Adaptive and Resilient Machine Learning Model for Anomaly and Intrusion Detection in Modern Networks

Submitted By

Md. Mosfiqur Rahman
(212-35-3184)
Department of Software Engineering

Supervised By

Dr. A. H. M. Saifullah Sadi
Professor
Department of Software Engineering

Thesis submitted in fulfilment of the requirements
for the award of the degree of Bachelor of Science in Software Engineering
Fall 2025

APPROVAL

This thesis titled on "**Adaptive and Resilient Machine Learning Model for Anomaly and Intrusion Detection in Modern Networks**", submitted by **Md. Mosfiqur Rahman (ID: 212-35-3184)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Dr. A. H. M. Saifullah Sadi
Professor
Department of Software Engineering
Faculty of Science and Information Technology Daffodil International University

Chairman

Dr. Rubaiyat Islam
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Dr. Md. Abdul Kader
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Nuruzzaman Faruqui
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Md. Mostafiz Khan
Managing Director
Tecognize Solutions Limited

External Examiner

DECLARATION

I hereby declare that the work presented in this thesis is my own original research work, carried out under the supervision of **Dr. A. H. M. Saifullah Sadi,** Professor, Department of Software Engineering, Daffodil International University.

This work has not been submitted anywhere, either in whole or in part, for any degree, diploma, or publication in this or any other university. All sources of information used in this thesis have been duly acknowledged.

Supervised By

Dr. A. H. M. Saifullah Sadi
Professor
Department of Software Engineering

Submitted By

Md. Mosfiqur Rahman

Md. Mosfiqur Rahman
ID: 212-35-3184
Department of Software Engineering

Abstract

Due to increased complexity and frequency of attacks, we should leave behind the old signature-based systems used to detect networks and adopt newer and dynamic detection systems of anomalies. The existing static machine learning models are poor at adapting to the network environment; they are not well able to generalize to new threats and performance decays when network patterns evolve. Our study addresses these issues through designing and thoroughly developing a new Hybrid Multi-Layered Stacking Ensemble Model that is robust and resilient in anomaly detection in the network. The approach uses the combination of five various classifiers, K-nearest neighbors, Gradient Boosting, support vectors machine, random forest and logistic regression, to enhance variety as well as minimize errors. To test the model with the recent real world CICIDS 2017 data that provides numerous older and recent attacks such as DDoS, PortScan, and Botnet and the NFS-2023-TE data that concentrates on the new IoT/ Edge environments, we tested the model on both. A careful testing in terms of accuracy, precision, recall, and F1-score indicates that the stacking method is superior. More specifically, the hybrid model achieved 98.79% accuracy on the CICIDS 2017 dataset which is understandably higher than that of individual learners. Despite the perfect detection of large attacks such as DDoS, PortScan (F1 -score 1.00), the model has a significant limitation in identifying small, low-impact attacks such as Bot attacks (F1 -score 0.55, recall 0.38) which, we took a closer look at. These findings validate the fact that the combination of various ensembles increases the strength of the system. Future research will also include the deep learning models to enhance feature extraction, develop real-time drift adaptation and investigate blockchain based federated learning to provide secure and collaborative resilience in distributed networks.

Table of Contents

ABSTRACT ---- **IV**

TABLE OF CONTENTS ---- **V**

LIST OF TABLES ---- **VI**

CHAPTER 1 ---- **1**

Introduction ---- **1**

1.1 Background and Context: The Evolving Landscape of Modern Network Security ---- 1

1.2 Motivation for Adaptive Intrusion Detection Systems (NIDS) ---- 1

1.3 Limitations of Static Machine Learning in Dynamic Environments ---- 2

1.4 Research Objectives and Scope of Investigation ---- 3

1.5 Core Contributions to the Field of Network Cybersecurity ---- 3

1.6 Organization and Structure of the Thesis ---- 4

CHAPTER 2 ---- **5**

Literature Review ---- **5**

2.1 Foundations of Network Intrusion and Anomaly Detection Paradigms ---- 5

2.2 Review of Single-Classifier Machine Learning Approaches in NIDS ---- 6

2.2.1 Discriminative Models: K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) ---- 6

2.2.2 Tree-Based Ensemble Methods: Random Forest (RF) and Gradient Boosting (GB) ---- 6

2.2.3 Probabilistic Model: Logistic Regression (LR) ---- 7

2.3 Advanced Ensemble Learning Techniques for Enhanced Robustness ---- 7

2.4 Analysis of Benchmark Datasets in Network Security Research ---- 8

2.5 Identification of Research Gaps and Thesis Justification ---- 9

CHAPTER 3 ---- **10**

Methodology ---- **10**

3.1 Architectural Overview of the Proposed Network Detection System (NDS) ---- 10

3.2 Data Acquisition and Preprocessing Methodology ---- 11

3.3 Mathematical Formulation of Base Classification Algorithms (Layer 1) ---- 13

3.4 Detailed Stacked Generalization and Ensemble Design ---- 16

3.4.1 Principle of Hard and Soft Voting Ensemble ---- 16

3.4.2 Multi-Layered Stacking Architecture Rationale (Hybrid Model) ---- 16

3.5 Performance Evaluation Metrics ---- **17**

CHAPTER 4 ---- **19**

Results & Discussion ---- **19**

4.1 Experimental Environment and Implementation Details ---- 19

4.2 Baseline Performance of Individual Base Classifiers ---- 19

4.3 Comparative Performance of Voting Ensembles ---- 24

4.3.1 Hard Voting Ensemble Analysis ---------- 24
4.3.2 Soft Voting Ensemble Analysis ---------- 25
4.4 Empirical Results of the Hybrid Stacking Model ---------- 26
4.5 Summary of Overall Detection Accuracy ---------- 29
CHAPTER 5 ---------- 31
Conclusion ---------- 31
REFERENCES ---------- 33

LIST OF TABLES

Table 1: Performance Comparison Of Accuracy For Base Models 19
Table 2: Stacking Classification Report 27

Chapter 1

Introduction

1.1 Background and Context: The Evolving Landscape of Modern Network Security

The extensive expansion of information technology within governments, corporations and households has altered cybersecurity. Cloud computing, IoT devices and edge systems are modern networks that provide numerous additional avenues of attack. Owing to this, the cyber threats have evolved. They have been massive, noticeable attacks such as DDoS, but this time around, they are silent, consistent and difficult to detect such as APTs and advanced Botnets. This demonstrates that the outdated security solutions are not sufficient. The old system of Network Detection adheres to predetermined rules or signatures. They are effective with familiar attacks, but not new or evolving attacks, termed a zero-day attack, as they have no signature. These defensive mechanisms are not desirable when attackers can rapidly develop new methods of attack.

That is alleviated by Machine Learning (ML). ML is able to learn on its own based on network traffic and identify anomalous behavior. This allows defenses to perform in advance of attack occurrence and maintain themselves without the need of constant human intervention. The modern cybersecurity requires a system that would be able to continue working even in cases when bad things occur or attackers switch strategies. To achieve that, we require Network Detection Systems which, on their own, can be adapted and remain robust even in case the data is highly dynamic, known as concept drift.

1.2 Motivation for Adaptive Intrusion Detection Systems (NIDS)

Machine learning (ML) is applied in network detection systems (NDS) since the system requires more accuracy and more stable performance. Although ML is a better technique than the older signature techniques, it is not completely flawless with actual network data. Those issues contain numerous features, time-varying, and the large disparity between cases of normality and attack. Certain models like Support Vector Machines (SVM) or Logistic Regression (LR) might struggle to distinguish overlapping classes that can result in excessive false alarms (False Positives) or false non-detection of attacks (False Negatives).

The Ensemble learning pools the forecasts of numerous less complex models. It tends to be more efficient as compared to a single model as it minimizes errors that a single model may have. The use of such methods as stacking and voting combines the powers of various algorithms to create a single powerful system. Such a layered strategy also makes the NDS stronger and capable of detecting a wider range of attacks, such as DDoS, various DoS attacks (Hulk, GoldenEye, Slowhttptest, Slowloris) and port scans, to a larger degree. The concept of using ensembles is not solely regarding the ability to capture more attacks, but also attempting to reduce essential errors. Reduced rate of false alarms implies that the operators will not be fatigued due to excess number of alerts. The fewer missed attacks will guarantee that real intrusions are not missed.

1.3 Limitations of Static Machine Learning in Dynamic Environments

Although machine learning promises a lot, recent fixed network intrusion detection (NIDS) models also possess a number of theoretical and practical issues that undermine their resiliency:

The main issue is that the tactics of attacks do not stand still. Due to this, the trends that a model learns may no longer be similar to the real traffic. A model that has been trained with regular business traffic (CICIDS 2017) is highly effective, but on Internet-of-things traffic (NFS-2023-TE) it errors significantly. K-Nearest Neighbors classifier achieved 98.28 per cent accuracy on CICIDS 2017 but dropped down to 91.00 per cent accuracy on NFS-2023- TE. The Logistic Regression was in fact going to improve to 93.00 per cent out of 85.59 per cent within the same shift. These developments demonstrate that one fixed model would not remain dependable in novel and evolving network circumstances; we require a system that is flexible. Serious failures in the detection of rare or subtle attacks may be hidden by high overall accuracy. Intrusions which are stealth, such as Botnet traffic appear as normal traffic. This brings about a huge disparity of bad traffic that is considerably lower. A combination of the stacking model achieved the highest overall accuracy of 98.79, which was quite low in detecting Botnet traffic with a perfect recall (1.00) and a very low recall (0.38) as well as an F1-score of 0.55. Meaning that large false negative rate implies that large numbers of real Botnet attacks will slip by thus compromising defense against sophisticated attacks that remain undetected.

Using more complex ML models should also be deployed based on the inferred computation consumption particularly fast real-time network streams. Many base classifiers in an ensemble may increase the accuracy but at the expense of overhead. The system of intrusion detection should therefore be able to identify the best features and narrow the dimensions to ensure that the selected model, including the part of stacking, can operate fast and yet the selected model can predict threats effectively and fast.

1.4 Research Objectives and Scope of Investigation

- To overcome the drawbacks of fixed, single-model approaches, this study has four objectives:
- Q1: Detection of network problems: Design and construct a new Hybrid Multi-Layered Stacking Ensemble Model to enhance five machine learning methods (KNN, GB, RF, SVM, LR) to detect network issues.
- O2: Test carefully the performance of this ensemble and its base classifiers on two real data sets, the well-known CICIDS 2017 set (DS1) and the newer set IoT/Edge NFS-2023-TE set (DS2). This will demonstrate the extent to which it can be generalized and remain robust in various settings.
- Q3: As part of advanced research standards, explain the choice of the selected base learners and the stacking method, how they are used, and how they combine results.
- O4: Conduct a more thorough class-by-class performance analysis, reducing false positives and false negatives, in particular, to rare attack types such as Botnets.

