ADVANCED NETWORK ATTACK DETECTION USING HYBRID FEATURE EXTRACTION AND MACHINE LEARNING TECHNIQUES

Ravi Raja Yadlapalli, M.Tech. Student, VFSTR, AP ravirajayadlapalli1995@gmail.com
Dr.K.B.Manikandan, Assistant Professor, VFSTR, AP sansuman612@gmail.com

ABSTRACT

Cyber-attacks are drifting more and more over network security, making compelling intrusion detection systems (IDS) harder to send. SVM and Naive Bayes algorithms, which provided reasonable accuracy, are part of the current system. The proposed work implements two hybrid models, where the first one comprises of Random Forest and AdaBoost and second one comprises of LGBM + MLP + RF + XGB. The Voting Classifier is applied on the first hybrid model, which will do the soft voting, and the Stacking Classifier is applied on the second hybrid model. These two hybrids perform very well in the vectorisation of safety assaults. The idea looks to give an intrusion detection system (IDS) that can rapidly and precisely associate organisation dangers continuously with ML procedures prepared on the NSL-KDD dataset. It will likely further develop network security proactively. To distinguish significant attributes, the plan goes back over the NSL-KDD dataset. This technique trains ML models to arrange typical motions and assault designs in parallel. After exhaustive testing, the first hybrid model (RF+AdaBoost) with the help Voting Classifier beats ordinary methodologies with extraordinary delicacy and progressively attacks identification. The NSL-KDD dataset is utilized to investigate network traffic information utilizing Machine learning (ML), which extraordinarily improves interruption discovery capacities. This work stresses how significant ML strategies and organization traffic investigation are to upgrading digital protection. Touchy information and construction are gotten, and the fabricated IDS gives promising outcomes for groundbreaking protection from arising digital risks.

KEYWORD: machine learning, feature selection, accuracy intrusion detection, network attacks Binary classification.

INTRODUCTION :

The internet is currently utilized for a large number of purposes in day-to-day existence. Availability to the web is extremely useful and offers numerous open doors for associations. Nonetheless, it additionally presents difficult issues and security gambles. Intruders are constantly attracted by significant data, which makes them defenseless to organize interruptions. Entering the framework without authorization is called interruption. It's hard to distinguish a bushwhacker dependent just upon its working framework, application, or IP address. The safeguards a chief takes to prevent programmers from forestalling admittance to data are referred to as organize security. To recognize attacks on track frameworks that are open, a kind of organization security innovation known as an intrusion discovery system (IDS) [17] is utilized. There are two kinds of intrusion discovery System. They are both predicated, independently, on inconsistencies and marks. IDS utilizes a mark to distinguish known assaults. Irregularity IDS is utilized to recognize surprising assaults [17]. Scientists have made progress toward an answer by carrying novel ways to ML Three categories of machine learning models exist: supervised learning, unsupervised learning, and reinforcement learning [12, 17, 18, 19, 22].In supervised learning, the arrangement issue is handled. To prepare the model, otherwise called the classifier, it utilizes a named dataset. Information is checked by grouping calculations, which likewise give a basic principle that can be utilized for naming (planning) recently made input vectors. Therefore, a part of unsupervised learning is grouping. Inside the preparation information, it tracks down designs. It is a regularly held supposition that the bunches will adjust very well to an instinctive order of test gatherings. The last strategy depends on an experimentation approach and is called reinforcement learning. Consequently, the model is allowed to direct the information; it is compensated for accurately doing the mentioned activity and punished for erroneously performing it. All calculations, in the interim, enjoy benefits and weaknesses of their own. In our article, we decided to utilize a calculation that works best when joined with half and half component determination methods. We utilize the most utilized NSL-KDD dataset (3, 5), for preparing. The voting classifier is applied to the hybrid model which comprises of Random Forest and AdaBoost shows the best accuracy results under rigorous training and testing, which beats the current systems in accuracy metric till date.

LITERATURE SURVEY :

