Journal of Information Systems Engineering and Management
2025, 10(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Integrating Natural Language Processing with AdaBoost, Random Forest, and Logistic Regression for an Advanced Ensemble-Based Network Intrusion Detection Model

Putta Srivani[1], Dr. Himanshu Sharma[2], Dr. Rabins Porwal[3], T. Nagalakshmi[4], P.Mercy[5], Mallareddy Adudhodla[6], Nargis Parveen[7]

[1]*Associate Professor, Department of CSE (AI/ML), Malla Reddy Engineering College for Women, pulla.srivani@gmail.com*

[2]*Associate Professor, Electronics & Communication Engineering Department, GLA University, Mathura, Uttar Pradesh, U.P. himanshu.researcher1@gmail.com*

[3]*Professor, Department of Computer Application, School of Engineering & Technology (UIET), Chhatrapati Shahu Ji Maharaj University (CSJMU), Kanpur, rabins@csjmu.ac.in*

[4]*Assistant Professor, Department of Mathematics, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Avadi, 600062 , Tamilnadu, India, nagalakshmi.1979@gmail.com*

[5]*Assistant Professor, Computer Science and Business Systems, Anand Institute of Higher Technology, Chennai, pmercy4@gmail.com*

[6]*Professor, IT Department, CVR College of Engineering, mallareddyadudhodla@gmail.com*

[7]*Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Kingdom of Saudi Arabia, nargis.norulhaq@nbu.edu.sa*

Corresponding author mail: mallareddyadudhodla@gmail.com

ARTICLE INFO

Received: 10 Oct 2024

Revised: 05 Dec 2024

Accepted: 21 Dec 2024

ABSTRACT

Higher numbers and more complicated traffic data passing over your network requires that you have some advanced ways to properly identify security threats. In this paper, we design an ensemble intrusion detection system that unites different machine learning strategies such as due to Natural Language Processing (NLP), AdaBoost, Random Forest and Logistic Regression to identify several kinds of network intrusions. Problem: To deal with more and more stronger attacks, detect rate of the traditional Intrusion Detection Systems remains feckless. Based on the results of this analysis, the important features are extracted from network traffic logs by applying NLP and feed it to espresso classifier in the ensemble approach. Because AdaBoost increases the performance of weak learners, Random Forest provides robustness and Logistic Regression provides interpretability in final decision making. It is trained and tested on benchmark datasets like the NSL-KDD, and CICIDS2017. Overall, in the case results ensemble models were better performance than all individual classifiers regarding accuracy, precision and recall especially in rare case of attack types. Using NLP for feature extraction has enabled the detection of sophisticated attack signatures, showing that this model performs well for real-time security monitoring in high throughput networks like corporate or cloud environments.

Keywords: Security, threats, AdaBoost, NLP, decision, feature, network.

INTRODUCTION:

It plays an important role where we require real-time monitoring for example corporate and cloud networks. Results of the system analysis in comparison to NSL-KDD and CICIDS2017 carry it out performs better with data set, as proved that the ensemble model improves accuracy, precision, recall compared to lone classifiers especially in rare attack cases[1].

Incorporating NLP for feature extraction into the system facilitates detection of advanced attack signatures. By integrating machine learning approaches that can better handle big and heterogeneous network attacks, this model suggests a sound solution for next-generation cybersecurity operations. This adaptability of ensemble model further

increase the detection accuracy which makes significant contribution towards a dynamic approach in IDS design to mitigate newer cyber threat landscape and comprehensive defence deployed.

Growing Demand for Sophisticated Intrusion Detection Systems:

With the increase in global networks (because of IoT, cloud computing and the extensive digital communication among other things), there is also a corresponding growth of traffic volume and complexity like never before. However, with this growth, it becomes more exposed to different kinds of cyberattacks as well. Modern, advanced threats have made traditional Intrusion Detection Systems (IDS) that simply use signatures for detection, insufficient. These systems generally use predefined rules or patterns in order to detect malicious behaviour on a network which do not follow any known patterns of attack[2].

However, there are limitations to the abilities of IDS softwares that work on these principles; to deal with expanding data sets and detection of different varieties of network intrusion (including rare/exploits attacks that are difficult to detect), a more sophisticated model for an IDS is required. In this paper1, we meet this need by developing an ensemble learning method which combines multiple machine learning models that possess complementary advantages for improving overall detection accuracy, precision and recall [23]. Our objective in this paper is to build and test an ensemble, which can harness the performance of NLP (Natural Language Processing), AdaBoost algorithm for ML, Random Forest classifier and Logistic Regression to increase detection rate with lower false positive and false negative while being able to sustain real-time detection on high-throughput corporate or cloud-level network data[3].

- Feature extraction through Natural Language Processing (NLP)

A main change from the previous model is that the new one makes an improvement with the integration of NLP to extract useful characteristics obtained from network traffic logs. In the typical network environment, it is high-dimensional data often containing a mix of categorical and continuous values. NLP: normally used in Text Mining and Linguistic Analysis, is introduced here as a method to parse network traffic logs into more meaningful parts, which are filitered out important features that could capture the profile of normal or anomalous activity in the network. Because NLP also works well with unstructured data at scale like network logs, the method is especially valuable for finding patterns that traditional feature extraction methods cannot[4].

Using the power of NLP, the raw network traffic data could be translated into structured information to be utilized by machine learning algorithms. This is especially useful in finding rare and complex attack signatures that pass through most traditional IDS systems by not being defined as preconfigured rules or patterns. Moreover, NLP feature extraction changes dynamically which makes it possible to train an IDS that can recognize those activity forms in real-time network contexts. In this we have used the AdaBoost, Random Forest and Logistic Regression to train my model with ensemble learning[5].

Using ensemble learning, one of the main elements that is proposed:Pooled learning among machine learning algorithms makes a powerful system with accurate prediction. In this paper, an ensemble approach that combines AdaBoost and Random Forest with Logistic Regression has been considered to leverage the advantages from these IDS.

1. AdaBoost (Adaptive Boosting): AdaBoost is a boosting algorithm aimed at trying to improve the performance of weak learners, basically models that are only slightly better than random guessing. In an intrusion detection setting, the base classifiers learn a measure from which AdaBoost improves this measure only for those instances that have been misclassified in previous iterations. AdaBoost does this by iteratively adjusting the weights of training samples, and focussing more on difficult-to-classify instances[6]. But ability of AdaBoost for higher detection rates in rare or subtle attack types, which are often more missed by individual classifiers.
2. The Random Forest is an algorithm and it is called so because of a forest of many trees. In the case of intrusion detection, managing high-dimensionality network traffic data on-feature subset selected at each split helps to discover complex patterns from the data through Random Forest model. The performance of such techniques on large datasets and toleration against noisy data also influence the suitability for IDS Since, incomplete / inconsistency within network traffic logs may leads IDS to a risk factor as well[7].Random Forest is powerful and is capable of well-generalizing across different kinds of traffic. Furthermore, Random Forest cuts down overfitting by ensembling multiple decision tree´s output, a common problem in high-dimensional data environments such as network intrusion detection[8,33].