1.5 Core Contributions to the Field of Network Cybersecurity

The important contributions that this work made are:

- **Theoretical:** Obtained and applied a theory of heterogeneous ensembles (Stacked Generalization) adjusted to the multi-class problem of heterogeneous network traffic of varied network traffic problems, which demonstrates the stability and robustness of different algorithms.
- **Methodological:** Proposed and confirmed an excellent multi-layered hybrid much more effective and efficient integrating five distinct base classifiers through meta-learning providing a blueprint of a model of an NDS system in the future.

- **Empirical:** Demonstrated the hybrid stacking model achieved an accuracy of 98.79 on the CICIDS 2017 benchmark. It also compared it with the NFS-2023-TE set that was used to measure the gap in generalization that can offer valuable data in understanding the way models can adapt to the changing network environment.

1.6 Organization and Structure of the Thesis

The remainder of the paper is structured in the following way: Chapter 2 is the review of the existing literature on NIDS, machine learning approaches, and ensemble techniques. Chapter 3 describes the algorithms, such as data cleaning and feature generation, as well as the computation underlying the main algorithms. Chapter 4 gives the experimental design, complete findings, and compares base models, voting ensembles, and the ultimate hybrid stacking model. Chapter 5 assesses the findings, the strengths, weaknesses, and implications of the findings to adaptive resilience. Chapter 6 is the conclusion in which the main findings are pointed out, and the directions of future research are indicated.

Chapter 2

Literature Review

2.1 Foundations of Network Intrusion and Anomaly Detection Paradigms

The NIDS are of two types; signature-based, or misuse, often called, detection and the anomaly-based detection. Signature-based systems search the traffic or system calls in a signature database against known attack patterns. They are effective in detecting previously known threats, but they fail to detect new, zero-day attacks owing to the fact that they build on prior knowledge. Detection based on anomaly does not work in the same way. It constructs an image of normal network behavior and then marks as suspicious anything that is significantly different in its behavior. This approach requires machine learning since it allows the system to learn what normal behavior will look like using big sets of data. One of the most important issues is to maintain the system efficient in the face of a zero-day attack and concept drift [1]. New types of attacks may emerge at a rapid rate, so the association among features and the data distribution may shift, leading to the breakdown of models which were trained on older data. This issue implies that ensembles of adaptive NIDS, including ensembles, should be designed which can maintain good classification boundaries when the environment is changing [28].

The Role of Feature Engineering in Efficiency and Accuracy

The selection and design of features are critical to trade off between the accuracy of detection of attacks and the speed of the system, particularly when the network traffic is large. The most helpful attributes can be identified with the help of methods such as filter techniques (SelectKBest and Information Gain) and wrapper or embedded methods (Random Forest feature importance). This is achieved through picking out the most significant features, reducing the number of dimensions, which saves on costs of computing, increases the speed of training and may even help the model perform better in new data sets by avoiding the curse of dimensionality [3]. Overall, initial studies were devoted to this step. It relied on the feature importance of Random Forest and the Select Best approach to rank the variables so that the most essential traffic features like the statistics of the packet length and inter-arrival times which proved to be the most important ones are retained.

2.2 Review of Single-Classifier Machine Learning Approaches in NIDS

The hybrid ensemble model is created through the choice of base classifiers that create maximum diversity in algorithms, therefore, their errors do not intersect. Five classifiers will be selected, and these will include KNN, SVM, RF, GB, and LR. They all apply a varied method: KNN is distance-based, SVM is margin-based, RF is a tree ensemble variety which is variance-reducing, GB is a tree ensemble variety which is also a bias-reducing, and LR is a linear probabilistic [8].

2.2.1 Discriminative Models: K-Nearest Neighbors (KNN) and Support Vector Machines (SVM)

K-Nearest Neighbors (KNN) is a very basic, instance based learning algorithm which classifies a new sample by examining the majority class of its K nearest neighbors in the feature space. KNN is simple to configure and tends to perform well at the baseline, with an accuracy of 98.28 on the CICIDS 2017 dataset. Its biggest weakness is that it may be slow: the distance to classify time increases significantly with the size of the dataset since the algorithm needs to compute distance to all training points with each new prediction [17].

Support Vector Machines/SVMs are margin based classifiers which attempt to find the optimal hyperplane that characterizes the highest gap between varying classes. SVMs are best used in large-dimensional spaces and also can operate on non-linear boundaries with the use of a kernel function such as the Radial Basis Function (RBF). They can also withstand noise or overtraining and hence they can be applied in imbalanced datasets. Moderately, SVM in this research achieved 87.95% accuracy in the CICIDS 2017 dataset [9], which means that it was difficult to discover one hyperplane that was effective on all eight types of attacks.

2.2.2 Tree-Based Ensemble Methods: Random Forest (RF) and Gradient Boosting (GB)

Random Forest or RF is a set of trees which are constructed on the basis of bagging (Bootstrap Aggregating). All the trees are trained using a random sample of data and a random set of features. The randomness makes the variance of prediction less and tends to avoid the overfitting of single trees. RF has appeared to be a good generalizer, as it scored 97.00 percent on the CICIDS 2017 dataset [13], and it also produced consistent rankings in terms of feature importance in this experiment [13].

RF demonstrated strong generalization capabilities, achieving 97.00% accuracy on the CICIDS 2017 dataset. RF also played a methodological role in this study by providing reliable feature importance rankings.

Gradient Boosting or GB is a step-by-step sequential ensemble technique that enhances predictions. It includes additional weak learners typically, small decision trees, that concentrate on correcting the errors of the ones that came before it by fitting the negative gradient of the loss function. GB is more likely to be accurate by minimizing bias. GB attained 96.62% accuracy in this work similar to RF [13]. Its success in multi-class problems is because of its focus on misclassified samples.

2.2.3 Probabilistic Model: Logistic Regression (LR)

Logistic Regression (LR) is a linear regression model that approximates the probability of an item in a data set to be in a category by using the sigmoid regression to compute the estimates. Linear models are not always capable of capturing the non-linear dynamics of network traffic and LR can provide a valuable baseline. It had the lowest precision on the CICIDS 2017 dataset (85.59 percent) indicating that non-artificial network anomalies require more complicated models. However, LR remains part of the ensemble as a meta-classifier since it is simple to interpret, and remains constant when presented with the predictions of the first-layer models [18].

2.3 Advanced Ensemble Learning Techniques for Enhanced Robustness

Ensemble techniques are core to good anomaly detection and consist of using a number of models to get better results than the individual models.

Heterogeneous Ensembles: Stacking and Meta-Learning Architectures

A superior ensemble method is called stacking. It combines the predictions of various base models, and then places them in one or more layers and then the special meta model learns how to combine them in an optimal manner [14]. The concept of multiple stacking layers is based on its relative robustness on challenging classification problems, particularly when the data is unbalanced and has a large number of classes. It has been found that multi-layer stacking can outperform simpler models such as bagging or boosting and even a single deep neural network.

It is capable of obtaining F1-score equal to 0.99 and false-positive probability equal to 0.001 which is critical to identify small intrusions [20]. This high performance justifies the hypothesis that a combination of various algorithms in Layer 1 and a second model in Layer 2 to correct their errors is a highly robust detector.

2.4 Analysis of Benchmark Datasets in Network Security Research

The quality of NIDS models is determined by the quality of datasets that they are trained on. Validity and generalizability will be dependent on the quality and realism of those datasets.

CICIDS 2017 Dataset

The 2017 data of the CICIDS issued by the Canadian Institute of Cyber Security is a contemporary reference to NIDS [3]. It has five days of real network traffic in an environment with a complete topology (firewalls, routers, various operating systems) and user profiles, unlike old synthetic sets, e.g. KDD 1999. The number of its common attacks and features per instance is 11 and 85 respectively. Since it includes such attacks as DDoS, DoS Hulk, PortScan and Botnet, it can be trusted as a testbed of the existing and previous types of attacks [27].

NFS-2023-TE Dataset

The NFS-2023-TE was also used to compare the model with new threats. It concentrates on the security of IoT and edge-device. The number of attack types in a big IoT network is 33 and devices are some of the attackers and victims. Among them, the main ones are Spoofing (ARP, DNS), DoS (TCP/HTTP/SYN/UDP flood), Reconnaissance, and Mirai [5]. Experiments on this set indicate the generalizability of the models in case of a change in the feature space and attack styles in a drastic manner, simulating concept drift in the real world.

2.5 Identification of Research Gaps and Thesis Justification

In spite of the progress, the work on ensemble learning of NIDS has the following significant gaps:

- Gap 1: Only a small number of studies optimize multi-layer stacking of imbalanced and low-footprint attacks. Multi-layer stacking is little studied though there is single-layer stacking available which is conducted to enhance detection of infrequent attacks such as Bot traffic. The present environments perform dismally with these minority classes.
- Gap 2: There is a lack of studies into cross-domain generalization. Most are rated high on one set of data and are not concerned about the performances of the models on an entirely different graphical domain. We immediately require a study with a unified framework, such as stacking, in comparing the findings between an enterprise dataset (CICIDS 2017) and an IoT dataset (NFS-2023-TE), with a measure of actual adaptability and resilience.

The problems are addressed in the proposed hybrid multi-layer stacking model. The study will generate a high-performance, stable anomaly detector by combining diverse base learners (meeting Gap 1) with stringent testing on both datasets (meeting Gap 2) to improve the ability to better withstand complex threats and can remain robust as the networks change..

Chapter 3

Methodology

3.1 Architectural Overview of the Proposed Network Detection System (NDS)

The algorithm employs a stacking mechanism which integrates a number of classifiers in order to address the limitations of each. The first stage is data collection and cleaning followed by feature creation and lastly training and validation of the multiple models.