Robotic assaults are turning out to be more regular and more unbending. Thus, intrusion detection systems, or IDSs [17], have become a critical part of an organization's security system. Various calculations have been utilized to accomplish abnormal findings. They made huge allowances utilizing artificial intelligence (AI). In this manner, bringing down the misleading negative and working on the nature of finding are the essential objectives. Be that as it may, eliminating handling time is critical. The education and test steps of the IDS can be finished with different informational collections from the writing. Choosing the critical components of the as of late named Information Disclosure in Data sets Informational index (NSL-KDD)(3, 5) that have the most bearing on the revelation's result is one of the objectives of this work. Thus, we will utilize the dataset's agitating part. For our Organization IDS (NIDS) to be understood, as our underlying system, we carried out the Condensed Nearest Neighbors (CNN) calculation in F.Z. Belgrana et al. (2021) [1]. A truly productive technique for relapse and grouping that considers test dissemination. CNN keeps up areas of strength for with results while decreasing the information vector aspect, which thusly yields in diminished framework asset utilization and handling time reserve funds. As a contingency plan, we proposed utilizing a Neural Network (NN) to preclassify our proficiency informational collection. We give an examination of our strategies interestingly, with K Nearest Neighbors (KNN) procedures to exhibit their adequacy. Moreover, we balance our techniques with two extra WEKA programming styles. Tests show that both of our proposed IDS techniques work on the pace of discovery and wipe out missed assaults while eliminating handling time. Security frameworks currently have serious worries because of the sharp ascent of organization business information. As broadly endlessly involved security techniques for correspondence organizations, intrusion detection systems (TDSs) are not a special case. An intrusion detection system (IDS) isolates network business information into typical and strange characterizations to screen network traffic information and recognize attacks. Inferable from the organization business information's high dimensionality, an interruption discovery framework may not generally have the option to rapidly and precisely recognize interruptions. To beat an interruption recognition framework's disappointment and work on its exhibition by improving on its design and stimulating the disclosure interaction, highlight choice becomes fundamental. Because of the dimensionality decrease issue raised by M.B. Shahbaz et al.(2016) [2], we have fostered a proficient component determination procedure that takes the connection between's a subset of qualities and the conduct class name into account. Two relationship rules are used to decide the level of dependence among elements and class names as well as among highlights:symmetrical uncertainty (SU) and correlation-based feature selection (CFS). The proposed procedure with less elements works far superior to existing methodologies as far as preparing time and model creation time while keeping up with or expanding framework delicacy, as indicated by trial discoveries on the NSL-KDD dataset (3, 5). Besides, a correlation of the proposed highlight choice design's viability with different characterization calculations shows that the J48 classifier, which has the most elevated delicacy and flawlessness values as well as the least miss rate and deception rate values, performs best while utilizing the recommended include determination style. The hindering impacts of cyberattacks on society have raised as of late because of the attacks' rising recurrence. Thusly, it is important to research network safety and forestall cyberattacks, with interruption identification filling in as a strong line of protection. Machine proficiency and profound education styles are generally used in both the investigation and improvement of the interruption

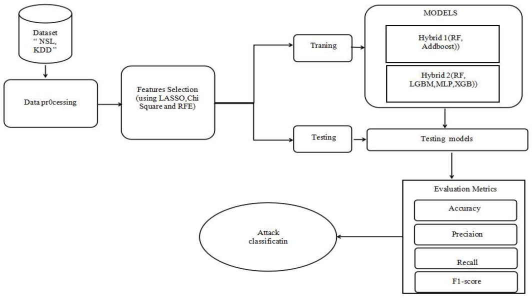
disclosure frameworks and the NSL-KDD (3, 5) dataset is persistently utilized in calculation investigation and check. As upheld by the NSL-KDD dataset, we present Y. Liu (2020) [3] as the two-stage dimensionality reduction (TSDR) include determination technique. The strategy significantly increments handling effectiveness while diminishing the dataset's dimensionality. The superior calculation viability of the new element determination technique is affirmed by the KNN [3] calculation. Contrasting the delicacy rate with the whole point calculation, there is not any decrease.
Network security is turning into a significant issue for every circulated framework. Numerous dangers are becoming more earnestly to identify by firewall and antivirus programming. Intrusion detection systems (IDSs) [17] are utilized to track down abnormalities in the organization's tasks to improve security. Deciding if the organization's approaching business is peculiar or legitimate is the issue of organization oddity finding. The computerized disclosure strategy used to interface the approaching strange business designs frequently utilizes machine education methods. We utilized the Data Gain-grounded strategy in the paper by Zaid Ibrahim Rasool Hani et al (2021) [4]. The strategy chooses the ideal number for a component from an NSL-KDD dataset. Moreover, we have consolidated the point determination process with the machine education model known as Support Vector Machines (SVM) by using the Fake Honey bee State calculation and the Streamlining Cuckoo Search Calculation to improve the hyperactive boundaries of SVM for powerful dataset classification.The execution of the proposed framework has been assessed utilizing the state-of-the-art NSLKDD interruption dataset. As per trial information, the recommended framework works better and accomplishes a more elevated level of delicacy than the other state of the art strategies in NSLKDD (3, 5). The most common way of interfacing interruptions is known as interruption disclosure. Nowadays, an interruption location framework (IDS) [17] is a pivotal instrument for checking networks since it can distinguish dubious examples that highlight potential framework assaults and screen both inbound and active traffic. As of late, a few specialists have created IDS utilizing information mining procedures. Begum et al's. (2017) [5] study analyzes the viability of information mining-based machine education procedures, for example, fluffy C-implies bunching and K-implies grouping, in recognizing interruptions over the NSL-KDD dataset. The essential assault orders that are really identified are DoS, R2L, U2R, and request.

