Network Intrusion Detection Using ADASYN and Hybrid Residual Blocks

Ashwini Mahadev Rahod

[1] Student, Computer Science Department, UVCE, Karnataka, India

ABSTRACT

An approach based on adaptive synthesis is presented to address the issues of low accuracy in classification and inadequate small sample feature collection of current intrusion traffic detection models. Enhanced residual network approach that makes use of Inception-Resnet and sampling modules. When applied to unbalanced data sets, this approach can efficiently optimize sampling and enhance the model limited capacity to extract features from a small sample. The unbalanced data training set is first oversampled to enhance the data distribution, then the non-data portion is subsequently separately. To simplify preparation, hot storing is processed and combined with the data portion. Lastly, data training, algorithm performance comparison, and performance evaluation are conducted using the refined residual network model. According to experimental findings, the enhanced residual network model can identify intrusion traffic with an accuracy of between 89.40% and 91.88%. The enhanced residual network model outperforms the traditional deep learning technique for performance, dependability & to detect intrusion.

Keyword: - *unbalanced data collection, residual neural network, adaptive synthetic sampling, and intrusion traffic detection*

---

1. Introduction

As 5G technology gains traction, internet connections are being implemented in larger and more intricate settings. Yet, because wireless networks are dispersed and open, they are becoming the primary target of attacks [1]; as a result, network security concerns are receiving more attention. Typical examples of network safety technologies are firewalls, encryption, as well as authentication systems. The reliability of network security mechanisms becomes apparent when traditional security technologies fall short [2]. Network attacks, including Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and probing attacks, pose significant threats. In this context, attack detection plays a crucial role, wherein Intrusion Detection Systems (IDS) classify attacks based on acquired attributes [3]. This classification helps in early anomaly detection and the implementation of necessary preventive measures.

In today's landscape, conventional machine learning methods are frequently employed for detecting attacks, encompassing essential stages like gathering data, selecting relevant features, and classifying them. Feature encoding, a prevalent training strategy, enhances the classifier's capacity to discern diverse attributes [4]. Different encoding strategies lead to different levels of discretization for the processed attributes, and the general trend is that increased discretization improves the performance of the model. The accuracy and efficiency detection directly affect the analysis and selection of features [6].

Qi and colleagues [7] significantly improved the efficiency of IDS by using the PCA method to select important features. They reduce feature dimensionality using correlation-based feature selection (CCS) [9] and optimal linear

discriminant analysis (LDA). Wang et al. [10] presented a general algorithm for increasing and decreasing features using edge density to produce a higher set of attributes.

Classification stands as the cornerstone in traditional intrusion detection methodologies, playing a pivotal role in preparing data and accurately categorizing network traffic. Xiao et al. introduced a straightforward Bayes-based classifier for attack detection [11]. Moreover, decision tree (DT), support vector machine (SVM), and artificial neural network (ANN) classifiers are commonly utilized for decision-making tasks. Recognizing the distinct advantages of various classifiers, Peng et al. [16] assessed the detection capabilities of different classification methods and devised a hybrid attacker classification predictor by amalgamating decision networks and k-means. Similarly, Tahir et al. [17] proposed a hybrid approach that combines SVM with k-means clustering.

The mentioned computational techniques in artificial intelligence (ML) rely on specific qualities and face challenges in adapting to the dynamic changes in network environments. In contrast, deep learning (DL) has gained prominence in battle recognition tasks due to its ability to autonomously generate concise key features in simple network communication without the necessity for intricate feature engineering. Tan and colleagues [18] employed particle swarm stabilisation (PSO) to improve an structure-optimal deep neural system (DBN). PSO-DBN demonstrated 92.44% accuracy when using the KDD99 dataset, according to experiments. Marir et al. [19] employed DBN and a component selection in combination with a classifier using SVM to improve the identification accuracy of IDS. Recurrent neural networks were used by Yin et al. [20] to tackle an intrusion detection job, and they produced amazing results on the NSL-KDD dataset. By using weight sharing and local perception, CNN lowers the model's complexity in comparison to the DL algorithms mentioned above. Furthermore, CNN's abstracted features frequently outperform conventional feature selection methods. As a result, CNN-based intrusion detection algorithms frequently exhibit higher accuracy when compared to traditional DL techniques. More training samples often help CNN learn stronger features, and the model's ability to discriminate is greatly influenced by these higher-quality features. Therefore, in today's dynamic network environment, theft detection algorithms thru Neural Networks (CNN) are considered more suitable. Although CNNs are now effectively used in various applications for intrusion detection [21], [22], the impact of data heterogeneity, including redundancy between channels, on model training has been neglected. Wu et al.] This lossy-based method greatly mitigates the negative effect of imbalance in the sample distribution on model performance. The ongoing research aims to further improve the model's recognition performance.

In general, the more convolutional layers are extracted, the more feature maps are recovered. However, this diversity may reveal some excessive data redundancy. Interestingly, the CNN-based attack detection method ignores the effect of channel component heterogeneity on frame classification accuracy. In conclusion, the IDS reported by CNN has two problems to be solved: 1. The sample distribution is uneven; 2. Data redundancy between channels. Finally, we develop a unique attack recognition method using residual network and ADASYN data augmentation algorithm..