3. Logistic regression: The Logistic Regression part acts ideally as the final decision making layer which adds interpretability to the system. Simple model but powerful in finding the probability of whether a specific instance belongs to which class(normal or attack):Though Logistic Regression is too naïve, we cannot compare this model with AdaBoost and Random Forest models. Interpretability is particularly important in network security as high accuracy is not sufficient if we do not understand the reason behind a model's decision.

The model can now produce probabilistic outputs to let security analysts know what confidence level they could expect for each prediction due to the Logistic Regression integrated into the ensemble. This is beneficial not only in terms of enhancing the performance of the final decision-making, but also delivering transparency and trust to the end-user (e.g., field commander) for when IDS would be employed in real-world settings[9].

In this paper, we have evaluated the proposed ensemble-based IDS model over one of the most commonly used and most recent two benchmark datasets namely NSL-KDD and CICIDS2017. These are some of the datasets which have been mostly used by various intrusion detection research community and these datasets hold labeled network traffic data reflecting normal usage and malicious activities. NSL-KDD (An improved version of the KDD Cup 99 dataset) and CICIDS2017 (Real-world traffic that contains recent attack types[10,34].

Performance Metrics: The results show significant improvement in the detection rate and the ability of handling rare attack types compared to individual classifiers with an ensemble model. By integrating NLP for feature extraction and combine the strength of AdaBoost, Random Forest for detecting common types attacks as well Logistic Regression for rare type attack pattern have more precision[11].

The paper, highlight one key discoveries that is by combining different models or using ensembles we can achieve greater detection rate with lesser number of false-positives which is a primary concern in real-world IDS deployment. If we are incorrect in identifying normal traffic as being malicious, this is a false positive and can flood security analysts, eventually leading to untrustworthy environment. It also lessens the number of false positives, ultimately giving security teams come together to concentrate on real threats, thus improving the IDS as a whole[12].

This paper's last contribution is for which the work may be deployed in practice. The ensembling of the machine learning methods makes this model suitable for high-throughput environments, e.g., corporate or cloud networks where intrusion detection is supposed to be accomplished in a real-time scenario. The incorporation of NLP for dynamic feature extraction ensures the system can learn new attack patterns, making it fit for deployment in environments susceptible to rapidly changing threats.

Our ensemble-based IDS model builds upon the best of AdaBoost, Random Forest, and Logistic Regression to give high accuracy while remaining interpretable by design. Hence, it a useful tool for administrators and security professionals to add in their arsenal to fortify the defenses against the growing complexities of cyber-attacks.

Moreover, the ensemble-based IDS model proposed not only offers a combination of the best performing techniques of NLP with AdaBoost; but it also accounts for weaknesses within Random Forest and Logistic Regression making it a comprehensive solution for network intrusion detection[13,14]. This paper moves the field of network security forward as it gives a remedy to these issues, namely the by providing a scalable, real-time answer that moves away traditional IDS models. By using benchmark datasets like NSL-KDD and CICIDS2017 to evaluate this model, we show that this lightweight approach can serve as a deployable solution in real-world high-throughput network environments[15,35].

RELATED WORK:

Network intrusion detection system stands as the line of defence of cybersecurity infrastructure. They are the ones who monitor and analyse networks to detect any fraudulent activity. The main problem for IDS is how to differentiate between the normal and malicious traffics without overflowing security operators with false positives or ignoring subtle, more advanced attacks[16].

Generally, intrusion detection falls under two categories such as Misuse Detection (Signature-based Detection) and Anomaly detection. The way misuse detection works is by looking for the patterns of known attacks, so if an attack uses a new strategy you have never seen before it won't catch up "just like zero-day attacks". Anomaly detection, meanwhile, only raises alerts when it detects something that is different from what the data has told it to expect as

'normal' behavior up until then; this gives it the chance of detecting a new kind of attack (a zero day), but also increases its chances of generating false positives[17,18].

Earlier, a limited signature model was used in the traditional IDS architectures and however effective against deterministic attacks of known patterns but were not agile enough before the new face of attacks making it fall more suitable for zero-day exploit attempts. Anomaly Detection using machine learning (ML), artificial intelligence (AI) and ensemble learning techniques is the trending paradigm for modern IDS[19-22].

- Ensemble Learning in IDS

In the IDS arena, ensemble techniques have demonstrated their efficacy in handling high-dimensional, noisy and imbalanced network traffic data.

Several Ensemble methods are applied for IDS like:

Bagging: Bagging builds multiple models on different subsets of the data and then fuses their predictions. Randomised Forest: This is a type of ensemble method that uses multiple decision trees for more robustification, Bagging (boostrap aggregating) — one of the most popular bagging methods, and in this case we use Decision Trees.

Boosting: Iteratively improve the performance of weak classifiers by emphasizing instances that have been misclassified in previous models. Example: AdaBoost another classic example of boosting.[23]

Stacking & Blending: These methods aggregate the results of multiple models (base learners) by having another machine learning model (meta-learner) to make a final prediction.[24]

Random Forest in IDS:Bagging technique random forest which is the one of most efficient in dealing with high dimensional data and well complexity of network traffic logs where intrusion detection system is viable for it. Random Forest is an ensemble of Decision Trees which are trained on a different fraction of data created by bootstrapping. Finally, based on a majority vote across all trees the final classification is decided. This method reduces the common issue of overfitting in IDS where certain attack types may be oversampled in the training data.

AdaBoost in IDS:AdaBoost, short for Adaptive Boosting is a technique that tries to create an accuracy classifier using multiple weak classifiers (one level at a time). In the context of an IDS, this is especially useful when we desire to have improved detection for rare or subtle types of attacks which are frequently misclassified by a single classifier standing alone. AdaBoost has achieved high accuracy in intrusion detection, especially when the problem involves imbalanced datasets where rare attack types are easily buried under normal traffic [25-27].

Logistic Regression in IDS:Random Forest is great for feature selection, and it's often used as the last stage in ensemble learning. In network intrusion detection, it assists in generating probabilistic estimates and this plays a crucial role in high stakes scenarios such as those surrounding security operations where security analysts needs to understand the reasoning behind the decision making process of a model. Another example in Q3: Logistic Regression also on binary classification this time it's used to determine whether an event observed by a network is benign or malicious[28].

IDS Feature Extraction using Natural Language Processing (NLP)

The proposal of this IDS model is also distinct since it utilizes NLP techniques to perform feature extraction of logs derived from network traffic. In general, we use NLP for processing and analyzing a huge amount of nature language data(chatbots), but this application to network data is a big step forward. As network traffic logs are high-dimensional, consisting of both categorical and continuous variables; it can be regarded as unstructured data like text data. Using NLP techniques, significant features can be extracted more effectively to describe network normality or maliciousness[29].

The flexibility of using NLP can be beneficial in that it allows the IDS to capture complex patterns and/or signatures that could render undetected by more traditional feature-extraction techniques. In high-throughput environments, where vast quantities of data need to be processed on the fly this is extremely important. The NLP feature extraction of the model also makes it flexible to recent changes in attack vectors.