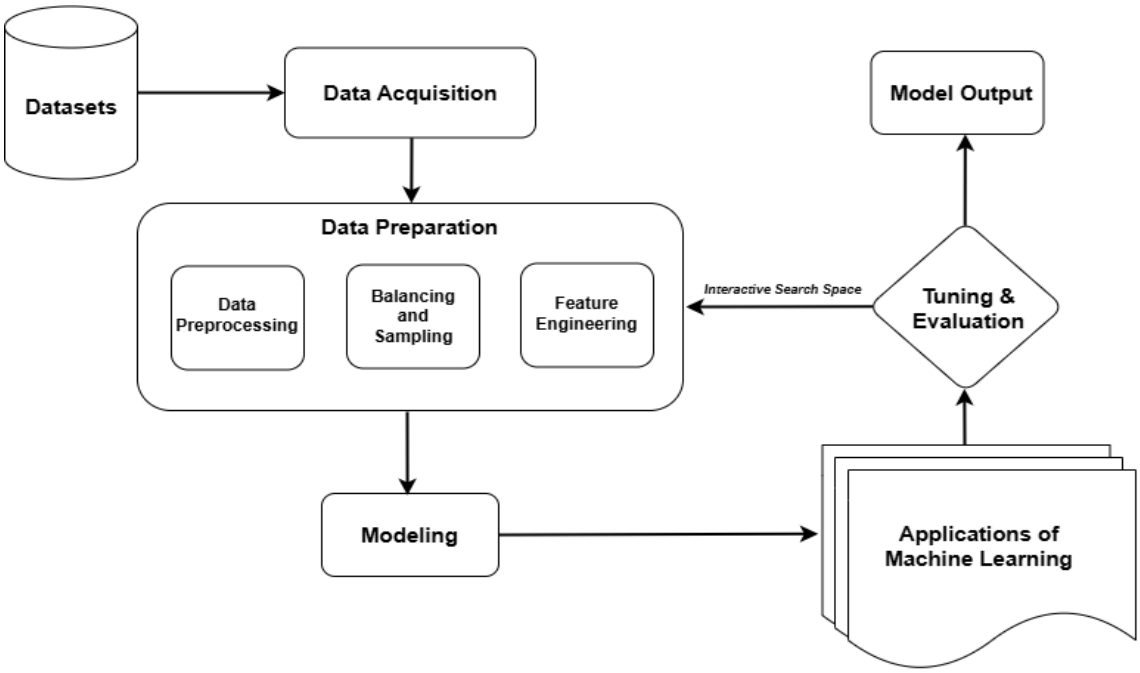


Figure 1: A data flow diagram representing the workflow of the proposed methodology.

There are two main layers of the stacking system.

- **Layer 1 (L1) – Base Models:** There are five machine-learning models, namely, Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Gradient Boosting (GB). These models are trained in the raw input data.
- **L2 (L2) Layer 2:** This is the final model which is typically the Logistic Regression trained on the predictions (or prediction probabilities) of the L1 models. These predictions are the new features which are better combined by the L2 model.

3.2 Data Acquisition and Preprocessing Methodology

Dataset Sourcing and Sampling

The two datasets were the CICIDS 2017 data (DS1) and the NFS-2023-TE data (DS2). The entire data set of CICIDS 2017 is massive (almost 2.8 million rows), so we selected a random sample of 60,000 of them. This holds sufficient examples of every type of attack as well as training is rapid particularly with slower models such as SVM and KNN.

Data Splitting

The sample was divided with the normal train_test_split utility in order to do the training and testing fairly. We retained 70 percent training, 20 percent testing and 10 percent validation. The training set does the training of the L1 models and generates cross-validated predictions of the L2 model. The test set supplies the ultimate accuracy figures. Parameters tuning or early termination The validation set is used to early terminate training or to tune parameters.

Feature Engineering, Selection, and Dimensionality Reduction

The data were rich in terms of features, i.e. CICIDS 2017 consisted of 85 variables. In order to ensure that the models are fast and prevent overfitting, we selected the most significant features. Our initial filter technique was SelectKBest and afterward we used an embedded technique (Random Forest importance). The figure below (2) illustrates the importance scores of SelectKBest DS1. Attributes such as the average backward packet size and the average backward segment size were interesting. They are good in displaying attacks which send large amounts of traffic including DDoS or DoS Hulk. Additional important characteristics included max backward packet size, standard deviation of forward inter-arrival time and standard deviation of packet length. These demonstrate the appearance of abnormal traffic in case the packets are too large or between too distant.

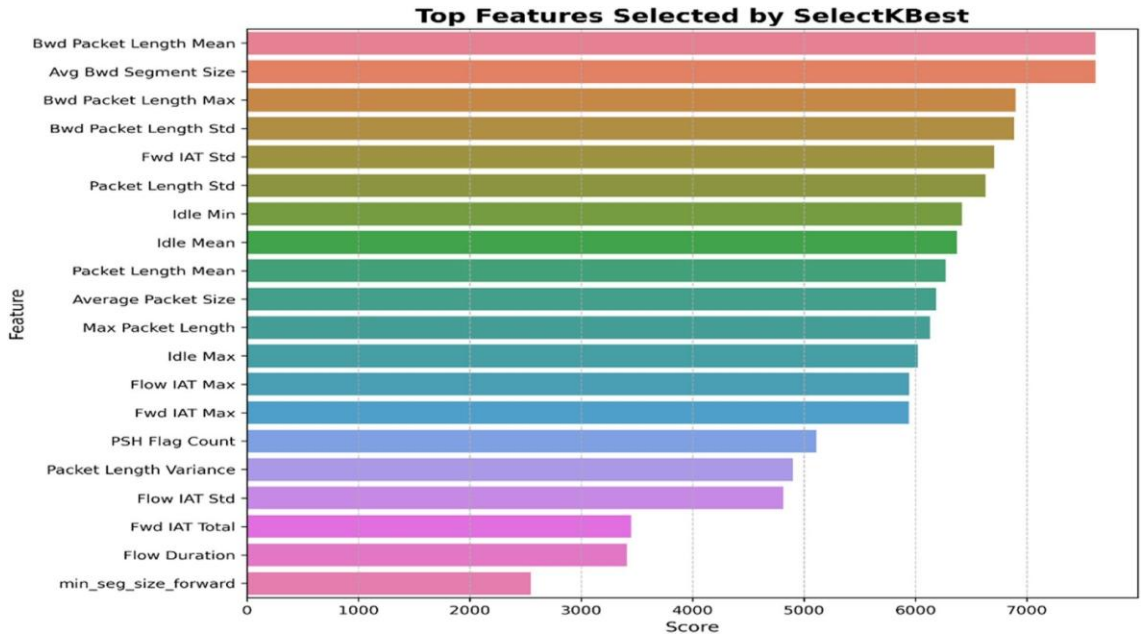


Figure 2: Evaluation of significant features extracted from the CICIDS2017 dataset.

Figure 3 repeats the same with DS2 that has IoT and edge traffic. In this case, the best aspects are flow-direction and state metrics, i.e., mean packet size between destination and source, mean packet size between source and destination, and reset packets between destination and source. It is demonstrated that this is because IoT attacks such as spoofing, Mirai, special DoS attacks can be more effectively identified based on how packets are transported back and forth as well as control flags such as RST, FIN, SYN, instead of only traffic volume.

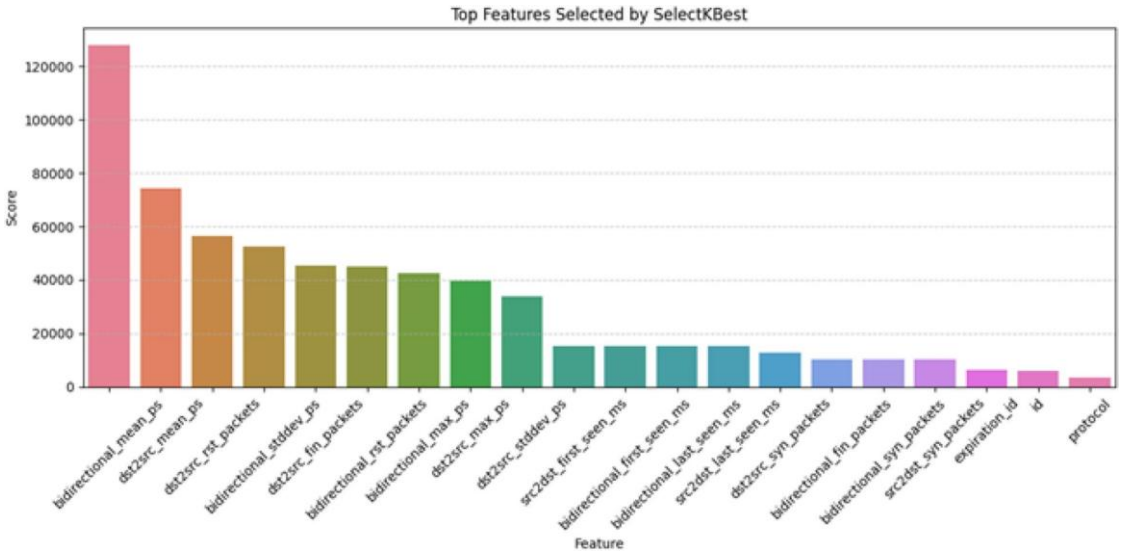


Figure 3: Significant features identified from the NFS-2023-TE dataset.

3.3 Mathematical Formulation of Base Classification Algorithms (Layer 1)

The five base classifiers were selected to ensure diversity across distinct machine learning paradigms.

K-Nearest Neighbors (KNN):

KNN operates on the assumption that similar data points exist in close proximity. Classification is determined by a weighted plurality vote among the *K* nearest neighbors, calculated using a distance metric, most commonly the Euclidean distance.

The Euclidean distance between two *F*-dimensional feature vectors *Xi* and *Xj* defined as:

$$\mathbf{d}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{F} (x_{i,k} - x_{j,k})^2} \quad (1)$$

Where *xi*, *k* and *xj*, *k* represents the *k*-th feature value for the respective vectors.

The classification rule assigns a probability $P(y = c|\mathbf{x})$ to class *c* based on the inverse distance weighting scheme, where closer neighbors contribute more significantly to the final decision [10]:

$$P(y = c|\mathbf{x}) = \frac{\sum_{i=1}^{K} I(y_i = c) \cdot w_i}{\sum_{i=1}^{K} w_i}, \quad \text{where } w_i = \frac{1}{\mathbf{d}(\mathbf{x}, \mathbf{x}_i} \quad (2)$$

..ere,

I(.) is the indicator function, *yi* is the known class label of the *i*-th neighbor, and *p* is a power (often p=1 or p=2) determining the strength of the inverse distance weighting.

Support Vector Machines (SVM):

SVM seeks to discover an ideal differentiating hyper plane in feature space. For irregularly recoverable networks, the kernel approach is needed to translate the input data into higher-dimensional space where linear separation becomes viable [23].