METHODOLOGY :

Proposed Work:

To remove the most pertinent elements from the per-handled NSL-KDD dataset, a crossover highlight extraction approach is recommended [3, 5]. Our technique depends on this dataset, which has been painstakingly chosen to safeguard data that is fundamental for AI strategies in network traffic examination and irregularity recognizable proof. To assess these models' ability to expect network dangers, broad testing is led on them. To reinforce network security, the best model is picked for additional examination and execution given assessment boundaries including precision and blunder rates. To improve prediction accuracy, two hybrid models are used in this work. The first hybrid model comprises Random Forest and AdaBoost where Voting Classifier is used to combine those and the second hybrid model comprises Random Forest, Multi-layer Perceptron with LightGBM, and XGBoost are also used where Stacking Classifier is used to combine to get the best results. This gathering strategy exhibited its viability in producing solid gauges with an astonishing almost 99% accuracy rate. To help client testing, we likewise made a natural front end with client validation capacities to ensure safe framework access. This was achieved utilizing the flask system. This expands our framework's ability for expectation, however, it likewise further develops security and client experience for testing and organization.

SYSTEM ARCHITECTURE:

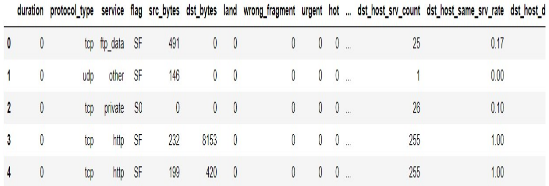


Proposed architecture

Fig shows the proposed architecture to enhance the accuracy rate. This diagram outlines a comprehensive framework for classifying network attacks using the NSL-KDD dataset. The process begins with data pre-processing to clean and prepare the dataset for analysis. Following this, feature selection techniques such as LASSO, Chi-Square, and Recursive Feature Elimination (RFE) are employed to identify the most relevant features for classification. Two hybrid models are trained and tested for their performance: the first one is a voting classifier that combines Random Forest (RF) with AdaBoost, while the second is the stacking classifier that integrates Random Forest with Light Gradient-Boosting Machine (Light GBM), Multi-Layer Perceptron (MLP), and Extreme Gradient Boosting (XGB). The models are evaluated based on standard performance metrics, including accuracy, precision, recall, and F1-score. The objective is to enhance the accuracy and robustness of attack classification in network security by leveraging hybrid machine-learning approaches.

DATASET COLLECTION:

The KDD cup99 dataset was the reason for the advancement of the public dataset NSL-KDD [3, 5] (Tavallaee et al., 2009). As indicated by Tavallaee et al. (2009), a factual investigation of the cup99 dataset uncovered critical issues that impact the invasion revelation delicacy and lead to a deceptive evaluation of Helps. The huge number of indistinguishable bundles in the KDD informational collection is the essential issue. After taking apart the KDD preparing and test sets, Tavallaee et al. viewed that as, separately, 78 and 75 per cent of the organization parcels are rehashed in the preparation and test datasets (Tavallaee et al., 2009). Because of the huge number of indistinguishable models in the preparation set, machine proficiency styles would be affected to turn out to be more one-sided toward commonplace circumstances, which would keep them from learning sporadic cases, which are many times more destructive to the PC framework. To address the previously mentioned issues, Tavallaee et al. made the NSL-KDD dataset in 2009 utilizing the KDD Cup'99 dataset, barring copy passages (Tavallaee et al., 2009). Fig 3.3.1 shows the NSL KDD dataset. There are 125,973 records in the NSL-KDD train dataset and 22,544 passages in the test dataset. It is plausible to use the whole NSL-KDD dataset without the requirement for an irregular example due to its significant size. This has prompted consonant and tantamount results from lively request meetings. There are 42 preparation interruption attacks and 41 attributes (i.e., highlights) in the NSL_KDD dataset. Nineteen highlights in this dataset mirror the kind of associations inside a similar host, though 21 credits relate to the actual association (Tavallaee et al., 2009).