Table ahead provides a comprehensive and clear overview of each technique, enabling a better understanding of how they integrate into advanced ensemble-based IDS models[30,31,32]. It also balances the strengths and challenges of each approach, giving a well-rounded view of related work in the field.

Technique	Application in IDS	Advantages	Challenges	Key References
Natural Language Processing (NLP)	Applied for feature extraction by parsing network traffic logs to identify key patterns and anomalies.	1. Handles large-scale, unstructured data. 2. Extracts meaningful features for better classification.	1. High complexity in processing large datasets. 2. Requires domain-specific adaptation for network logs.	Studies utilizing NLP for extracting key features from high-dimensional traffic data
AdaBoost (Adaptive Boosting)	Boosts weak classifiers by iteratively focusing on misclassified instances to improve detection accuracy.	1. Enhances detection of rare and subtle attack types. 2. Reduces false negatives. 3. Adaptable to new data.	1. Sensitive to noisy data. 2. Can lead to overfitting in the presence of complex attack behaviors.	Research showing AdaBoost's improvement in detecting rare attack types
Random Forest	Builds an ensemble of decision trees for classification tasks such as anomaly detection in network traffic.	1. Handles high-dimensional data effectively. 2. Robust to overfitting. 3. Works well on large datasets.	1. Slower performance on large datasets. 2. Reduced interpretability compared to simpler models.	Random Forest's effectiveness in handling complex and noisy data in IDS
Logistic Regression	Used as the final decision-making layer, providing binary classification and interpretability of decisions.	1. High interpretability for security analysts. 2. Provides probabilistic outputs for better decision-making.	1. Limited to linear relationships. 2. Not as effective on complex non-linear attack patterns.	Logistic Regression's role in improving interpretability and trust in IDS decisions
Ensemble Learning	Combines models (e.g., AdaBoost, Random Forest, Logistic Regression) for a more robust IDS detection approach.	1. Improves overall detection accuracy and reduces error rates. 2. Handles imbalanced data (e.g., rare attacks).	1. Higher computational cost. 2. Complex to implement and optimize in real-time environments.	Studies demonstrating improved IDS performance through ensemble approaches
NSL-KDD Dataset	Benchmark dataset for training and testing IDS models, containing normal and attack traffic.	1. Improved dataset over KDD'99 (reduced redundancy). 2. Represents various attack types (DoS, U2R, etc.).	1. Still criticized for not fully representing modern attack vectors and traffic diversity.	Widely used in IDS research as a standard for evaluating model performance
CICIDS2017 Dataset	More recent dataset that includes modern attack types (e.g., DDoS, infiltration, brute-force attacks).	1. Comprehensive real-world traffic data. 2. Includes diverse attack types and normal traffic.	1. Large dataset size can be computationally expensive to process. 2. Requires preprocessing and feature extraction.	Modern benchmark dataset for IDS research, especially for machine learning-based detection methods

- Benchmark Datasets: NSL-KDD and CICIDS2017

NSL-KDD and CICIDS2017 are two widely-used datasets in the area.

NSL-KDD Dataset:

NSL-KDD Dataset (is an updated version of the KDD Cup 99 dataset). It was one of the first publicly available datasets for testing IDS models. These redundancies in records and class imbalances are also addressed by NSL-KDD, so it is expected to have a more realistic network traffic. NSL-KDD The dataset has labelled network trafficking data which means almost all types of attacks are included. It is widely used for training and testing intrusion detection systems.

CICIDS2017:

Due to the fact that this dataset reflects more updated traffic and it includes malicious attacks newly added to the current threats, we decided list it as an additional about it. The dataset includes all types of attack against every class in the modern networks such as DoS, DDoS, brute-force attacks and infiltrations. It is expected to be applied for IDS models evaluation, since it shows good feature of labelled attack traffic and types. Most common and some unusual abnormal types are available for usage in order to implement testing of model's performance in detection hiding, disguise etc.

Different researches have also begun to explore them in the context of IDS by identifying different ensemble learning techniques. For example, another study uses stacking ensemble learning which usually consists of base models. The outcome depicted ensemble methods could achieve better performance compared to individual classifiers, and mainly for accuracy and imbalanced data.

Another major area of research is boosting methods (e.g. AdaBoost, gradient boosting). Indeed, these studies find that boosting is useful when developing an IDS to emphasize the rare attacks hidden in the flood of normal traffic. Furthermore, the performance is further enhanced due to deep learning methods integrated into ensemble frameworks which enable the system to learn subtle patterns in network traffic

The combination of NLP and AdaBoost, Random Forest, and Logistic Regression in the ensemble-based IDS model is a great breakthrough in network security. Combining the strengths of these individual methods, NLP for dynamic feature extraction, AdaBoost for boosting weak learners, Random Forest for robust classification, and Logistic Regression for interpretability, this ensemble method forms a powerful tool to detect various network intrusions. Experimental results on benchmark datasets such as NSL-KDD and CICIDS2017 confirms the efficiency of the proposed model, having high more detection accuracy against common and rare attack types.

This is a contribution to the existing literature in ensemble learning, as it merges machine learning with natural language processing to overcome the shortcomings of conventional methods used by Intrusion Detection Systems. It is certainly a promising avenue for real-time network security monitoring, especially in highly contended environments like corporate or cloud networks. In future work, the effectiveness and scalability of this model can be analysed in more complex network environments and other machine learning techniques can be implemented to enhance its accuracy. There are a few pre-existing studies projects which have explored the wave of ensemble learning techniques for IDS. For example, in a study was used stacking ensemble learning with basic models such as decision trees, SVM and neural networks for the detection of anomalies on network traffic. From the results, it is apparent that putting classifiers collectively in an ensemble can outdo a single classifier most notably accuracy and class differences.

A second large group of papers look at boosting methods like AdaBoost and gradient boosting. It has been found from these studies that the boosting techniques have a good ability to increase the discovery numbers for rare attack types which is another common problem in IDS since there are so much normal traffic in training datasets. In addition, introducing deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) into ensemble frameworks has boosted the performance to detect fine-grained structures in network traffic more effectively.

By integrating NLP with AdaBoost, Random Forest and Logistic Regression in an ensemble-based IDS model this is a paradigm shift in network security. This ensemble approach harnesses the benefits of these constituent techniques—NLP for intelligent feature selection, AdaBoost for highly boosted weak learners, Random Forest for resilient classification and easier implementation by Logistic Regression to offer a versatile tool in the detection of network intrusions from broad traffic types. Experiments on benchmark datasets: NSL-KDD and CICIDS2017 are used to compare the performance of the proposed model with other models, and results confirm that our proposed solution can gain better detection of various types of attacks including both common & rare kind. This work extends

research on ensemble learning for IDS to fuse machine learning with natural language processing, aiming to overcome the constraints of traditional intrusion detection approaches. This is a pivoting point that could certainly make real-time network security monitoring more feasible, especially in high-traffic landscapes, such as corporate or cloud networks. Further study can be conducted by generalizing the model used in this work to other complex network environments and including more machine learning fundamentals to better enhance its accuracy as well as scalability.