$C \sum_{i=1}^{n} \xi_i$ The primal optimization problem for soft margin classification minimizes a penalty term alongside the weight vector norm, subject to classification constraints:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad (3)$$

Subject to constraints:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \quad \text{for } i = 1, \ldots, n \quad (4)$$

Here, *w* is the normal vector to the hyperplane, *b* is the bias ξ_i are the non-negative slack variables allowing for misclassification, and *C* is the penalty parameter controlling the trade-off between margin size and misclassification error.

The most effective approach for complex network traffic data is often the Radial Basis Function (RBF) kernel, which computes the equivalent dot product in the transformed space [11]:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) \quad (5)$$

Where *y* gamma is a scaling parameter.

Random Forest (RF):

Random Forest constructs *T* decorrelated decision trees, relying on the principle of bagging (Bootstrap Aggregating) to reduce variance. Each tree is grown by recursively partitioning the data space based on feature splits that maximize class purity, commonly measured using the Gini impurity index.

The Gini impurity $Gini(\mathbf{D})$ for a dataset *D* with *C* classes is calculated as:

$$Gini(\mathbf{D}) = 1 - \sum_{j=1}^{C} p_j^2 \quad (6)$$

Where *pj* is the probability of an instance belonging to class *j* in the dataset D [12]. A split is selected to maximize the information gain, defined as the reduction in Gini impurity after the split.

The final classification prediction $\hat{Y}_{RF}(\mathbf{x})$ is determined by the majority vote (mode) among the predictions of the *T* individual trees:

$$\hat{Y}_{RF}(\mathbf{x}) = \text{Mode}\{T_t(\mathbf{x})\}_{t=1}^{T} \quad (7)$$

Gradient Boosting (GB):

Gradient Boosting builds its model sequentially, aiming to minimize an arbitrary differentiable loss function *L* by fitting each new weak learner to the negative gradient of the loss function, termed the pseudo-residuals [13]. The additive model construction at step m is given by:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma_m h_m(\mathbf{x}) \quad (8)$$

Where $F_{m-1}(\mathbf{x})$ is the prediction from the previous stage, *hm(x)* is the new weak learner, and *ym* is the step size (shrinkage) [13].

The weak learner *hm(x)* is trained to approximate the negative gradient (pseudo-residual) for each observation *i*:

$$r_{i,m} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_{m-1}} \quad (9)$$

The optimal step size *ym* for the new tree is found by minimizing the loss over the current approximation F_{m-1}

$$\rho_m \approx \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h_m(\mathbf{x}_i)) \quad (10)$$

Logistic Regression (LR):

Logistic Regression serves as a baseline model that transforms the linear combination of input features into a probability estimate via the sigmoid function σ(z) for binary classification. The sigmoid function is defined as:

$$\sigma(z) = P(Y = 1|X) = \frac{1}{1 + e^{-z}} \quad (11)$$

Where z = w. x + b is the linear combination of input features x and weights w.

For the multi-class anomaly detection task involving eight distinct classes, the generalization of the sigmoid function, the Softmax function, is used to assign probabilities to each class *j*:

$$P(Y = j|\mathbf{x}) = \frac{e^{\mathbf{w}_j \cdot \mathbf{x} + b_j}}{\sum_{k=1}^{C} e^{\mathbf{w}_k \cdot \mathbf{x} + b_k}} \quad (12)$$

This probability output is essential for the soft voting and stacking approaches. Training involves maximizing the likelihood of the observed training data, which is equivalent to minimizing the cross-entropy loss [22].

3.4 Detailed Stacked Generalization and Ensemble Design

3.4.1 Principle of Hard and Soft Voting Ensemble

Before implementing stacking, two simpler aggregation methods were evaluated:

- **Hard Voting:** This classifier aggregates the discrete class predictions of the base models. The final output is determined by the majority class label.

$$\hat{Y}_{V,\text{hard}} = \text{Mode}\{\hat{Y}_k\}_{k=1}^{K} \quad (13)$$

- **Soft Voting:** This method is preferred when base classifiers provide probabilistic outputs. It aggregates the predicted probabilities for each class, often using weights derived from validation performance, and selects the class with the highest combined probability.

$$\hat{Y}_{V,\text{soft}} = \arg\max_{c} \sum_{k=1}^{K} w_k P_k(Y = c|\mathbf{x}) \quad (14)$$

The superior performance of soft voting (98.30% accuracy) compared to hard voting (96.97% accuracy) confirms the benefit of weighting predictions by confidence scores in complex, multi-class environments.

3.4.2 Multi-Layered Stacking Architecture Rationale (Hybrid Model)

The proposed hybrid model employs stacked generalization to optimally combine the strengths of the five heterogeneous base classifiers (L1). The goal is to leverage the diversity of the base models—whose distinct mechanisms lead to complementary error patterns—to achieve better generalization [8].

- **Layer 1 (L1) Training:** The five classifiers (KNN, GB, RF, SVM, LR) are independently trained on the full training dataset D_*train.*
- **Meta-Feature Generation:** To prevent data leakage, the predictions used to train the meta-classifier are generated using k-fold cross-validation on D_*train*. For each fold, k-1 subsets are used to train the L1 model, and the k-th subset is used to generate an out-of-sample prediction. These predictions are concatenated to form the new meta-feature matrix, Z. Z has K columns (one for each base classifier) and *N_train* rows [26].
- **Layer 2 (L2) Training:** A meta-classifier (Logistic Regression is commonly employed for its stability and ease of training on the meta-features) is trained on the meta-feature matrix *Z* and the true labels of the training data. This process teaches the meta-model how to optimally weight or combine the initial predictions, thus minimizing the overall prediction error.

The final stacking prediction $\hat{Y}_{\text{Stack}}(\mathbf{x})$ is given by the output of the MetaModel applied to the L1 predictions for a new instance x:

$$\hat{Y}_{\text{Stack}}(\mathbf{x}) = \text{MetaModel}(\{\hat{P}_k(\mathbf{x})\}_{k=1}^{K}) \quad (15)$$

Where $\hat{P}_k(\mathbf{x})$ is the output prediction (or probability vector) from the k-th base classifier trained on the entire *D_train* set.

3.5 Performance Evaluation Metrics

The evaluation of the NDS model's performance relies on standard metrics derived from the confusion matrix, which records the counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for each class.

- **Accuracy**: Represents the overall proportion of correct classifications (both benign and malicious traffic).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (16)$$

- **Precision (Positive Predictive Value):** Measures the relevance of the detected anomalies. High precision is crucial in NIDS to minimize False Positives (FP), which cause alert fatigue and diminish operational efficiency.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

- **Recall (True Positive Rate/Sensitivity):** Measures the ability of the model to identify all actual anomalous instances. High recall is critical for minimizing False Negatives (FN), ensuring that actual threats are not missed, which is essential for system resilience.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

- **F1-Score:** The harmonic mean of precision and recall. This metric is particularly important for evaluating performance on highly imbalanced datasets or for specific, rare attack categories, as it penalizes models that exhibit poor performance in either precision or recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

Chapter 4
Results & Discussion

4.1 Experimental Environment and Implementation Details

The code was coded in Python, with Scikit-learn as the base models, the voting system and the stacking classifier. The 60,000-row samples of DS1 and the corresponding slice of DS2 allowed training of the many multifaceted models, among them the kernel based SVM. All the models were trained using the 70 percent training data and final scores measured using the 20 percent test data.

4.2 Baseline Performance of Individual Base Classifiers

The two data sets produced different results on the base models hence the fact that they are not consistent in their responses to changes in features. Table1 presents the success of the models on DS1 and DS2. KNN scored 98.28 % on DS1 and 91 % on DS2. Random Forest was 97.00 per cent on DS1 and 96 per cent on DS2. On DS1, Gradient Boosting received 96.63 per cent and on DS2, 92. These models were good in that they are able to identify complicated patterns. The best precision that a generic machine-learning model could achieve was approximately 87.95 1/2, probably due to the inability to differentiate between similar categories. Logistic Regression was the most weak with a precision of approximately 85.59.

Table 1: Evaluation and comparison of accuracy performance across base models.

Model	Accuracy (CICIDS2017)	Accuracy (NFS-2023-TE)
Random Forest	0.97	0.96
Support Vector Machine	0.88	0.90
K-Nearest Neighbors	0.98	0.91
Gradient Boosting	0.97	0.92
Logistic Regression	0.86	0.93

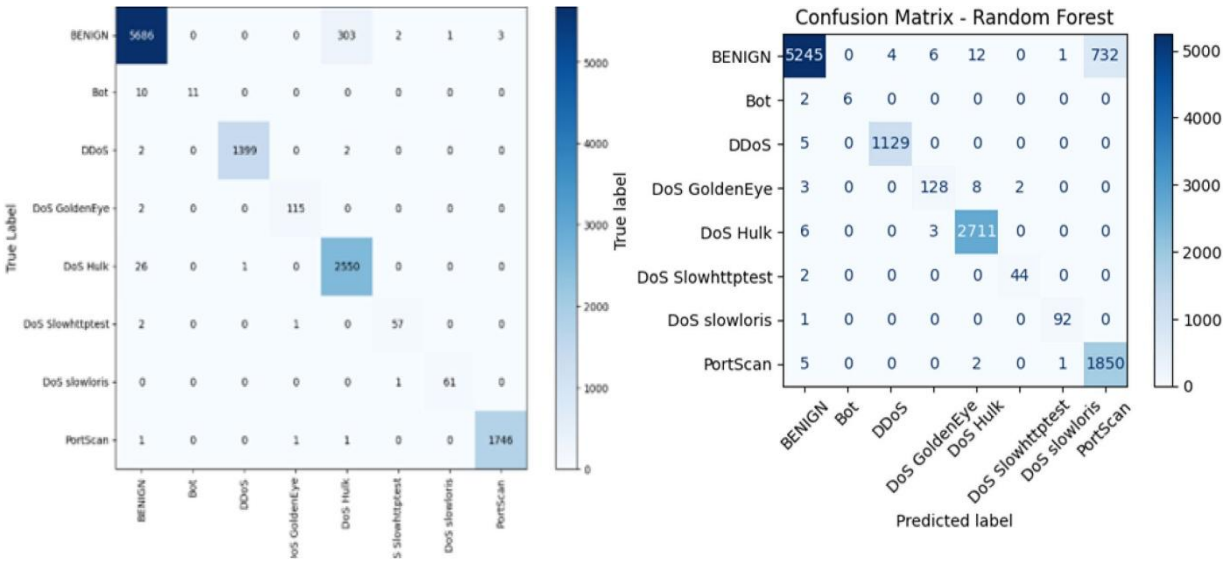


Figure 11: Random Forest confusion matrix results for CICIDS2017 and NFS-2023-TE datasets.