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_d
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	

Fig NSL KDD dataset

DATA PROCESSING:

Information handling is the method involved with transforming natural information into significant data for organizations. Information researchers frequently handle information by social affairs, arranging, imagining, testing, and changing data into reasonable portrayals like papers or diagrams. There are three techniques for handling information: mechanical, electrical, and preparing. A definitive objective is to further develop data esteem and work with navigation. Organizations can work on their tasks and go with opportune vital choices. Aftereffects of mechanized information handling, which are practically identical to PC programming improvement, are significant in this. A lot of information, particularly large information, might be changed into valuable insights to help top-notch tasks and decision-production with its help.

FEATURE SELECTION:

RFE, LASSO and CHi-square are popular Feature selection methods in ML literacy.LASSO regression is a form of logistic regression that incorporates shrinkage.The L1 regularization method applied in Lasso regression imposes a penalty equivalent to the absolute value of the coefficients.RFE is a feature selection algorithm of the wrapper type. This is the most common way of recognizing the most helpful, amicable, and non-excess highlights to remember for a model. The chi-square test evaluates the dependency between each feature and the target variable, aiding in the selection of features that most significantly influence the classification outcome. As the amount and assortment of datasets increment, it is fundamental to decrease their size efficiently. Upgrading a prophetic model's presentation and bringing down the computational weight of demonstrating are the essential objectives of point choice. The method involved with choosing the most critical highlights to incorporate into machine proficiency calculations is known as point determination, and it is one of the essential parts of point design. By killing unessential or futile qualities and diminishing the assortment of information to those that are generally pertinent to the machine proficiency model, highlight determination strategies are utilized to bring down the quantity of information factors. The essential benefits of highlighting determination ahead of time rather than depending on the machine education model to pick the striking qualities.

ALGORITHMS:

Random Forest:

Random Forest is an ensemble learning technique that works by generating numerous decision trees during the training phase and then producing the class that is the most frequent (for classification) or the average prediction (for regression) of the individual trees. This method is intended to address the overfitting issue commonly associated with single decision trees.

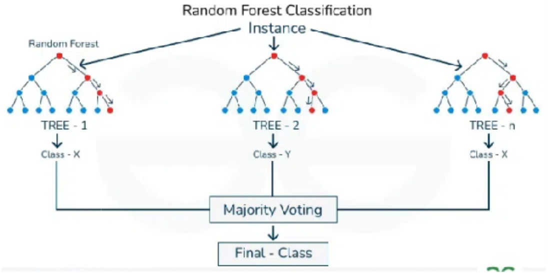


Fig: Random Forest Classifier

RANDOM FOREST CONSTRUCTION:

Fig shows how random Forest Classifier is constructed.

Tree Growth:

A subset of qualities m is decided indiscriminately for every node. Out of these m qualities, the best parted is chosen. The absence of connection between the trees is helped by their capriciousness.

Prediction:

Each tree h_t provides a predicted class. The final prediction y^ is the mode of all predictions.

Y^=mode {ht(x)} ^{T} t=1

For regression: Each tree h_t provides a predicted value. The final prediction

Y^ is the average of all predictions.

Y^ $= \frac{1}{T} \sum_{t=1}^{t} h_t(x)$

ADABOOST: AdaBoost (Adaptive Boosting) is an ensemble learning method that combines several weak classifiers to create a strong classifier, primarily for classification tasks. The core idea of AdaBoost is to enhance the model's performance by iteratively adjusting the weights of instances that are difficult to classify.

For every node, a subset of qualities m is picked randomly. One is chosen as the best part of these m credits. The trees' unreasonableness helps with the absence of an association between them.

Initialize weights: Assign equal weights to all training instances.

For t = 1 to T (T is the number of weak learners):

a. Train a weak learner (e.g., decision tree) using the training data.

b. Calculate the weighted error rate of the weak learner.

c. Given the mix-up rate, decide the weight update boundary.

d. Change the preparation occurrence loads: Raise the loads allocated to mistakenly arranged cases. Reduce the loads of cases that were effectively sorted. e. Change the reconsidered loads with the end goal that they amount to

Final prediction: Utilizing a weighted greater part vote, in which understudies who improve assigned larger weights, aggregate the forecasts of all underperforming students.

Here's a more detailed explanation of each step:

Step 1: Initialize weights

Assign equal weights w_i = 1/N to all N training instances.

Step 2: Train weak learner

Train a weak learner (e.g., decision tree) on the weighted training data.

Step 3: Calculate error rate

Compute the weighted error rate error of the weak learner.

Step 5: Compute weight update parameter

Calculate the weight update parameter α_t based on the error rate error.

Step 6: Update weights

Increase the weights of misclassified instances by multiplying them with exp (α_t).