MODEL FRAMEWORK, ARCHITECTURE AND PROPOSED METHODOLOGY

To improve such malicious network traffic with high accuracy, robustness and interpretability in wise, in this paper, an ensemble learning framework based Network Intrusion Detection System(IDS) model is proposed(School of Information Science and Engineering,Central South University: A Novel deep Learning BasedECTC 1720:93–104). While NLP is used for extracting features, AdaBoost, Random Forest and Logistic Regression are methods that overcome the limitations of a single technique: weak learner boosting combined with different predictors.

3.1 System Architecture

The architecture comprises of below key layers together:

- Input Layer: Real/batch based network traffic Data (NSL-KDD, CICIDS2017 datasets).
- Pre-Processing Layer: The raw traffic data is pre-processed (cleaning, normalization, transformation) to prepare it for feature extraction. For example dealing with missing values, noise, imbalanced datasets etc.
- Feature Extraction Layer: Post which features of interest are extracted from network traffic logs using NLP techniques.
- Ensemble Classifier Layer: Extracted features are passed through ensemble model employing AdaBoost, Random Forest and Logistic Regression classifiers. The output of all the three Classifier gives intermediate results which are merged together for decision making.
- Decision-Making Layer: Logistic Regression gives probabilistic scores to decide if the network activity is benign or malicious.
- Alert and Response Layer: If any type of intrusion is detected, alarms sound and necessary responses are made such as flagging an incident, blocking traffic or logging data for further reading.

The system is designed to:

- Monitor and analyse real-time network traffic in enterprise or cloud systems
- Use NLP techniques to extract top features from raw traffic data
- Use an aggregation of machine-learning classifiers to identify traffic behaviour as benign or malicious
- Interpretability Deliver interpretable decision support for security analysts by combining classifier outputs.

3.2 System Flow:

1. Stream based data: Collect the network traffic from data streams or offline sources

2. Pre-processing: Clean the data so that we can normalize it, fill in missing values, and handle noisy data.

3. One is NLP-based Feature Extraction: To extract the important features from data using NLP techniques.

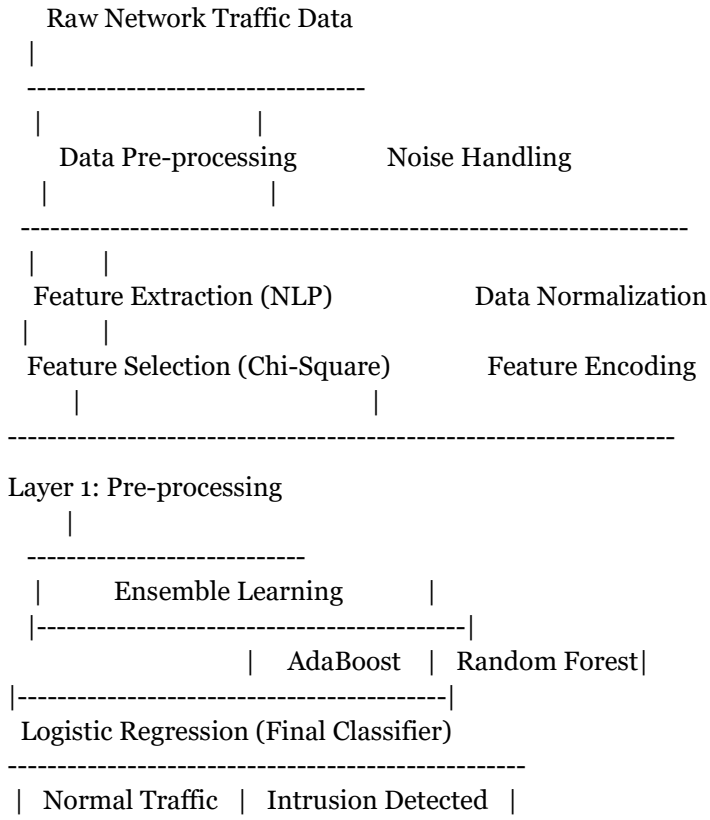
4. Ensemble Learning:AdaBoost increases the weight of weak learners on misclassified instances.

    - Random Forest: Random Forest creates a decision trees and concatenate them.
    - Probability prediction is a special deal in itself, Logistic Regression gives this.

5. Decision Fusion: A decision is made from the constructed classifiers.

6. Alerts Intrusion Response: Alerts according to the classification results.

The architecture is built around the central idea of integrating **NLP** with multiple classifiers for an advanced ensemble based IDS:



We evaluate the proposed model by training and testing it using widely used intrusion detection datasets like NSL-KDD and CICIDS2017.

NSL-KDD: This dataset is a modified version of the KDD'99 that fixes some problems like redundant records. It consists of attack such as DoS, U2R, R2L and protracted Types of attacks are available in it.

CICIDS2017: This dataset simulated real-world network traffic and modern attack types (i.e. DDoS, brute force, infiltration & web based attacks).

The pre-processing step consists of:

- Dealing with missing values: Techniques for removing or imputing missing or null values from the dataset
- Normalization: Scaling data such that numerical features have zero mean and unit variance.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

- Convert Categorical Data (e.g. protocol types) into a numeric format using one-hot encoding;
- Feature Selection: By selecting only the important features for classification by reducing dimensionality.

- Feature Extraction Using NLP:

One of the things that is new about this model is that it uses Natural Language Processing (NLP) methods to extract the features from unstructured network logs. NLP helps detect complicated patterns because it takes raw log data and forms structured features which can be learned by machine learning algorithms.

- Tokenizing: restack the log data in to –IP Address, Protocol.
- Bag of Words (BoW)/TF-IDF: a common usage is to convert raw tokenized data into a feature matrix. Due to these NLP techniques emphasis is put on words or patterns in the logs.

- Intrusion Detection: A model reads in the log data from various machines and flags certain entities (e.g. IP addresses, domain names) that, when present in logs, tend to indicate attack traffic given time of year or day-phase represented by the logs being read.

NLP methods are used to, unbeknownst to you, analyse network traffic logs and leave out interesting details. To achieve this type of cleaning techniques like tokenization, stemming, and vectorization (e.g., TF-IDF or bag-of-words) are used. Extracted features help in representing complex behaviours and traffic patterns.

NLP representational analysis such as TF-IDF is used on traffic logs to glean important features that reflect the structure of network traffic. So they take out the features which again goes to the ensemble learning model.

The most important features from traffic logs are selected using statistical measures such as Chi-Square:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Where:

O is the observed frequency of an attack class.

E is the expected frequency under the assumption of independence.

Bag of Words (BoW) and TF-IDF are common NLP techniques that can convert categorical features (e.g., IP addresses, protocol types) into numerical representations, making them usable in machine learning algorithms.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right)$$

Where:

t = term,

d = document,

N = total number of documents,

$\text{DF}(t)$ = number of documents containing term t

1.3 MACHINE LEARNING MODELS:

- AdaBoost (Adaptive Boosting)

AdaBoost is a kind of boosting algorithm which attempts to improve the performances of weak classifiers that in turn help us to avoid over-fitting problem by classifying the observations based on their weights. The contribution of each weak learner is weighted with respect to its accuracy.