The random forest Confusion matrix on DS1 indicates that it can be used in the majority class. It rightly classified 5686 benign events but wrongly classified 303 benign events as PortScan. This demonstrates that it is able to miss high-volume normal traffic as a scan. In the case of the DoS Hulk attack, it was able to identify 2,550 cases and very minimal number were falsely classified as benign. It was also accurate in identifying 1399 DDoS cases. It was however more struggling with minority classes such as Bot and DoS Slowhttptest which it misclassified more frequently. The reason is that such attacks are not as frequent and not easily learnt by tree-based models.

The RF performance is altered using the NFS-2023-TE dataset (DS2). In this case the number of BENIGN instances correctly classified is 5,245 whereas false positives are misclassified as other attacks (732). PortScan attack had 1,850 True Positives, and slight misclassifications. With 1,129 True Positives, the model has strong resistance to the high volume DDoS attack. This finding indicates that although the RF model tends to be highly effective, the misclassification trend varies, and it is in line with the fact that the discriminating factors that worked well in DS1 cannot be convincingly applied to the IoT scope of DS2.

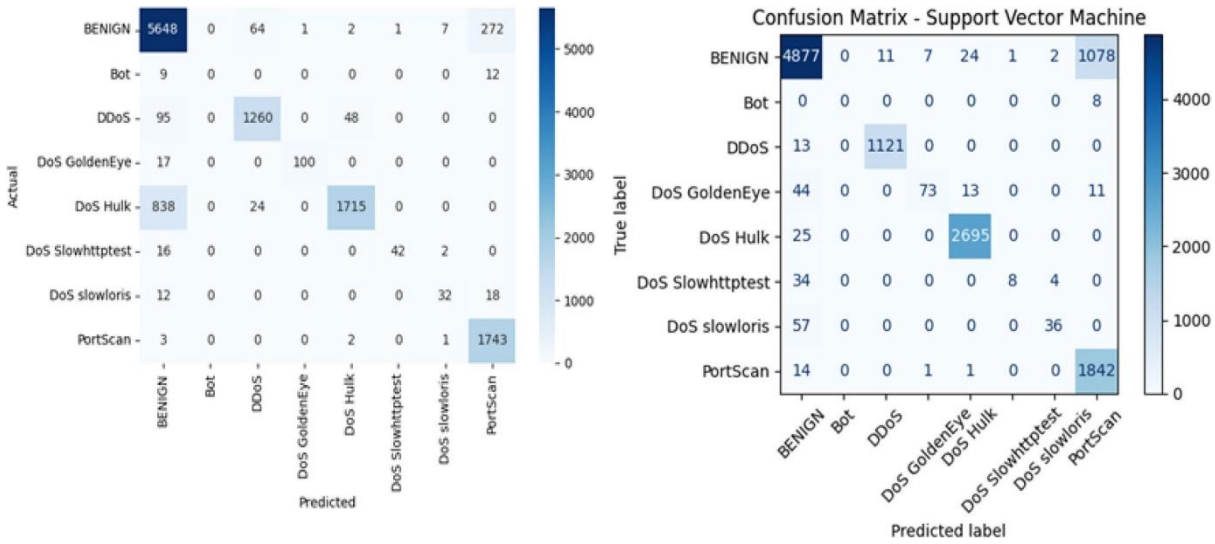


Figure 12: Support Vector Classifier confusion matrix results for CICIDS2017 and NFS-2023-TE datasets.

The Support Vector Classifier (SVC) proved to be very difficult, especially on the DoS Hulk category which is on DS1. Although it has made 5,648 accurate predictions when dealing with BENIGN traffic and 1,743 making correct predictions when dealing with portscan (True Positives), the model has made disastrous mistakes by misclassifying 838 DoS Hulk as BENIGN (False Negatives). This volumetric attack FNR value indicates that the separation boundaries created by the SVM linear or kernel model had a hard time theorizing a neat division of the feature space between this attack and regular traffic. Moreover, the DDoS type has 1,260 correct predictions, yet 95 samples were classified as BENIGN and 48 belonging to other types of attacks, which proves the instability of the model that cannot generalize its high bias on the complex separation of multi-classes.

The SVC got 4877 True Negatives with BENIGN traffic on the DS2 dataset, although it incorrectly identified 1078 with some type of attack (False Positives).3 DDoS category was getting 1,121 True Positives. The PortScan category achieved 1,842 True Positives.3 The general trend supports the moderate performance of the SVC (87.95% on DS1) as well as its sensitivity to feature representation, which is the key reason behind its inclusion in a diverse ensemble where its shortcomings may be mitigated by the rest of the complementary classifiers.

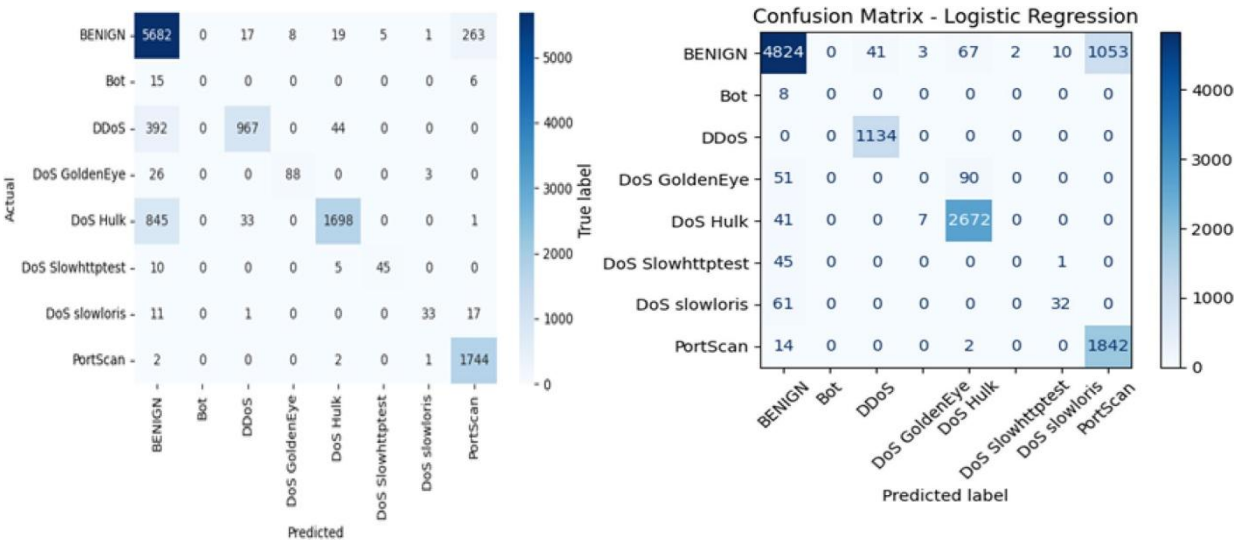


Figure 13: Logistic Regression confusion matrix results for CICIDS2017 and NFS-2023-TE datasets.

The lowest baseline accuracy (85.59) on DS1 was obtained using the Logistic Regression (LR) model. This has been established through the confusion matrix that includes poor performance in the classification, especially with the high volume attacks. The number of DDoS instances that were actually identified (True Positives) was only 967, which is compared to the high number of the False Negative (FNR). Most importantly, the LR model falsely labeled 845 instances of DoS Hulk as BENIGN, the largest single misclassification rate of any base model, and is highly indicative of its core inability to capture the non-linear decision boundaries needed to detect network anomalies.

Surprisingly, the LR model was accurate on DS2 (93.00%). It is depicted in the confusion matrix that there are 4,824 correct predictions of BENIGN, and 1,053 incorrect predictions (False Positives). The DDoS attack was ideally categorized and had a True Positive of 1,134. The outstanding difference in performance between the two datasets, even though FNR in the case of BENIGN traffic is significant, is a manifestation of how the bias of the linear model can coincidentally align with feature distribution of the IoT data (DS2) than the conventional enterprise data (DS1), but its overall weakness is a limitation to its use in operations.

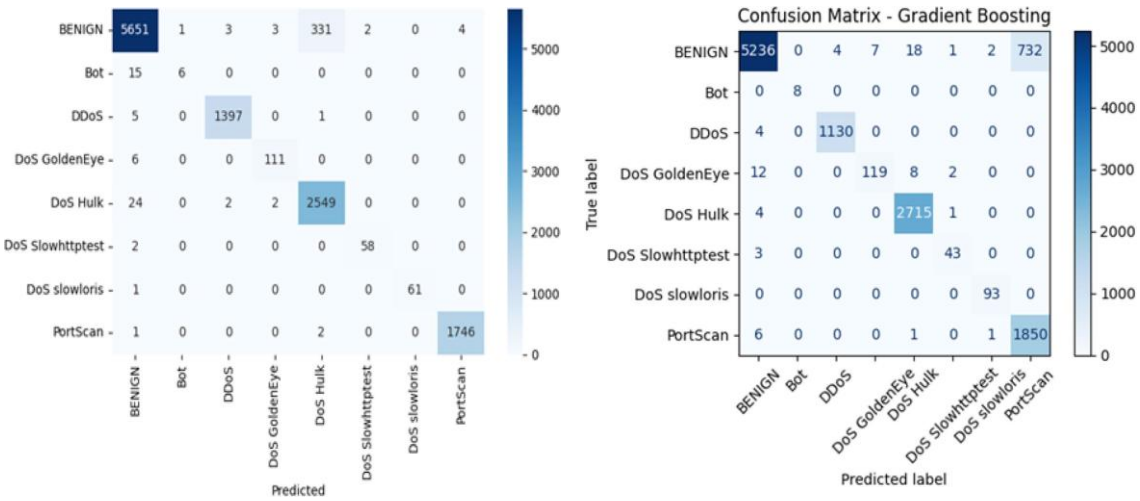


Figure 14: Gradient Boosting confusion matrix results for CICIDS2017 and NFS-2023-TE datasets.