Decrease the weights of correctly classified instances by multiplying them with exp (-α_t).

Step 7: Normalize weights

Normalize the updated weights to ensure they sum up to 1.

Step 8: Final prediction

A weighted greater part vote of every frail student, where every student's expectation h_t(x) is intensified by its comparing weight, brings about the last forecast H(x) for another example x.

VOTING CLASSIFIER:

A voting classifier is an ensemble machine learning model that combines multiple individual models (classifiers) and outputs the prediction based on a majority vote. In the hard voting classifier, each model votes for a class label and the class with most votes is chosen as a final prediction.

Steps in voting classifier:

1. Initialize the models: Create the instances of Random Forest and AdaBoost models.
2. Fit the models: Train both models on the training dataset.
3. Make predictions: Each model makes predictions on the input data

4. Aggregate the predictions: For hard voting, each model votes for a class, and the class with a majority vote is selected.

HYBRID 1 :

It combines an AdaBoost Classifier with 100 estimators and a Random Forest Classifier with 50 estimators into a single ensemble model. The Voting Classifier is configured with soft voting, meaning it uses the predicted probabilities from each base classifier to make the final prediction. After training the ensemble on the training data (X_train and y_train), it predicts the labels for the test data (X_test). The performance of the Voting Classifier is then evaluated using various metrics: accuracy, precision, recall, F1 score, specificity, and error rate. The results are stored using the store Results function, with the model name and the computed metrics as inputs. This approach effectively combines the strengths of AdaBoost and Random Forest, potentially leading to improved predictive performance compared to individual classifiers.

STACKING CLASSIFIER (LGBM + MLP + RF + XGB) :

Stacking, likewise goes by the name "stacked generalization," is a troupe learning technique that improves expectation execution by consolidating many ML models. The fundamental idea is to total the expectations from many base models via preparing a meta-model (likewise alluded to as a blender or stacker). The base models in the gathering for the Stacking Classifier (LGBM + MLP + RF + XGB) are as per the following. Tree-based learning procedures are utilized in the slope-supporting system known as Light Gradient Boosting Machine, or LightGBM. It is famous for having incredible exactness, and effectiveness, and for dealing with monstrous information. Multi-layer Perceptron, or MLPs for short, are a sort of feed-forward fake brain network comprised of a few levels of connected nodes. MLPs are frequently utilized for various applications, like relapse and grouping, and are fit for capable of learning non-direct capabilities.Another unimaginably compelling and versatile angle-helping machine arrangement is called XGB: Extreme Gradient Boosting (XGBoost). It contains built-in regularization and missing data handling capabilities, and it is extremely parallelizable.Utilizing a meta-model — by and large, one more ML calculation prepared on the expectations of the premise models — the Stacking Classifier incorporates the outcomes from these four base models.

This is an itemized depiction of the Stacking Classifier's activity:

1. Models of Train Bases: Utilize the preparation information to independently prepare the LGBM, MLP, RF, and XGB base models.
2. Set up Meta-Highlights: To create forecasts in light of the preparation information, utilize each educated base model. The info highlights for the meta-model will be these conjectures, which are alluded to as meta-highlights.
3. Partitioned Information Make two subsets from the preparation information: a meta-preparing set and a meta-approval set.
4. Train Meta-Model: Utilizing the meta-highlights (forecasts from base models) as info and the genuine marks as targets, train the meta-model (e.g., another ML method like logistic regression or a neural network) on the meta-preparing set.
5. Validate Meta-Model: Survey the meta-model's adequacy utilizing the meta-approval set.
6. Make Predictions: On account of new, unused information: a. Make meta-highlights by anticipating each base model. b. To get a definitive gauge, feed the meta-highlights into the prepared meta-model.

More noteworthy prescient execution may much of the time be accomplished by joining different base models (LGBM, MLP, RF, and XGB) with varying strengths and impediments than by utilizing any solitary model. Further developed exactness and heartiness are the result of the meta-model's capacity to coordinate the benefits of the base models and compensate for their specific deficiencies.

HYBRID :

It incorporates a Random Forest Classifier with 1000 estimators and an MLP Classifier with a maximum of 3000 iterations as base estimators. The final estimator in the stack is an LGBM Classifier with 1000 estimators. The Stacking Classifier is trained on the training data (X_train, y_train) and used to predict the test data labels (X_test). Various performance metrics, including accuracy, precision, recall, F1 score, specificity, and error rate, are computed and stored using the store Results function. This ensemble approach leverages the strengths of different classifiers to potentially improve overall model accuracy and robustness.