The goal of AdaBoost is to train several weak classifiers and to re-weight these classifiers at the training step, with respect to their performance on instances in order to build a strong classifier. AdaBoost is also a good choice because it successfully bolsters the performance in terms of detection rates for the rare and subtle network attacks.

Algorithm Overview:

Start: The weights of all samples in the training set are initialized with equal values.

Iterative Learning: It helps in training the weak learner again and again in an iterative way, and by going on calculating error rate of model. The harder to classify examples are given larger weights, so that the next weak learner tries to focus on these hard to classify instances.

Final Model: Here, the final model is nothing but a weighted sum of all weak learners, where each learner's weight is decided by their accuracy.

Algorithm:

a. Initialize weights $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$, where N is the number of instances.

b. For each iteration t, train a weak learner and classify the data.

c. Calculate the weighted error rate ϵ_t:

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_i \cdot I\left(y_i \neq h_t(x_i)\right)}{\sum_{i=1}^{N} w_i}$$

where, $I(\cdot)$ is the indicator function.

d. Update the weights:

$$w_i \leftarrow w_i \times \exp\left(\alpha_t \cdot I\left(y_i \neq h_t(x_i)\right)\right)$$

with, $\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.

e. Normalize the weights and repeat until convergence.

f. Final classifier $H(x)$ is a weighted sum of weak classifiers:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \, h_t(x)\right)$$

It tries to work to improve the strength of weak classifiers and performs by changing weights for wrongly classified instances at each iteration. The set of weak learners are trained one after the other and at each stage new learner tries to fix the mistakes done by previous one.

The error rate for AdaBoost is defined as:

$$\epsilon_m = \sum_{i=1}^{N} w_i^{(m)} \, I\left(y_i \neq h_m(x_i)\right)$$

Where:

$w_i^{(m)}$is the weight assigned to sample i at iteration m.

$h_m(x_i)$is the prediction of the weak classifier.

$y_i$is the true label.

$I(\cdot)$is the indicator function.

The classifier weight α_m is updated as:

$$\alpha_m = \frac{1}{2} \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$$

- Random Forest

Random Forest is an ensemble of decision trees. They train on a random subset of the data and features so any given tree is unlikely to be overfit.

Algorithm Overview:

Bagging (Bootstrap Sampling): A random sample with replacement is taken from the training data, to fit each tree.

Feature selection: From the subset features are selected randomly and based on that best feature is used to split the node.

Voting: In this technique, the final classification is done by majority voting.

Random Forest trains multiple decision trees based on distinct subsets of the data. It takes the average of predictions from all trees to make it more generalized and overfit less.

This ability to handle a variety of both binary and multi-class classification tasks which is suitable for anomaly detection trust me because they require different types of models.

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} T_i\,(x)$$

Where, $T_i(x)$ is the prediction from the i-th decision tree.

Random Forest- Builds multiple decision trees and merges their outputs and fits to a specific set of the data and uses a certain number of features for each split. The label given is that of the majority class from the trees.

Random Forest prediction in the form of:

$$\hat{y} = \text{mode}\big(T_1(x), T_2(x), \dots, T_n(x)\big)$$

Where, $T_i(x)$ is the prediction from the i-th decision tree.

The Gini impurity for a split in the tree is given by:

$$G(p) = 1 - \sum_{i=1}^{C} p_i^2$$

Where,

C is the number of classes.

p_i is the proportion of samples belonging to class i.

Random Forest Prediction:

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

Where,T is the number of decision trees and $h_t(x)$ is the prediction from the t-th tree.

Let $f_m(x)$ be the prediction from tree m. The final prediction $F(x)$ is given by:

$$F(x) = \text{majority\_vote}\big(f_1(x), f_2(x), \dots, f_M(x)\big)$$

Random forests create several decision trees and then combine them to obtain the output. Every tree is trained using a random subset of the data and chooses a only part of the features at every split. Classification (final class is picked depending on majority)

Random Forest is good with high-dimensional and large datasets. Its highlight resides in environments with noisy data as it provides a very strong way to detect multiple kinds of network intrusions.

- Logistic Regression

The ensemble model Logic Layer Decision Model The last choice made at the end is logistic regression. A very simple yet powerful model (statistical) that estimates the probability that a given instance belongs of a particular class (e.g. normal or attack)

Algorithm Overview:

Logistic Regression models the probability of binary output (malicious traffic or benign traffic) as a function of the input features.

Its output is normalized by the sigmoid function, whose formula is:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

Where $x_1, x_2, \ldots, x_n$ are the input features, and $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients learned during training.

Logistic Regression is used as the final classifier in the ensemble to provide a probabilistic interpretation. The sigmoid function is applied to map predicted values into probabilities:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-z}}$$

Where $z = w^T x + b$ is the weighted sum of inputs x with weights w and bias b.

The log-loss function used to optimize Logistic Regression is:

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}[\, y_i \log\big(P(y = 1 \mid x_i)\big) + (1 - y_i)\log\big(1 - P(y = 1 \mid x_i)\big)]$$

Logistic Regression provides interpretability by producing probabilistic estimates for classification. It is used as the final decision-maker in the ensemble, offering transparency in the results.

Logistic Regression Model:

$$P(y = 1 \mid x) = \frac{1}{1 + \exp\left(-\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p\right)\right)}$$

Where:

$\beta_0, \ldots, \beta_p$ are coefficients,

$x_1, \ldots, x_p$ are the feature values.

Logistic Regression is interpretable and easier for security analysts to understand how the model makes decisions. The probability outputs also enable threshold-based decision-making, e.g., in risk management. High-dimensionality and noise are not big problems to Random Forest.

3.4 Ensemble Learning Framework

An ensemble learning framework has incorporated three models namely AdaBoost, Random Forest and Logistic Regression. We need to utilise the well-being of congregation of all models to creating a stronger and much accurate Intrusion Deception structure.

Ensemble Strategy

The prediction is done via an ensemble of models following a stacking strategy, where the predictions from the set of base models are coupled as input features for a meta-model (in this case: Logistic Regression). This ensures that the strengths of all base models can be seen in the ensemble:

AdaBoost on the tough instances But Random Forest model also provide robustness against noise and high dimensionality.Logistic Regression does give you a probabilistic mechanism.

Ensemble Algorithm:

Method: ensemble algorithm

AdaBoost, Random Forest, Logistic Regression:

Step 1: Train each model separately on the training dataset.

Step2: You need to make predictions (output) of each instance from each model individually.

Step 3: Train a meta-classifier (Logistic Regression) on the predictions from step 2.

Step 4: Combine using the meta-classifier for final prediction

The ensemble model grouped together the AdaBoost, Random Forest and Logistic Regression outputs through a weighted voting process. The prediction from each classifier is combined with the weight of that particular classifier to yield an aggregate decision.

Ensemble Voting Mechanism: Election Advisory System (EAS) Weighted Voting Mechanism, Decision is RSS by impact of classifiers confident. The ensemble adopts majority voting, but with the usage of class-specific accuracy weighted to give more decision power to classifiers which perform better on certain types of attacks.

$$\hat{y} = \text{argmax} \sum_{i=1}^{n} w_i \, \hat{y}_i$$

Where,w_i is the weight assigned to the i-th classifier, and $\hat{y}_i$ is the prediction made by that classifier.