GB model was able to achieve high accuracy (96.62) on DS1. BENIGN and PortScan had high True Positive/Negative rates with 5,651 and 1,746 true positives and negatives, respectively. The accuracy was also high with DDoS (1,397 True Positives), DoS Hulk (2,549 True Positives).331 BENIGN samples were however misclassified as PortScan (False Positives). Moreover, the model had a big issue with the minority Bot group with the total number of instances correctly identified being 6/21 (True Positives), a high FNR of this type of stealthy attack. This is the inherent weakness of boosting techniques as they are used on severely imbalanced data unless special weighting of the minorities-class is explicitly used. GB performed well on the DS2 dataset, with the highest amount of correct BENIN predictions of 5236, and the most common misclassified other attacks of 732. DDoS (1,130 True Positives) and DoS Hulk (2,715 True Positives) volumetric attacks were strongly identified. This trend validates the high generalization ability of tree based ensembles, which are inherently resistant to non-linear feature representations on new-data.

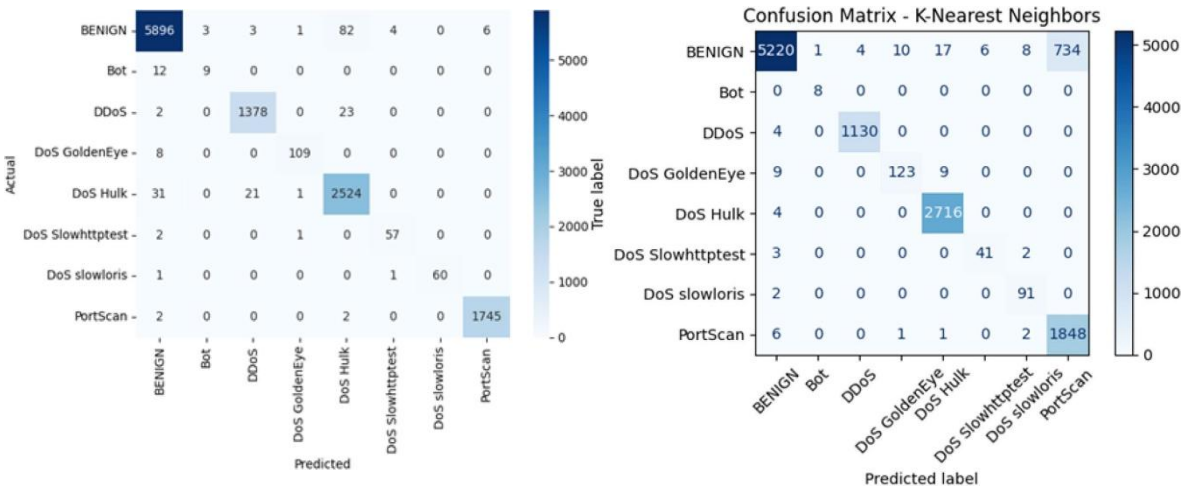


Figure 15: K-Nearest Neighbors confusion matrix results for CICIDS2017 and NFS-2023-TE datasets

K-Nearest Neighbors (KNN) model on DS1 had the greatest baseline accuracy (98.28%). The confusion matrix proves the outstanding results of the BENIN class where 5,896 instances are correctly recognized (True Negatives) and only a small number of misclassifications (82 BENIN samples wrongly classified as DoS Hulk). The detection in DDoS was strong (1,378 True Positives). Like GB, the biggest weakness is the identification of Bot category whose 9 cases were identified correctly, and 12 cases were misclassified. Although this model has a high total accuracy, the minority class has a severe FNR, demonstrating a high security gap. The performance of KNN significantly dropped on the DS2 dataset (91.00% accuracy). Although the BENIGN traffic did not decrease to 5,220 True Negatives, the misclassification rates rose.3 DDoS (1,130 True Positives) and PortScan (1,848 True Positives) still had high detection rates. The decreased performance justifies the sensitivity of the model to alterations in high-dimensional data (concept drift), since a distance-based measure is ill-posed to radically different feature space of the IoT setting.

4.3 Comparative Performance of Voting Ensembles

4.3.1 Hard Voting Ensemble Analysis

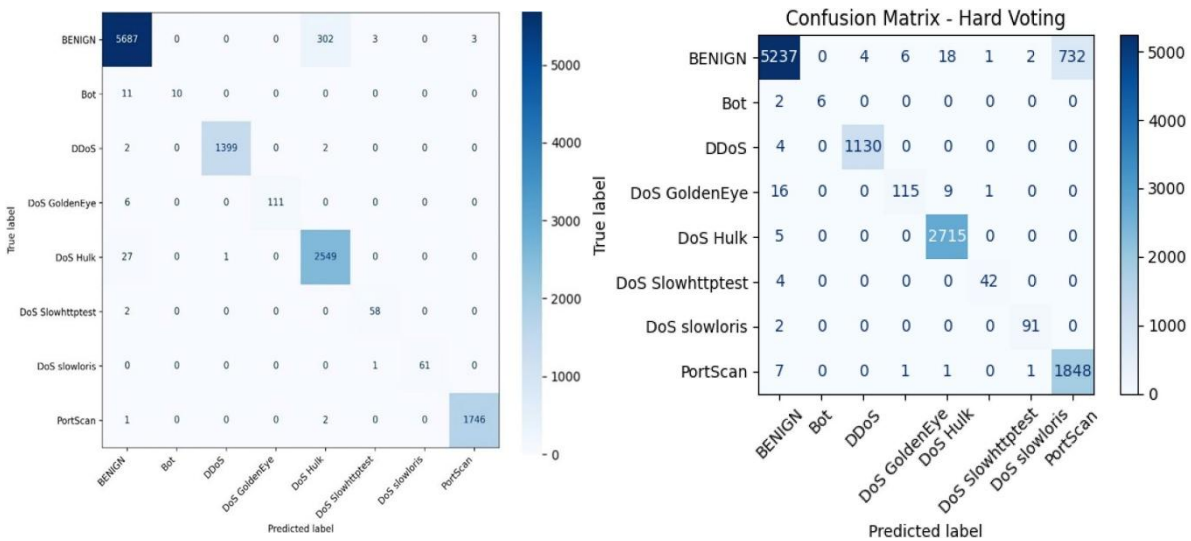
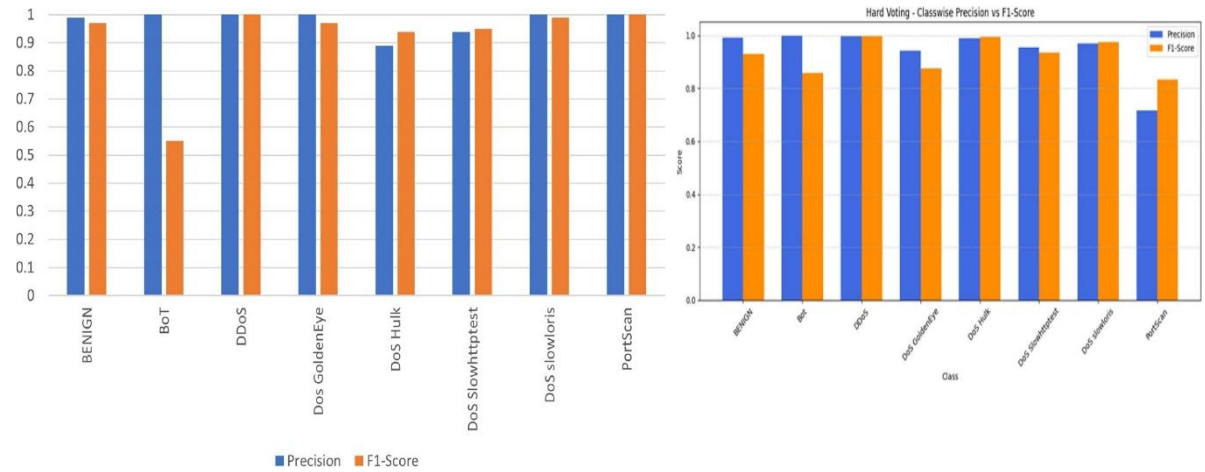


Figure 16: Hard voting ensemble confusion matrix results for CICIDS2017 and NFS-2023-TE datasets.

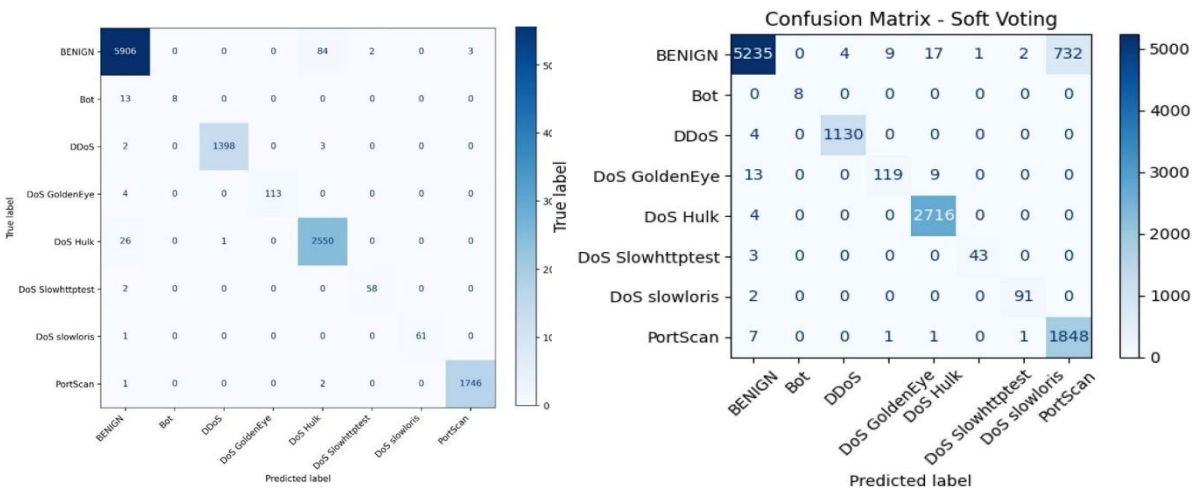
Hard Voting classifier was shown to be more effective in aggregation, with the highest quality of classification over the weakest base models, 5687 BENIGN and 2549 DoS Hulk. Although strong, 302 BENIG cases were falsey classified as DDoS with high count of False Positive of this attack as compared to the Soft Voting technique. PortScan has scored high with 1,746 correct predictions.