EXPERIMENTAL RESULTS :

Precision: Precision estimates the level of accurately sorted examples or occurrences among the positive examples. In this way, the Precision might be determined utilizing the accompanying condition:

$$\text{Precision} = \frac{True\ Positives}{(True\ Positives \qquad\qquad Positives)}$$

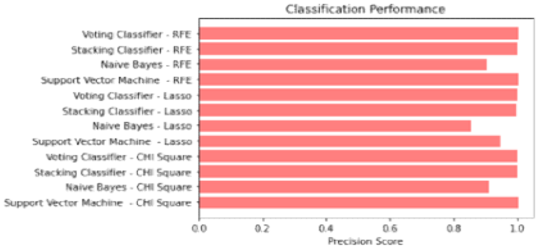


Fig Precision comparison graph

Fig 4.1 shows the comparison of different ML algorithms for Precision metrics. The image appears to be a bar chart displaying various classifier models with feature selection methods (RFE, Lasso &Chi-square) along the y-axis and a performance precision metric, along the x-axis. Each bar represents a different classifier's performance. From top to bottom, the classifiers are Voting(AdaBoost+RF),Stacking Classifier(Extreme Gradient Boosting, LightGBM, MLP (Multi-Layer Perceptron), Random Forest),Naive Bayes, Support Vector Machine (SVM).The lengths of the bars indicate the relative performance of each classifier, with Voting Classifier with RFE showing the highest performance and Naive Bayes the lowest among the displayed classifiers.

Recall: The level of precisely arranged occasions or events among the positive models is assessed by precision. This implies that the accompanying rules may be utilized to work out the precision:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

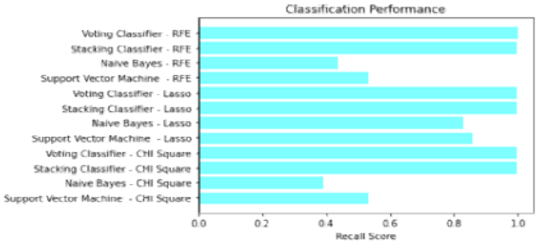
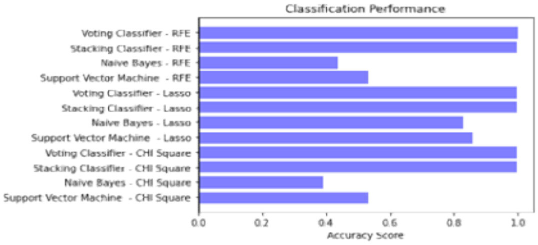


Fig Recall comparison graph

Fig shows the comparison of different ML algorithms for the Recall metric. The image appears to be a bar chart displaying various classifier models with feature selection methods (RFE, Lasso&ChiSquare) along the y-axis and a performance precision metric, along the x-axis. Each bar represents a different classifier's performance. From top to bottom, the classifiersare Voting (Adobos+RF), Stacking Classifier (Extreme Gradient Boosting, LightGBM, MLP (Multi-Layer Perceptron), Random Forest), Naive Bayes, Support Vector Machine (SVM). The lengths of the bars indicate the relative performance of each classifier, with the Voting Classifier with RFE showing the highest performance and Naive Bayes-Chi-square the lowest among the displayed classifiers.

Accuracy: The level of accurate expectations spread the word about in a characterization work is as accuracy, and it shows how exact a model's forecasts are by and large.



$$Accuracy = \frac{TP+TN}{TP+FP+TN+F}$$

Fig 4.3 Accuracy graph

Fig 4.3 shows the comparison of different ML algorithms for the Accuracy metric. Shows the comparison of different ML algorithms for the Accuracy metric. The image appears to be a bar chart displaying various classifier models with feature selection methods (RFE, Lasso &ChiSquare) along the y-axis and a performance precision metric, along the x-axis. Each bar represents a different classifier's performance. From top to bottom, the classifiers are Voting (AdaBoost+RF), Stacking Classifier (Extreme Gradient Boosting, LightGBM, MLP (Multi-Layer Perceptron), Random Forest), Naive Bayes, Support Vector Machine (SVM). The lengths of the bars indicate the relative performance of each classifier, with the Voting Classifier with RFE showing the highest performance and Naive Bayes-Chisquare the lowest among the displayed classifiers.