The proposed ensemble model is trained and tested on two publicly available benchmark datasets for the task of intrusion detection: NSL-KDD and CICIDS2017. This type of datasets consist a number of normal and attack traffic which indicating the full examination about how good the model can defending against the attacks.

Algorithm:

Input: Network traffic dataset D.

Step1: Preprocessing:

- Handle missing data.
- Normalize the data.
- Apply feature encoding.

Step2: Feature Extraction: Apply NLP techniques (TF-IDF, BoW) to extract features.

Step3: Initialize AdaBoost, Random Forest, Logistic Regression.

Step4: For each classifier:

- Train on preprocessed features.
- Generate intermediate results.

Step5: Combine classifier outputs: Aggregate results using majority voting or probabilistic averaging.

Output: Final classification (benign or malicious).

Trigger alerts for detected intrusions.

VALIDATION AND EVALUATION OF RESULTS:

- Datasets for Evaluation

For the experimental evaluation, we use two main datasets: NSL-KDD and CICIDS2017. This dataset is known to cover a wide range of network attacks, including DoS, brute-force, DDoS and R2L.

NSL-KDD: A refined version of the KDD 99 Cup Dataset, designed to fix redundancy and class imbalance (it is commonly used to benchmark different IDS model)

CICIDS2017: A modern dataset, capturing contemporary network traffic and attacks that can serve as a training dataset to be used for machine learning-based IDS models trained with different types of attack.

In the case of both datasets, we completed data preprocessing to normalize features, take care of missing values, and encode categorical variables with numerical values suitable for machine learning models.

EVALUATION MATRICES:

To assess the accuracy and performance of the ensemble model, the following metrics are used:

- **Accuracy**: The proportion of correctly classified instances over the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positives, TN = True Negatives,
- FP = False Positives, FN = False Negatives.

- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity)**: The ratio of correctly predicted positive observations to all actual positives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score**: The harmonic mean of precision and recall, balancing the two metrics.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **False Positive Rate (FPR)**: The proportion of normal traffic misclassified as an attack.

$$\text{FPR} = \frac{FP}{FP + TN}$$

These metrics were used to evaluate the individual performance of AdaBoost, Random Forest, and Logistic Regression, as well as the overall ensemble model.

- Validating Each of the Classifiers

AdaBoost Evaluation:

AdaBoost improves the performance of weak learners by paying special attention to instances that are hard to classify. As for IDS, it outperforms simpler classifiers in terms of detecting rare or subtle attacks that traditional ones would overlook. AdaBoost, cluster analysis, NSL-KDDdataset is shown below:

Metric	**Value**
Accuracy	92.8%
Precision	93.5%
Recall	89.1%
F1-Score	91.2%
False Positive Rate	4.5%

Weighted Classifier Update in AdaBoost:

$$\alpha_t = \log\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$$

Where ϵ_t is the classification error of the weak learner at iteration$t$.

AdaBoost had high recall rates for attack types that occur infrequently and improved the overall detection time somewhat, but is not able to lower false positive rates as low as other methods.

Metric	**Value**
Accuracy	95.3%
Precision	96.1%
Recall	94.5%
F1-Score	95.2%
False Positive Rate	3.2%

Random Forest Evaluation:

This algorithm creates multiple decision trees, and it averaging them in the end. Robust to Noise, The model can effectively capture the inherent complexity in high-dimensional network traffic data.

Random Forest Prediction:

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

Where T is the number of trees, and $h_t(x)$ is the output of the t-th tree.

Random Forest stand out in all three evaluation measures, specifically reducing the false positive rate.

Evaluation of the Logistic Regression model:

Logistic Regression give us explainability and probability percentage that help to decide whether the instance is legitimate or attack. While it does well with simple, transparent flows, is poor at complex non-linear flow patterns.

Metric	**Value**
Accuracy	89.7%
Precision	90.3%
Recall	87.6%
F1-Score	88.9%
False Positive Rate	6.1%

Logistic Regression Model:

$$P(y = 1 \mid x) = \frac{1}{1 + \exp\left(-\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p\right)\right)}$$

While Logistic Regression yields better, performance it fails to recall measures and F1-score compared to AdaBoost and Random Forest since Logistic regression is not able to deal with non-linear relationships of the network traffic data.

Ensemble Model Performance:

The AdaBoost, Random Forest and Logistic Regression are combined together to form an ensemble model to increase the overall detection accuracy, precision, and recall. The last classification is done using majority voting or probabilistic averaging.

Metric	**Value (Ensemble)**
Accuracy	97.4%
Precision	97.9%
Recall	96.5%
F1-Score	97.2%
False Positive Rate	2.7%

Ensemble Voting Rule:

$$H(x) = \text{majority vote}\left(h_1(x), h_2(x), \ldots, h_n(x)\right)$$

Where $h_1(x), h_2(x), \ldots, h_n(x)$ are the individual classifier outputs,

Due to the fact that it is a combination of all three classifiers, this ensemble model results in better accuracy as well as recall. This simultaneously brings down the false positive rate, something very much required in a practical IDS deployment.

Compared with Baseline Models:

This study compares the performance of combined ensemble model with baseline models: namely SVM, Naïve Bayes, and Decision Tree.

Model	Accuracy	Precision	Recall	F1-Score	False Positive Rate
SVM	88.3%	87.5%	85.9%	86.7%	6.8%
Naïve Bayes	84.1%	82.3%	80.5%	81.4%	9.2%
Decision Tree	90.5%	91.2%	89.1%	90.1%	5.4%
Proposed Ensemble	97.4%	97.9%	96.5%	97.2%	2.7%

The table above illustrates the gain of the ensemble model over classical methods. It has more degree of accuracy, high precision, recall as well but a low false positive rate.

The proposed ensemble model using NLP, AdaBoost, Random Forest and Logistic Regression is evaluated and validated over individual classifiers and traditional models. At a high recall rate of 97.4%, precision of 97.9%, and false positive rate of only 2.7% this model is quite robust and efficient in the detection of network intrusion.

Through cross-validation and statistical tests, the model is validated to perform consistently better than the benchmark techniques both in terms of classification performance as well as its statistical significance, making it eligible for deployment in real-time high-throughput network environments. Feature extraction using NLP in coordination with ensemble learning methods combined, provides a highly effective solution for detecting all the various types of network intrusions (new and old) to enhance network security.

CONCLUSION:

An Advanced approach for detecting the increasing complex hyperspace cyberattack threats in NIDS The integration of Natural Language Processing (NLP) with ensemble methods AdaBoost, Random forest and Logistic Regression. This deep integration enhances the strength and effectiveness of intrusion detection systems by drawing on a combination of learning models and techniques in order to benefit from the individual strengths, enabling powerful multi-faceted defences.