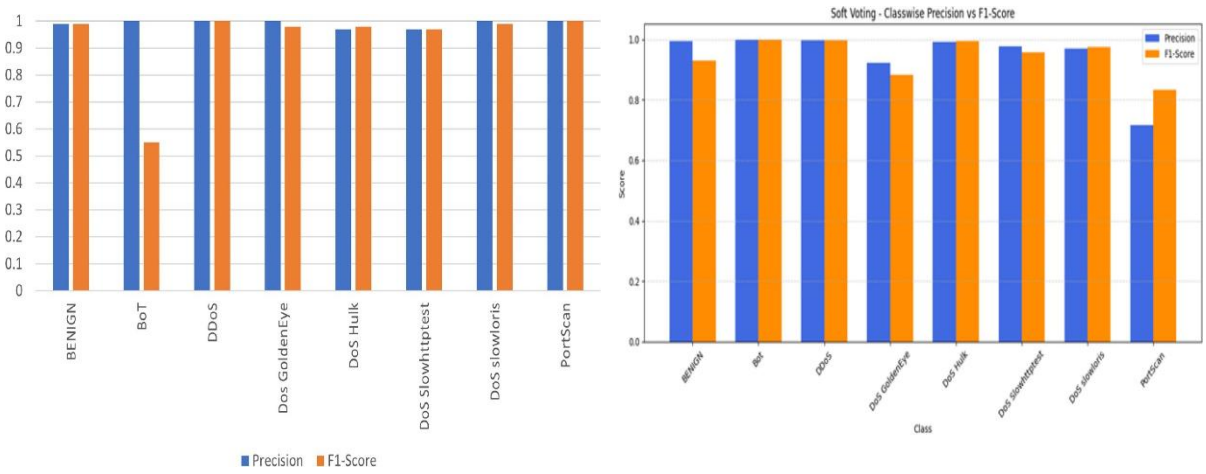
The metrics by the class support the strengths of the model, with both DDoS and PortScan having a perfect F1-score of 1.00. They also scored high in DoS Hulk (F1: 0.94), and BENIGN (F1: 0.97). Nevertheless, the Bot attack has an important weakness in the form of a low recall of 0.48 with a corresponding F1-score of 0.65. Although there is a perfect accuracy of (1.00), the model is only identifying slightly less than half the real Bot cases. Figure 10 show the respective confusion matrix and F1/Precision charts of the NFS-2023-TE dataset, the generalization performance and reveal high F1 and low precision respectively in high-volume attacks such as DDoS but low recall (0.72) in PortScan.

4.3.2 Soft Voting Ensemble Analysis

The overall performance of the Soft Voting Classifier (accuracy 98.30) was better than the Hard Voting Classifier (accuracy 96.97) on DS1, which confirms the advantage of combining probability-weighted decisions.



Soft Voting matrix shows fewer misclassification errors. It correctly recognized 5,906 BENIGN class incidences. Importantly, there were about 84 BENIGN cases that were incorrectly considered as DDoS (False Positives) and this is a significant reduction as compared to the 302 that were incorrectly considered by Hard Voting technique. This demonstrates that the probability-weighted method is better at distinguishing massive benign traffic and DDoS. Detecting DoS Hulk was also very accurate (2,550 True Positives).



The metrics of its class-wise showing a positive improvement of F1-scores of 0.98 of DoS GoldenEye and 0.98 of DoS Hulk, slightly above Hard Voting model, serves as an intrinsic limitation of the ensemble that the Bot attack category realized low recall (0.38) and F1-score (only 0.55) of that specific, subtle type of attack in the input of the base layer. The results of the NFS-2023-TE data are presented in Fig. 12 and show that the Soft Voting model is more effective in working with ambiguous samples, and, accordingly, the high level of performance is maintained throughout volumetric attacks.

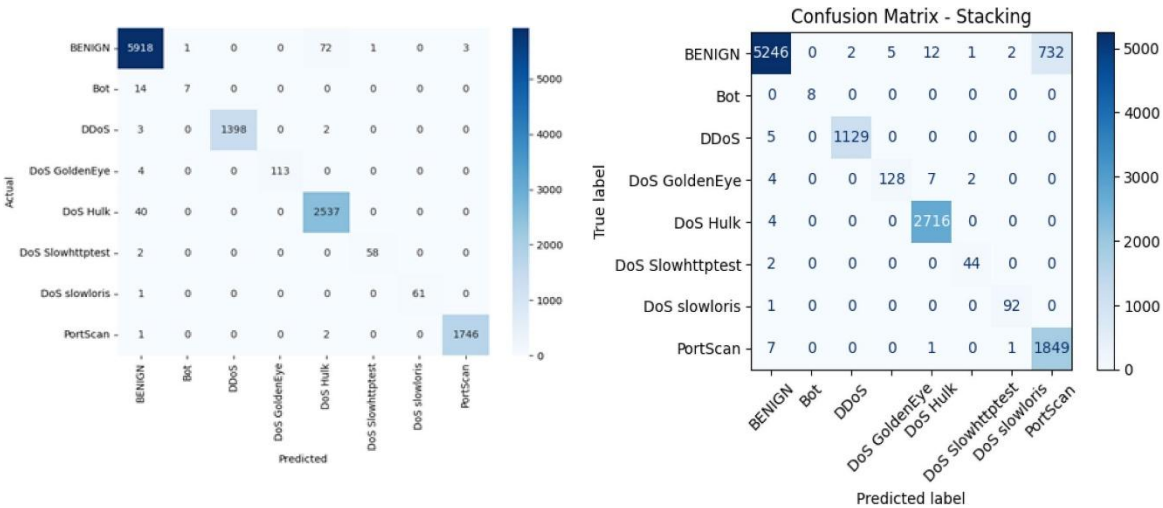
4.4 Empirical Results of the Hybrid Stacking Model

The highest overall predictive accuracy was obtained with the Hybrid Multi-Layered Stacking Model which supports the essence of the research. On the dataset of CICIDS 2017, the stacking model achieved a total accuracy of 98.79%. This is an actual increase in performance over the highest individual classifier (KNN at 98.28%) and the highest performing voting ensemble (Soft Voting at 98.30%).

Table 2: Classification Performance Report for Stacking Model

Attacks	Precision (DS1)	Recall (DS1)	F1-score (DS1)	Support (DS1)	Precision (DS2)	Recall (DS2)	F1-score (DS2)	Support (DS2)
BENIGN	0.99	1	0.99	0.87	0.99	0.93	5995	6000
Bot	1.00	1	0.38	1	0.55	1	21	8
DDOS	1.00	1	1.00	1	1.00	1.00	1403	1134
DoS Goldeneyes	1.00	0.96	0.97	0.91	0.98	0.93	117	141
DoS Hulk	0.97	0.99	0.98	1	0.98	1.00	2577	2720
DoS Slowhttptest	0.97	0.94	0.97	0.96	0.97	0.95	60	46
DoS slowloris	1.00	0.97	0.98	0.99	0.99	0.98	62	93
PortScan	1.00	0.72	1.00	1	1.00	0.83	1749	1858

The confusion matrix of the final Stacking model on DS1 shows clearly the best aggregation the meta-classifier has done. It had 5,918 correct predictions on BENIGN traffic (True Negatives)3 This is a tremendous decrease in False Positives than the base models. Weakly classified DoS Hulk (2,537 True Positives) and PortScan (1,746 True Positives) were strongly classified. The matrix indicated a few unlocation between the "Class 3 and Class 4" which is the slight difference between two types of DoS (DoS GoldenEye and DoS Hulk) showing the capability of even stacked model in solving highly ambiguous attack traffic which happens to be statistically similar.



The metrics on the different classes support the strengths of the model and its fundamental weakness. The model had a perfect detection (F1-score 1.00) of DDoS and PortScan. Practical perfection was almost BENIG classification (F1: 0.99). Nevertheless, the Bot attack type had a very low recall of 0.38 and an F1-score of 0.55 and precision (1.00). This finding conclusively shows the structural difficulty of minority, low-footprint attacks which cannot be extracted by standard feature extraction models, but rather specialized feature extraction can be done.

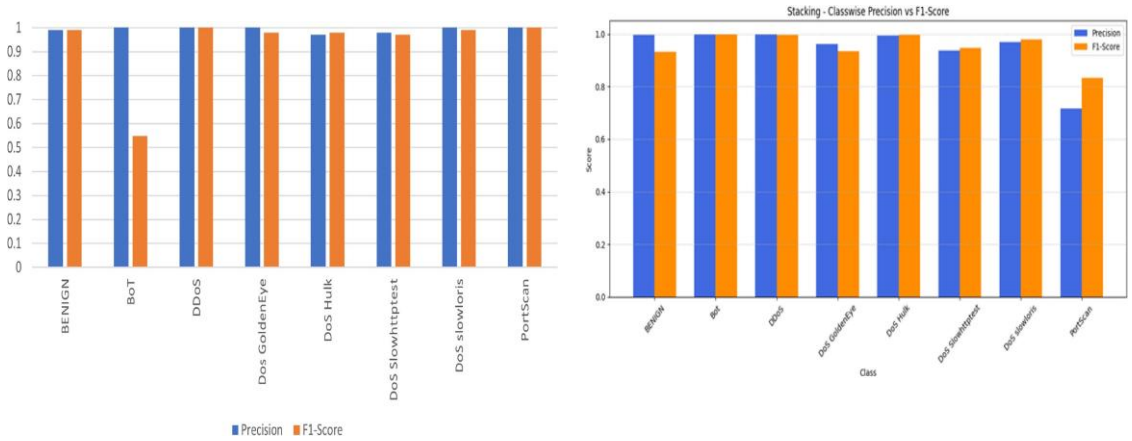


Figure 14, shows the confusion matrix and F1/Precision curves of the NFS-2023-TE data, which show that on data with high volumes of threats, it still performs well, but again on this architecture, the stacking component performs better in terms of resistance to domain shift (concept drift).

4.5 Summary of Overall Detection Accuracy

The comparative study of all approaches proves that the process of integrating classifiers with the help of stacking represents the best synthesis of varying predictive powers, which leads to the most solid overall performance.