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$F1\ Score = 2 * \frac{Recall\ \times\ Precsion}{Recall\ +\ Precision} * 100$$

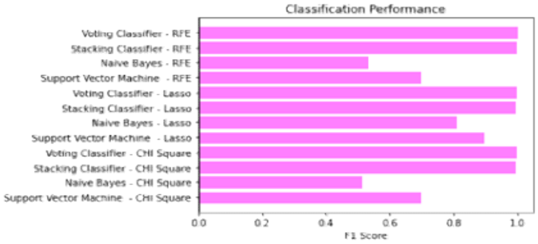


Fig :F1Score

Fig 4.4 shows the comparison of different ML algorithms for the F1 score metric. The image appears to be a bar chart displaying various classifier models with feature selection methods (RFE, Lasso &ChiSquare) along the y-axis and a performance precision metric, along the x-axis. Each bar represents a different classifier's performance. From top to bottom, the classifiers are Voting (Adobos+RF), Stacking Classifier (Extreme Gradient Boosting, LightGBM, MLP (Multi-Layer Perceptron), Random Forest), Naive Bayes, Support Vector Machine (SVM). The lengths of the bars indicate the relative performance of each classifier, with Voting Classifier with RFE showing the highest performance and Naive Bayes the lowest among the displayed classifiers

Table 1: Comparison of evaluation metrics with various models.

ML Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
AdaBoost-Chi-Square	0.738	0.736	0.738	0.691
Random Forest-	0.995	0.995	0.996	0.997

Chi-Square				
Hybrid 1 - Chi-Square	0.996	0.997	0.996	0.997
LGBM-Chi-Square	0.840	0.860	0.840	0.848
MLP-Chi-Square	0.932	0.930	0.932	0.931
XGB-Chi-Square	0.996	0.997	0.996	0.996
Hybrid 2-Chi-Square	0.959	0.958	0.959	0.958
AdaBoost-Lasso	0.685	0.772	0.685	0.702
Random Forest-Lasso	0.996	0.997	0.996	0.996
Hybrid 1 - Lasso	0.997	0.997	0.996	0.996
LGBM-Lasso	0.926	0.925	0.926	0.925
MLP-Lasso	0.991	0.991	0.991	0.991
XGB-Lasso	0.997	0.997	0.997	0.997
Hybrid 2-Lasso	0.995	0.995	0.995	0.995
AdaBoost-RFE	0.962	0.966	0.967	0.994
Random Forest-RFE	0.996	0.997	0.997	0.995
Hybrid 1-RFE	0.998	0.999	0.998	0.999
LGBM-RFE	0.923	0.934	0.939	0.966
MLP-RFE	0.993	0.993	0.991	0.996
XGB-RFE	0.997	0.996	0.994	0.993
Hybrid 2-RFE	0.996	0.997	0.996	0.997

Table 1The performance metrics of various machine learning models using different feature selection methods reveal significant insights. For the Chi-Square method, the Hybrid 1 model excelled with an accuracy of 0.996, closely followed by XGB and Random Forest, both achieving high precision, recall, and F1 scores. LGBM and MLP also performed well, with accuracies of 0.840 and 0.932, respectively. The Lasso feature selection showed the highest accuracy with Hybrid 1 and XGB, both at 0.997, while Random Forest and MLP also achieved excellent results. Adobos had a lower accuracy of 0.685 with Lasso. For RFE, Hybrid 1 stood out with an accuracy of 0.998, followed by XGB and Random Forest at 0.997 and 0.996, respectively. LGBM had a lower performance with RFE at 0.923 accuracy. Overall, Hybrid models, particularly Hybrid 1, consistently demonstrated superior performance across all feature selection methods, highlighting their robustness and effectiveness in improving predictive accuracy and other metrics.

CONCLUSION &FUTURE SCOPE :

Utilizing ML procedures, the exploration effectively constructed a crossover including an extraction technique to recognize network attacks. By eliminating repetitive information from the 1999 KDD Cup dataset, the NSL-KDD dataset [3, 5] improves on information and gives a proper premise to ML investigation and irregularity recognizable proof. The review prepared models utilizing a twofold order approach, permitting exact forecasts of typical and assault types through cautious element determination and pre-processing. Voting Classifier (RF+AdaBoost) performs strikingly well, testing at a high exactness pace of close to 100%. The calculation is tried in a front-end interface that permits clients to enter highlight values, working with client communication and offering a helpful illustration of the calculation's viability in certifiable situations. Assessment measures showed that the superior adaptation of the task beat the prior results, for example, exactness and mistake rates. Outstanding upgrades improved speed and reinforced the framework's safeguard against network dangers. The task underscores how significant organization traffic investigation is to fortifying organization protection from potential interruptions. As well as pushing for proactive endeavours to keep away from network interruptions, the drive stresses the meaning of areas of strength for a discovery framework.

