datasets which are publicly available and deployed in web-based interface which allows the users to detect threats.

A. Dataset Collection

The Dataset which we used in this study is ISCX2012 a well-known dataset which contains labeled network traffic data.[14] This dataset include both benign (normal) and malicious (attack) network traffic,that allow machine learning models to learn distinct patterns. This dataset consists of multiple attack vectors that are Dos-attacks , brute-force attacks and scanning attacks.

B. Model Training and Performance

Machine learning models were trained using ISCX2012 data that are Decision Tree , Random forest , XGBoost,Adaboost , ANN and CNN.[15]



	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Delta Received Packets	Delta Received Bytes	Delta Sent Bytes	Delta Sent Packets	Connection Point	Total Load/Rate	Total Load/Latest	Unknown Load/Rate	Unknown Load/Latest	Latest bytes counter	Active Flow Entries	Packets Looked Up	Packets Matched	Label	Rin La
0	Port#1	132	9181	6311853	258	0	0	280	2	1	0	0	0	0	0	9	767	688	TCP SYN	Ana
1	Port#2	187	6304498	15713	171	146	5908366	5969	84	2	0	0	0	0	0	9	767	688	TCP SYN	Ana
2	Port#3	235	6311967	8030	98	2	278	280	2	3	0	0	0	0	0	9	767	688	TCP SYN	Ana
3	Port#4	59	7878	16439	182	2	278	280	2	4	0	0	0	0	0	9	767	688	TCP SYN	Ana
4	Port#1	188	6304547	16497	183	0	0	280	2	1	0	0	0	0	0	7	489	403	TCP SYN	Ana
5	Port#2	10	856	8130	60	0	0	280	2	2	0	0	0	0	0	7	489	403	TCP SYN	Ana
6	Port#3	60	8082	6311515	233	2	278	280	2	3	0	0	0	0	0	7	489	403	TCP SYN	Ana
7	Port#4	179	16055	8040	59	2	278	280	2	5	0	0	0	0	0	7	489	403	TCP SYN	Ana
8	Port#1	121	8487	6311952	239	79	5491	5777176	145	1	0	0	0	0	0	7	409	353	TCP SYN	Ana
9	Port#2	60	8314	8198	62	2	280	280	2	2	0	0	0	0	0	7	409	353	TCP SYN	Ana
10	Port#1	11	946	8214	62	0	0	278	2	1	0	0	0	0	0	8	1233	1145	TCP SYN	Ana
11	Port#2	512	29635	6332447	623	100	5400	5678	102	2	-1501	0	-1501	0	-1501	8	1233	1145	TCP SYN	Ana
12	Port#3	232	6311393	16529	356	2	278	5678	102	3	-1501	0	-1501	0	-1501	8	1233	1145	TCP SYN	Ana

Fig. 5. Network Traffic Flow

To compare the efficiency of these models following performance metrics are used :
Accuracy : Basically accuracy measures overall correct predictions.
precision: It measures correct predictions made for intrusions alongside all other predicted intrusions.
Recall : This section of Recall determines the effectiveness of the model in recognizing actual intrusions.
F1-score: This section checks whether the balance between recall and precision is maintained for evaluation or not.

C. Experimental Results

The demonstration of these models in following insights :
Random Forest and XGBOOST stand out as the two models with the highest detection rates compared to other models against different types of attack patterns.
Decision Tree and ADABOOST had provided fast inference times but less accurate in finding intrusion patterns.[16]

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree (DT)	89.4	87.2	85.6	86.4
Random Forest (RF)	94.2	92.8	91.5	92.1
XGBoost	95.1	93.7	92.3	93.0
AdaBoost	91.8	89.5	88.2	88.8
Artificial Neural Networks (ANN)	92.6	90.3	89.1	89.7
Convolutional Neural Networks (CNN)	93.5	91.0	90.5	90.7

TABLE I
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS

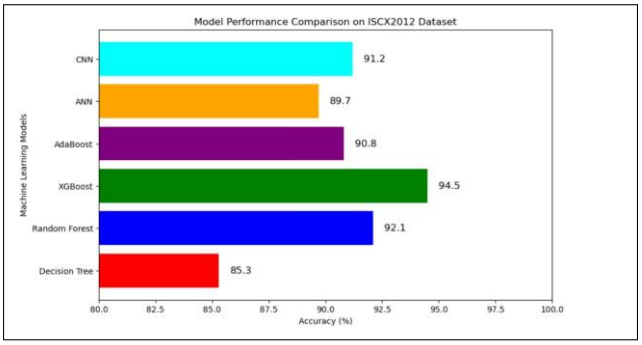


Fig. 6. Model Performance Comparison on ISCX2012

These results showing that effictivness of deep learning models in NIDS offers an efficient accuracy.

D. Web Application Integration

The uploaded models were implemented on a web-based application where the users can classify network traffic in real-time. The interface of the application contains:
Data Set Uploading - Allows users to upload network traffic data from ISCX2012 database for analysis.
Model Selection - Selection of various models of machine learning.
Live Threat Detection - Indicates if the incoming traffic is normal or it is malicious.