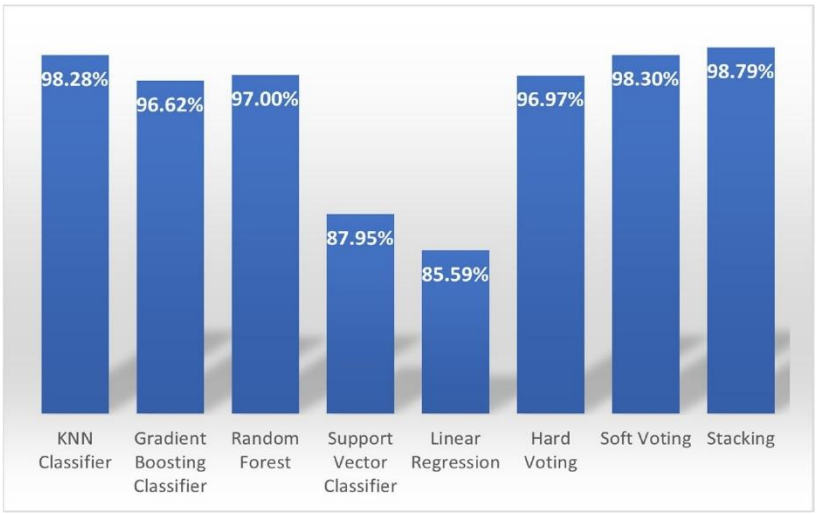


Figure 17: Evaluating Accuracy of All Techniques on the CICIDS2017 Dataset

This graph summarizes the results of the experiment on the DS1 benchmark graphically. It affirm that the individual base classifiers (KNN: 98.28% and RF: 97.00 and GB: 96.62 and SVM: 87.95 and LR: 85.59) create a broad performance range. This baseline is obviously outperformed by the ensemble methods, as Soft Voting (98.30%), Hard Voting (96.97%), are more successful than most individual models. The Stacking model has the highest performance of 98.79, which quantitatively demonstrates the worth of the stacked generalization strategy in maximizing stability in prediction and the overall classification performance in a static setting.

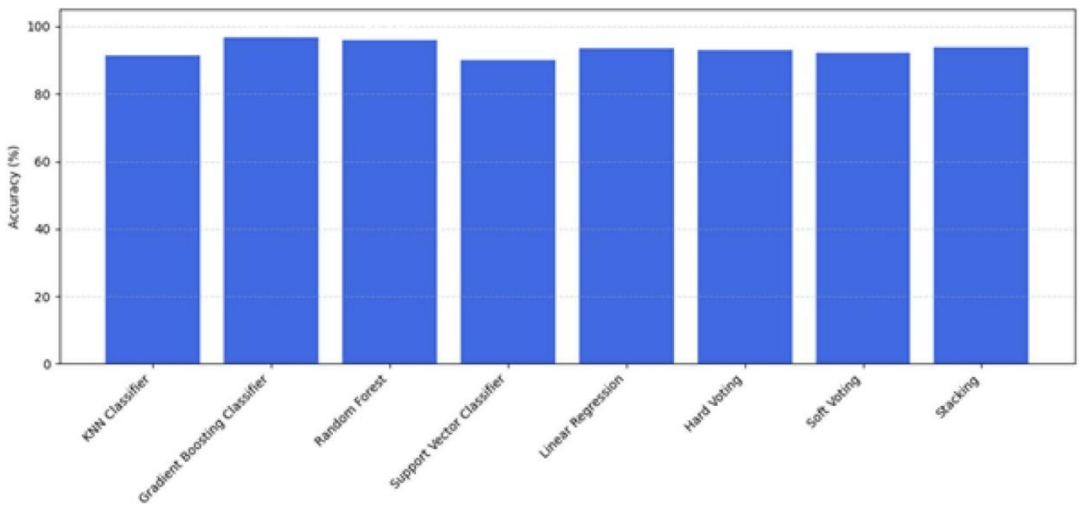


Figure 18: Evaluating Accuracy Across Different Techniques on the NFS-2023-TE Dataset

The NFS-2023-TE dataset (DS2) comparison is a crucial metric of generalization and resistance to domain shift, and although the absolute accuracy of individual models will change (e.g. LR to 93.00% and KNN to 91.00), the relative performance of the ensemble techniques does not change considerably, and Stacking has a score of one of the highest accuracies. This chart is a graphical empirical demonstration on the concept drift phenomenon and supports the assumption that the ensemble structure does offer some level of inherent stability that is invaluable in making the system resilient enough to its domain of operation in heterogeneous network settings [2]. The empirical findings evidently prove the better performance of Hybrid Multi-Layered Stacking Model, with the maximum accuracy of 98.79 percent on the CICIDS 2017 dataset, which is a better performance than the single best learner (KNN with the maximum accuracy of 98.28 percent). This enhancement is not just statistical noise, but it is the effective implementation of the principle of error decorrelation provided by heterogeneous ensemble learning.

Single models like SVM and logistic regression will frequently cause enormous errors when data indicates that the attack traffic resembles normal traffic (like a DoS Hulk attack can be detected as BENIGN). Trees based models such as Random Forest and Gradient Boosting are highly sensitive to minute variations in the data yet effective at detecting complicated decision boundaries. The stacking technique is done by using the predictions of the base models as new features of input. These prediction patterns are then learned by a meta-classifier to map the true label and this decreases both bias and variance simultaneously. This will demonstrate why it is not sufficient to average the outputs of the various models (hard or soft voting) since this will not be able to learn the optimal weights of the strengths and weaknesses of each model.

Chapter 5

Conclusion

This paper has managed to construct and evaluate an Adaptive and Resilient Networking Anomaly Recognition Hybrid Multiple-layer Stacking Ensemble Architecture based on Multiple Layer Stacking. The findings are in line with the hypothesis that a significant improvement in the accuracy of predictions of complex, multi-class security issues is achieved when various types of models are combined. The stacking model was the most accurate classifier using the CICIDS 2017 data: 98.79, which compares to 98.28 of the best KNN model and easier voting ensembles. The derivation of multiple base algorithms is also explicated in the paper, and how the various error signals of cross-validation and meta-learning combine with each other. We have also used the comparison of the model with the NFS-2023-TE dataset and identified the natural gap in the performance of the fixed NIDS model, which identifies the necessity of an adaptable design. The model is effective on large attack type, providing the optimal F1-scores on DDoS and PortScan traffic. Nonetheless, it is not very effective with small Bot attacks (F1 -score 0.55, Recall 0.38), which proves that high average accuracy is not synonymous with complete security.The next step to a more flexible system would be to create a more sophisticated approach in the future which can draw the time-specific aspects that distinguish between stealthy attacks and regular noises. The existing stack design provides a base upon which adaptability can be provided, although in order to maintain good performance under changing networks with continuous concept drift, it requires external tools to monitor the ongoing changes and refresh the model.

The use of a wide variety of differing models simultaneously has formed a solid base on which future study can be carried out by concentrating on three keys to real-time security, first, the low detection of subtle attacks should possess features beyond manual extraction. Deep Learning architectures, in particular, recurrent neural networks with convolutional layers (such as CNN -LSTM) and potentially attention mechanisms, should be explored in future work. CNN layers are effective at identifying local patterns, LSTMs identify time relationships that are significant to identify stateful attacks, and attention allows the model to focus on the most important features.

The performance difference between the CICIDs 2017 and the NFS- 2023-TE indicates that networks evolve. One of the future directions is the construction of an Adaptive Drift Detection Framework. This system would monitor the real time incoming data of the network using statistical means. When it observes a large alteration in the information or the performance of the model, it would initiate incremental learning or rapid model transfer in order to update the stacking model or reeducate the base learners using the new instances. Future research should consider a Blockchain Antonado Federated Learning architecture to support the requirements of large, distributed networks, such as the IoT or multi-company design. Federated Learning also allows numerous organizations to jointly train the high-performance ensemble without access to raw traffic information, ensuring privacy and compliance with such regulations as GDPR. By adding Blockchain, there is a secure and transparent layer by:

- Making sure that all model updates that are distributed by participants can be tracked and audited.
- Decreasing the threat of model poisoning or tampering with data by use of immutable ledgers and consensus.

This combination provides the secure, privacy preserving and scalable manner of distributing the resilient anomaly detection model through diverse and untrusted network environments.

REFERENCES

1. D. E. Denning, “An intrusion-detection model,” *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222-232, Feb. 1987.
2. M. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines,” in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2002, pp. 1702-1707
3. V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: a survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, Jul. 2009.
4. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Computers & Security*, vol. 28, no. 1-2, pp. 18-28, Feb. 2009.
5. Z. Wang, Y. Liu, D. He, and S. Chan, “Intrusion detection methods based on integrated deep learning model,” *Computers & Security*, vol. 103, Apr. 2021.
6. A. Alsaleh and W. Binsaeedan, “The influence of salp swarm algorithm-based feature selection on network anomaly intrusion detection,” *IEEE Access*, vol. 9, pp. 11 2466-11 2477, Aug. 2021.
7. Q. V. Dang, “Studying the fuzzy clustering algorithm for intrusion detection on the attacks to the domain name system,” in *2021 5th World Conf. Smart Trends Syst. Secur. Sustainab. (WorldS4)*, London, UK, 2021, pp. 271-274.
8. J. Kumari and A. K. Mehta, “A hybrid intrusion detection system based on decision tree and support vector machine,” in *2020 IEEE 5th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Greater Noida, India, 2020, pp. 396-400.
9. S. Waskle, L. Parashar, and U. Singh, “Intrusion detection system using PCA with random forest approach,” in *2020 Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Coimbatore, India, 2020, pp. 803-808.
10. H. Zhao et al., “A graph attention network-based intrusion detection system for IoT,” *IEEE Access*, vol. 11, pp. 54 211-54 222, 2023.
11. Y. Zhou, J. Liu, and H. Chen, “A hybrid deep learning approach for network intrusion detection based on CNN and LSTM,” *IEEE Access*, vol. 11, pp. 98 765-98 778, 2023.
12. M. Latah and I. Toker, “An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks,” *CCF Transactions on Networking*, vol. 3, no. 3, pp. 261-271, 2020.
13. Z. Maseer, R. Yusof, B. Al-Bander, A. Saif, and Q. K. Kadhim, “Meta-analysis and systematic review for anomaly network intrusion detection systems: Detection methods, dataset, validation methodology, and challenges,” *arXiv preprint*, Aug. 2023.
14. X. Sáez-de-Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, “Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks,” *arXiv preprint*, Mar. 2023.
15. R. Almuhanna and S. Dardouri, “A deep learning/machine learning approach for anomaly based network intrusion detection,” *Frontiers in Artificial Intelligence*, vol. 8, 2025.

16. A. Ali, S. Naeem, S. Anam, and M. M. Ahmed, “Machine learning for intrusion detection in cyber security: applications, challenges, and recommendations,” *Innovative Computing Review*, vol. 2, no. 2, pp. 41-64, 2022.
17. J. Rose, M. Swann, G. Bendiab, S. Shiaeles, and N. Kolokotronis, “Intrusion detection using network traffic profiling and machine learning for IoT,” *arXiv preprint*, Sep. 2021.
18. E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, “Anomal-E: A self-supervised network intrusion detection system based on graph neural networks,” *arXiv preprint*, Jul. 2022.
19. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *arXiv preprint*, Apr. 2019.
20. “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,” *Knowledge-Based Systems*, vol. 189, pp. 105124, 2020.