Continuous model improvement is crucial for ML models in cybersecurity applications to stay effective against evolving threats. Integration with threat intelligence can enhance the identification of the latest threats by using shared data and intelligence feeds. Automated incident response features, such as alert generation, blocking malicious traffic, and implementing corrective measures, can extend the program's capabilities. Network visualization and forensic tools can aid in incident analysis and prevention by providing deeper insights into attacks. To handle large datasets efficiently, the application should be designed for scalability and performance optimization. Cloud integration can enhance the application's accessibility, flexibility, and maintenance ease while improving user experience and reporting systems can provide stakeholders with detailed insights into threats and overall security posture.

REFERENCES :

1. F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani and A. Taleb-Ahmed, "Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23-29, doi: 10.1109/IoTaIS50849.2021.9359689.
2. M. B. Shahbaz, Xianbin Wang, A. Behnad and J. Samarabandu, "On efficiency enhancement of the correlation-based feature selection for intrusion detection systems," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON),2016,pp.1-7,doi: 10.1109/IEMCON.2016.7746286.
3. T. Yu, Z. Liu, Y. Liu, H. Wang and N. Adilov, "A New Feature Selection Method for Intrusion Detection System Dataset – TSDR method," 2020 16th International Conference on Computational Intelligence and Security (CIS),2020,pp.362-365,doi:10.1109/CIS52066.2020.00083.
4. Ali Hussein Shamman Al-Safi , Zaid Ibrahim Rasool Hani, , Musaddak M. Abdul Zahra,"Using A Hybrid Algorithm and Feature Selection for Network Anomaly Intrusion Detection", Journal

of Mechanical Engineering Research and Developments, ISSN: 1024-1752, CODEN: JERDFO, Vol. 44, No. 4, pp. 253-262. Published Year 2021.

5. P. S. Bhattacharjee, A. K. Md Fujail and S. A. Begum, "A Comparison of Intrusion Detection by K-Means and Fuzzy C-Means Clustering Algorithm Over the NSL-KDD Dataset," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2017, pp. 1-6, doi: 10.1109/ICCIC.2017.8524401.

6. Y. J. Haranwala, S. Haidri, T. S. Tricco, V. P. da Fonseca and A. Soares, "A Dashboard Tool for Mobility Data Mining Preprocessing Tasks," 2022 23rd IEEE International Conference on Mobile Data Management (MDM), 2022, pp. 278-281, doi: 10.1109/MDM55031.2022.00059.

7. Z. Wang et al., "Image Noise Level Estimation by Employing Chi-Square Distribution," 2021 IEEE 21st International Conference on Communication Technology (ICCT), 2021, pp. 1158-1161, doi: 10.1109/ICCT52962.2021.9657946.

8. M. S. S. Sumi and A. Narayanan, "Improving Classification Accuracy Using Combined Filter+Wrapper Feature Selection Technique," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-6, doi: 10.1109/ICECCT.2019.8869518.

9. M. Zhu, X. Huang and H. Pham, "A Random-Field-Environment-Based Multidimensional Time-Dependent Resilience Modeling of Complex Systems," in IEEE Transactions on Computational Social Systems, vol. 8, no. 6, pp. 1427-1437, Dec. 2021, doi: 10.1109/TCSS.2021.3083515.

10. Y. Kim, J. Hao, T. Mallavarapu, J. Park and M. Kang, "Hi-LASSO: High-Dimensional LASSO," in IEEE Access, vol. 7, pp. 44562-44573, 2019, doi: 10.1109/ACCESS.2019.2909071.

11. N. S. Rahmi, N. W. S. Wardhani, M. B. Mitakda, R. S. Fauztina and I. Salsabila, "SMOTE Classification and Random Oversampling Naive Bayes in Imbalanced Data : (Case Study of Early Detection of Cervical Cancer in Indonesia)," 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA), 2022, pp. 1-6, doi: 10.1109/ICITDA55840.2022.9971421.

12. L. Zeyang, "Research on Intelligent Acceleration Algorithm for Big Data Mining in Communication Network Based on Support Vector Machine," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2021, pp. 479-483, doi: 10.1109/AUTEEE52864.2021.9668793.

13. G. R. Kini and C. Thrampoulidis, "Phase Transitions for One-Vs-One and One-Vs-All Linear Separability in Multiclass Gaussian Mixtures," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 4020-4024, doi: 10.1109/ICASSP39728.2021.9414099

14. J. Sewall, S. J. Pennycook, D. Jacobsen, T. Deakin and a. S. McIntosh-Smith, "Interpreting and Visualizing Performance Portability Metrics," 2020 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 2020, pp. 14-24, doi: 10.1109/P3HPC51967.2020.00007