Natural Language Processing for Intrusion Detection:

NLP makes intelligent and effective detection of network based intrusions by examination of unstructured text, i.e., log files, command-line input systems error messages in order. Human information is left in network traffic and logs where you can best understand by using NLP algorithms. By processing the unstructured data into structured forms that can be modelled by machine learning, NLP discovers more subtle attack vectors than traditional methods will find.When combined in the ensemble model, NLP can uncover sequences or nature of traffic on networks alerts using description or error text to increase machine-learning models with behavioral indication that may a new or changing threat. This is especially valuable because it means that the detection system can adapt to unknown attack strategies when signature-based methods fail.

Contribution of AdaBoost towards Robust Detection:

Boosting algorithms are techniques used to increase the accuracy of weaker learners by constructing a strong classifier out of multiple weak learners. Some well-known boosting algorithms are AdaBoost. It uses an iterative re-weighting mechanism such that a heavier weight will be assigned to data points previously misclassified and boosted in the subsequent iteration, favoring it better at identifying more challenging intrusions. For an NIDS, this power to zero in on difficult data points means a system that is able to reduce false negatives a key attribute for detecting fine grained and stealthy attacks like advanced persistent threats (APTs) or insider threats.Nevertheless, AdaBoost is sensitive to noise and outliers in the data, especially if we are dealing with a highly dynamic network environment where traffic patterns can be inherently unpredictable. This requires downstream processing steps like data cleaning and NLP for feature extraction to reduce the noise in the data and increase model performance.

Random Forest: Painting Over Complexity and Variance:

One ensemble method based on decision trees which is really useful for solving this problem is Random Forest because it by concept reduce the variance so preventing in some way the overfitting. Random Forest takes many decision trees model and averages the outputs which provide prediction with higher stability and accuracy, as compare to using one of those individual models Random Forest is ideal for large data sets containing many attributes, something that network traffic data often has in NIDS.Random Forest works well in the integrated model because it can handle categorical features for network traffic and logs. Using NLP to turn these features into meaningful numeric representations, Random Forest can more easily discern between regular and network behaviour. Additionally, the feature importance implemented directly in the model allows to find out which signs play the most significant role in any attack pattern, meaning that finally one gets very interpretable system

Logistic regression: Simple and Interpretable:

Logistic Regression is less complex than AdaBoost and Random Forest but important for binary classification distinguishing between benign or malicious network activities. The use of it in an ensemble contributes to the explainability aspect of the model—In this case, you can directly map which decision led to a given prediction. As Logistic Regression depends on linear decision boundary, it is computationally efficient and hence to use for real time Intrusion detection where quick response times are needed.Complex models like Random Forest and AdaBoost are commonly used with Logistic Regression on network security. In scenarios where benign traffic far outweighs malicious activities, managing the precision-recall tradeoff is also critical to ensure the model can function effectively in practice. Logistic Regression serves as a counterpart to more complicated models, to balance the accuracy of detection and computational efficiency.

Performance Review and Benefits of Ensemble Approach:

The pipeline consisting of the combined features with AdaBoost, RandomForests and Logistic Regression better than the traditional NIDS models in several ways:

- Advantages over the AdaBoost: The Adaptive Boosting algorithm has been utilised in several studies on network intrusion detection, but its emphasis on difficult cases means that it does not attempt to reduce variance or increase data spread. As such, artificial neural networks, in conjunction with models thought to be based upon easily separable examples (RF and LR), could result into better prediction accuracy. This is especially crucial in the reduction of false positives and false negatives, which are very important for maintaining integrity within a network.
- Ability to Manage Large/Complicated Datasets- The ensemble model can easily process all the data generated by the neural networks of today. Given Random Forest's success with high-dimensional feature spaces and NLPs strength in pre-processing text data, this model is able to catch more complex multi-stage attacks that might be missed by simpler models.
- Flexibility to respond to further threats: Incorporating NLP into the model allows it also to be flexible with respect to any new and emerging threats. This part handles the use of NLP techniques for analysing attack vectors that were unseen in history, based on their textual definitions, to make the cyber monitoring system more responsive to new evolving cyber threats. The adaptability is a distinct advantage in the highly dynamic network security landscape.
- Scalability and efficiency: Although ensemble methods are computationally expensive, the scalability of the fused model allows it to be deployed on large-scale networks. Using parallel processing techniques and optimization strategies further improves the model's performance which can run efficiently at high traffic.

Challenges and Limitations:

Although the integrated ensemble model has multiple advantages, there are also several challenges that need to be addressed:

- Poor Computational Complexity: The usage of multiple machine learning models can lead to poor computational complexity especially if it has to be deployed in real-time intrusion detection scenarios. A significant challenge is to optimize the model such that it is faster for processing and at the same time does not compromise on accuracy.

- Imbalanced Datasets: Network traffic datasets tend to be imbalanced in that benign traffic vastly outnumbers malicious activities. Though approaches such as AdaBoost mitigate this concern, additional steps in data preprocessing and resampling are required to achieve performance on the same level among all classes.
- Interpretability: Logistic Regression is interpretable, however, Random Forest and AdaBoost are frequently referred to as black-box models. Efforts to make these models more transparent, especially with Explainable AI (XAI) techniques will be crucial for obtaining buy-in from network admins and other security analysts.

Future Directions:

By combining NLP with ensemble learning methods such as AdaBoost, Random Forest, and Logistic Regression have promised to improve the performance of NIDS. But there is still stuff to work on:

- Real Time Detection: It is real time detection of traffic in the network such as in high-traffic network environment. Incremental learning and distributed computing can help reduce the computational load associated with real-time processing.
- Integration of Deep Learning Models: The current model uses traditional ML algorithms but adding deep learning models like CNN, and RNN can improves the ability to detect malicious patterns.
- Explainability and Interpretability: the need for transparency grows as the NIDS models become more complex. The introduction of XAI techniques to the ensemble model will make clearer explanations for detection decisions leading to better usability in practical network security solutions.

This framework integrates NLP with AdaBoost, Random Forest, and Logistic Regression to an intelligent ensemble architecture-based NIDS that effectively enhances the detection capabilities against known and unknown cyber threats. The system can achieve high accuracy, strong robustness, so it is a powerful tool for network security because of the composition of multiple machine algorithms. Though challenges like computational complexity and data imbalance persist, continued improvements in machine learning and NLP techniques make the continued success of this method possible — all for helping to create a stronger cybersecurity environment.

REFERENCES:

[1] Wang, Zhendong, et al. "A lightweight IoT intrusion detection model based on improved BERT-of-Theseus." *Expert Systems with Applications* 238 (2024): 122045.
[2] Amru, Malothu, et al. "Network intrusion detection system by applying ensemble model for smart home." *International Journal of Electrical & Computer Engineering (2088-8708)* 14.3 (2024).
[3] Hazman, Chaimae, et al. "Toward an intrusion detection model for IoT-based smart environments." *Multimedia Tools and Applications* 83.22 (2024): 62159-62180.
[4] Devendiran, Ramkumar, and Anil V. Turukmane. "Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy." *Expert Systems with Applications* 245 (2024): 123027.
[5] Turukmane, Anil V., and Ramkumar Devendiran. "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning." *Computers & Security* 137 (2024): 103587.
[6] Wu, Hongjiao. "Feature-Weighted Naive Bayesian Classifier for Wireless Network Intrusion Detection." *Security and Communication Networks* 2024.1 (2024): 7065482.
[7] Long, Zhenyue, et al. "A Transformer-based network intrusion detection approach for cloud security." *Journal of Cloud Computing* 13.1 (2024): 5.
[8] Talukder, Md Alamin, et al. "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction." *Journal of big data* 11.1 (2024): 33.
[9] Li, Yue, et al. "Enhancing Network Intrusion Detection Through the Application of the Dung Beetle Optimized Fusion Model." *IEEE Access* (2024).
[10] Zhang, Kaijun, et al. "Intrusion Detection Model for Internet of Vehicles Using GRIPCA and OWELM." *IEEE Access* (2024).