Fig. 7. Web Interface for Intrusion Detection

VIII. CONCLUSION AND FUTURE WORK

The most important aspect with regards to the ML-NIDS is the pre-training set utilized to capture the classifier exhibit. But that, it is unrealistic to obtain a preparing dataset that includes all possible network intrusions that happen in the wild. It is crucial to integrate other segments of the data and determine how best to classify an intrusion even if the dataset does not contain enough intrusion data. This paper will discuss a new way rest of the problem. While using different datasets, the proposed approach was able to show that the inadequacies of the pre-existing ML-NIDS systems can be greatly improved within. Undoubtedly, this newly proposed method has its own show of drawbacks. For example, it may not be adequate to decide whether the learning range is exceeded by using just the feature of forward packet counting. However, other ways of determining such make the use of transforms more direct to understand. If you require further assistance, we have our support team available 24/7 who will be carefree to help with you any questions or queries you may have. The number of sessions that can be accomplished at any given moment is limited, though the prospects of enhancing the location rate for some classes are rather low. Nevertheless, the option to flexibly broaden the classification run in the feature space by utilizing a dataset of constrained information is truly remarkable in its benefits. Moreover, classification speed enhancement is also possible, which gives an indication that the proposed strategy, once its real NIDS hardware is integrated, will greatly aid in securing large networks. In the next step of our research, we will focus on how this method can be implemented to accommodate more features. If successful, it will provide the ML-NIDS with the capability to maintain a high classification

discovery rate without too much strain on the classification speed. The other major advancement can incorporate real-time threat mitigation capabilities. Besides early intrusion detection, the system can proactively isolate, contain, or block suspicious network traffic. There should also be provisions of further strengthening security measures by allowing the system to incorporate automated responses against identified threats as well defined within a certain set of user policies. Automating responses with user-defined policies can greatly decrease the magnitude of the damage caused by attacks.

REFERENCES

[1] A. Borkar, A. Donode, and A. Kumari, "A survey on intrusion detection system (IDS) and internal intrusion detection and protection system (IIDPS)," in Proc. Int. Conf. Inventive Comput. Informat. (ICICI),Nov. 2017, pp. 949–953, doi: 10.1109/ICICI.2017.8365277.

[2] Z. Zhou, C. Zhongwen, Z. Tiecheng, and G. Xiaohui, "The studyon network intrusion detection system of snort," in Proc. Int. Conf.Netw. Digit. Soc., May 2010, pp. 194–196, doi: 10.1109/ICNDS.2010.5479341

[3] M. F. Zolkipli and A. Jantan, "A framework for malware detection using combination technique and signature generation," in Proc.2nd Int. Conf. Comput. Res. Develop., May 2010, pp. 196–199, doi:0.1109/ICCRD.2010.25.

[4] H. Zhang, "Design of intrusion detection system based on a new patternmatching algorithm," in Proc. Int. Conf. Comput. Eng. Technol., Jan. 2009,pp. 545–548, doi: 10.1109/ICCET.2009.244.

[5] V. Gupta, M. Singh, and V. K. Bhalla, "Pattern matching algorithms forintrusion detection and prevention system: A comparative analysis," inProc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Sep. 2014,pp. 50–54, doi: 10.1109/ICACCI.2014.6968595.

[6] A. Halimaa A. and K. Sundarakantham, "Machine learning basedintrusion detection system," in Proc. 3rd Int. Conf. Trends Electron.Informat. (ICOEI), Apr. 2019, pp. 916–920, doi: 10.1109/ICOEI.2019.8862784

[7] M. Green and L. White, "Advancements in Training and Deployment Strategies for AI-Based Ransomware Detection," *Journal of Information Security*, vol. 15, no. 1, pp. 67-80, Jan. 2025.

[8] A. Phadke, M. Kulkarni, P. Bhawalkar, and R. Bhattad, "A review ofmachine learning methodologies for network intrusion detection," inProc. 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC), Mar. 2019,pp. 272–275, doi: 10.1109/ICCMC.2019.8819748.

[9] L. Bondan, M. A. Marotta, M. Kist, L. R. Faganello, C. B. Both,J. Rochol, and L. Z. Granville, "Kitsune: A management system forcognitive radio networks based on spectrum sensing," in Proc. IEEENetw. Operations Manage. Symp. (NOMS), May 2014, pp. 1–9, doi:10.1109/NOMS.2014.6838316

[10] R. Gaddam and M. Nandhini, "An analysis of various snort based techniques to detect and prevent intrusions in networks proposal with coderefactoring snort tool in kali Linux environment," in Proc. Int. Conf.Inventive Commun. Comput. Technol. (ICICCT), Mar. 2017, pp. 10–15,doi: 10.1109/ICICCT.2017.7975177

[11] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu,"HAST-IDS: Learning hierarchical spatial-temporal features using deepneural networks to improve intrusion detection," IEEE Access, vol. 6,pp. 1792–1806, 2018

[12] C. Park, J. Lee, Y. Kim, et al., "An enhanced AI-based network intrusion detection system using GANs," IEEE Internet of Things Journal, vol. 10, no. 3, pp. 2330–2345, 2022.

[13] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Comput. Secur., vol. 31, no. 3, pp. 357–374, May 2012, doi: 10.1016/j.cose.2011.12.012.

[14] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generat- ing a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116, doi: 10.5220/0006639801080116

[15] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "A hierar- chical hybrid intrusion detection approach in IoT scenarios," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020, pp. 1–7.

[16] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape´, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," IEEE Trans. Netw. Service Manag., vol. 16, no. 2, pp. 445–458, Feb. 2019.