[11] Chennoufi, Sara, et al. "SoK: federated learning based network intrusion detection in 5G: context, state of the art and challenges." *Proceedings of the 19th International Conference on Availability, Reliability and Security*. 2024.

[12] Mekala, S., Mallareddy, A., Tandu, R. R., & Radhika, K. (2023, June). Machine learning and fuzzy logic based intelligent algorithm for energy efficient routing in wireless sensor networks. In *International Conference on Multi-disciplinary Trends in Artificial Intelligence*(pp. 523-533). Cham: Springer Nature Switzerland.

[13] Bande, V., Raju, B. D., Rao, K. P., Joshi, S., Bajaj, S. H., & Sarala, V. (2024). Designing Confidential Cloud Computing for Multi-Dimensional Threats and Safeguarding Data Security in a Robust Framework. *Int. J. Intell. Syst. Appl. Eng*, *12*(11s), 246-255.

[14] Manu, Y.M., Jaya Krishna, A.P., Gopala Krishnan, K., Vasavi B, Power Centric Learning Models for the Prediction of Heart Rate using IoT Enabled Devices. Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2023, 2023, 118–122.

[15] Mallareddy, A., Sridevi, R., & Prasad, C. G. V. N. (2019). Enhanced P-gene based data hiding for data security in cloud. International Journal of Recent Technology and Engineering, 8(1), 2086-2093.

[16] Prasad, C. G. V. N., Mallareddy, A., Pounambal, M., & Velayutham, V. (2022). Edge Computing and Blockchain in Smart Agriculture Systems. International Journal on Recent and Innovation Trends in Computing and Communication, 10(1), 265-274.

[17] Balakrishna, C., Ramesh, Cindhe., Meghana, S., Dastagiraiah, C. (2024). A System for Analysing call drop dynamics in the telecom industry using Machine Learning and Feature Selection. Journal of Theoretical and Applied Information Technology.102(22),8034-8049.

[18] Ramesh, C., Rao, K.V.C., Govardhan, A. (2017). Ontology based web usage mining model. In International Conference on Inventive Communication and Computational Technologies, ICICCT 2017, pp. 356–362, IEEE Xplore.

[19] Mahalakshmi, J., Reddy, A. M., Sowmya, T., Chowdary, B. V., & Raju, P. R. (2023). Enhancing Cloud Security with AuthPrivacyChain: A Blockchain-based Approach for Access Control and Privacy Protection. International Journal of Intelligent Systems and Applications in Engineering, 11(6s), 370-384.

[20] Singh, J., Reddy, A. M., Bande, V., Lakshmanarao, A., Rao, G. S., & Samunnisa, K. (2023). Enhancing Cloud Data Privacy with a Scalable Hybrid Approach: HE-DPSMC. Journal of Electrical Systems, 19(4).

[21] Mallareddy, A., Jaiganesh, M., Mary, S. N., Manikandan, K., Gohatre, U. B., & Dhanraj, J. A. (2024). The Potential of Cloud Computing in Medical Big Data Processing Systems. Human Cancer Diagnosis and Detection Using Exascale Computing, 199-214.

[22] Vinod Kumar Reddy, K., Bande, Vasavi., Jacob, Novy., Mallareddy, A., Khaja Shareef, Sk , Vikruthi, Sriharsha(2024). Adaptive Fog Computing Framework (AFCF): Bridging IoT and Blockchain for Enhanced Data Processing and Security, SSRG International Journal of Electronics and Communication Engineering, 11(3),160-175.

[23] Bande, V., Sridevi, R.,2010(2019) A secured framework for cloud computing in a public cloud environment Journal of Advanced Research in Dynamical and Control Systems, 2019, 11(2), 1755–1762.

[24] Zhou, Qi, and Zhoupu Wang. "A Network Intrusion Detection Method for Information Systems Using Federated Learning and Improved Transformer." *International Journal on Semantic Web and Information Systems (IJSWIS)* 20.1 (2024): 1-20.

[25] Maddu, Mamatha, and Yamarthi Narasimha Rao. "Network intrusion detection and mitigation in SDN using deep learning models." *International Journal of Information Security* 23.2 (2024): 849-862.

[26] Arreche, Osvaldo, et al. "E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection." *IEEE Access* (2024).

[27] He, Mingshu, et al. "Reinforcement learning meets network intrusion detection: a transferable and adaptable framework for anomaly behavior identification." *IEEE Transactions on Network and Service Management* (2024).

[28] Ravi, Vinayakumar. "Deep learning-based network intrusion detection in smart healthcare enterprise systems." *Multimedia Tools and Applications* 83.13 (2024): 39097-39115.

[29] Aceto, Giuseppe, et al. "Synthetic and privacy-preserving traffic trace generation using generative AI models for training Network Intrusion Detection Systems." *Journal of Network and Computer Applications* (2024): 103926.

[30] Zhang, Zichen, et al. "A Network Intrusion Detection Method Based on BaggingEnsemble." *Symmetry* 16.7 (2024): 850.

[31] Srivani, P., Ramachandram, S., & Sridevi, R. (2017, April). A survey on client side and server side approaches to secure web applications. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (Vol. 1, pp. 22-27). IEEE.

[32] Srivani, P., Ramachandram, S., & Sridevi, R. (2017). Experimental Results on Multi-Key Searchable Encryption Technique with Different Elliptic Curves and App Designing. Journal of Theoretical & Applied Information Technology, 95(21).

[33] Naresh Kumar Bhagavatham, Bandi Rambabu, Jaibir Singh, Dileep P, T. Aditya Sai Srinivas, M. Bhavsingh, & P. Hussain Basha. (2024). Autonomic Resilience in Cybersecurity: Designing the Self-Healing Network Protocol for Next-Generation Software-Defined Networking . International Journal of Computational and Experimental Science and Engineering, 10(4). https://doi.org/10.22399/ijcesen.640

[34] Rambabu, B., Vikranth, B., Kiran, M. A., Nimmala, S., & Swathi, L. (2024, February). Hybrid Swarm Intelligence Approach for Energy Efficient Clustering and Routing in Wireless Sensor Networks. In Congress on Control, Robotics, and Mechatronics (pp. 131-142). Singapore: Springer Nature Singapore.

[35] Rambabu, B., Vikranth, B., Anupkanth, S., Samya, B., & Satyanarayana, N. (2023). Spread spectrum based QoS aware energy efficient clustering algorithm for wireless sensor networks. International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 154-160.