This essay is structured into distinct sections as outlined below: Section II presents relevant literature, Section III provides a framework description, and outlines the optimization problem. The development of a two-phase alternative optimization technique is detailed in Section IV, followed by an evaluation of its effectiveness in Section V. Finally, Section VI concludes this undertaking.

2. LITERATURE SURVEY

Three phases have been included in the growth of attack recognition technology: deep learning, machine learning, and pattern matching algorithms. Feature-matching-based intrusion detection tasks were the first to employ the pattern matching method. Wu and Shen examined BM and AC, two traditional pattern matching algorithms, in [24] and suggested BMHS and AC-BM, which are correspondingly upgraded algorithms. Experiments shown that the

updated methods greatly improve IDS timeliness. The RabinKarp & Knuth- MorrisPratt pattern-matched algorithms were put through intrusion detection tests by Dagar et al. [25] to evaluate their execution efficiency. Nevertheless, it is difficult to adapt the aforementioned pattern matching algorithms to the contemporary network environment due to the wide variety of network threats nowadays.

Given their exceptional performance and seamless integration into intrusion detection systems (IDS), machine learning-based algorithms have swiftly supplanted traditional pattern matching techniques. Support Vector Machine (SVM), a model for supervised learning, has garnered extensive usage in the field of machine learning. Thaseen and Kumar [26] presented an innovative intrusion detection approach using multi-class SVMs and least-squares feature selection. Their simulation shows a significant improvement in model accuracy and performance achieved by eliminating redundant components. Ingre and collaborators [27] pioneered a groundbreaking intrusion identification system by integrating decision trees with a relevant aspect filtering technique. A dynamic iterative feature selection technique was created by Nancy et al. [28]. by adding to and fusing convolutional neural networks with the decision tree technique. An approach for intelligent fuzzy contextual decision trees was suggested by them. Using the KDD Cup dataset, the new system produced a high recognition rate of unknown assaults. In [29], a better IDS based on a feature selection method and a Bayesian network was put out. These machine learning detection techniques outperformed other methods in detecting intrusions tasks, but they also need extensive feature engineering and have highly adjustable model parameters. But without sophisticated feature engineering, the DL algorithms can automatically extract features from simple network data. As a result, the DL approach is becoming the subject of relevant intrusion detection research.

Employing a gradient descent optimizer, intrusion detection systems (IDS) utilize LSTM classification to effectively uncover temporal correlations among features [30]. Moreover, the BAT model employs bidirectional LSTM to capture coarse information, while an attention mechanism sifts through the network flow path generated by the BLSTM model to isolate key features for network traffic classification.

Gao et al. [33] devised an efficient attack recognition method by combining neural network (DNN) techniques with association rules, minimizing the correlation between individual features and labels using an a priori approach to enhance recognition accuracy. Additionally, Inin et al. [20] crafted a robust attack recognition model by merging a recurrent reinforced neural network (RNN) with a feature enhancement technique. However, it's worth noting that the adopted feature enhancement strategy also escalates the computational demands of the model. Within the realm of intrusion detection challenges, CNNs have demonstrated success owing to their heightened capability to extract features from internet data [34].

Lin et al. [5] introduced the CL-CNN model at the token level, aiming to enhance the accuracy of IDS detection. Wu et al. [23] further advanced the field by developing a CNN model with character-based encryption, offering heightened security. Their experiments underscored the necessity of converting raw data into a 2D representation, highlighting the efficacy of utilizing complex CNN architectures with 2D transformed data for improved detection efficiency compared to RNN approaches [20]. Additionally, Ding and Zhai [35] proposed the MS-CNN, which leverages multi-stage features to boost model expressiveness. These features are amalgamated using a softmax classifier, resulting in increased model capacity by capturing additional information. Etang and Wan [36] employed the relational CNN framework to extract diverse features, significantly enhancing the model's expressiveness and efficacy.

The interchannel redundant information in the convolution layer is ignored by the aforementioned attack recognition algorithms, despite the fact that they use CNN to increase detection accuracy. However, since we are unsure of which channels are redundant, we are unable to immediately delete certain channel information. A split-based instantaneous processing (SPC) block was proposed by Zhang et al. [37] as a workable solution of the interchannel data redundancy issue. This partitioning of the convolution layer's channel into a representational portion and an uncertain redundant section is followed by hierarchical processing. Drawing inspiration from this concept, we introduce and implement an SPC-equipped CNN (SPC-CNN) specifically designed for tasks related to attack detection. The results of the simulation indicate that SPC-CNN surpasses the conventional CNN model in overall performance. Moreover, the ADASYN incremental data plan model is employed to address the challenge of undersampling small samples while being less susceptible to larger samples. Lastly, the AS-CNN model, SPC-CNN, and ADASYN algorithm are jointly utilized for intrusion detection tasks. The simulation demonstrates that the combined AS-CNN model outperforms SPC-CNN in isolation.

3. SYSTEM MODEL

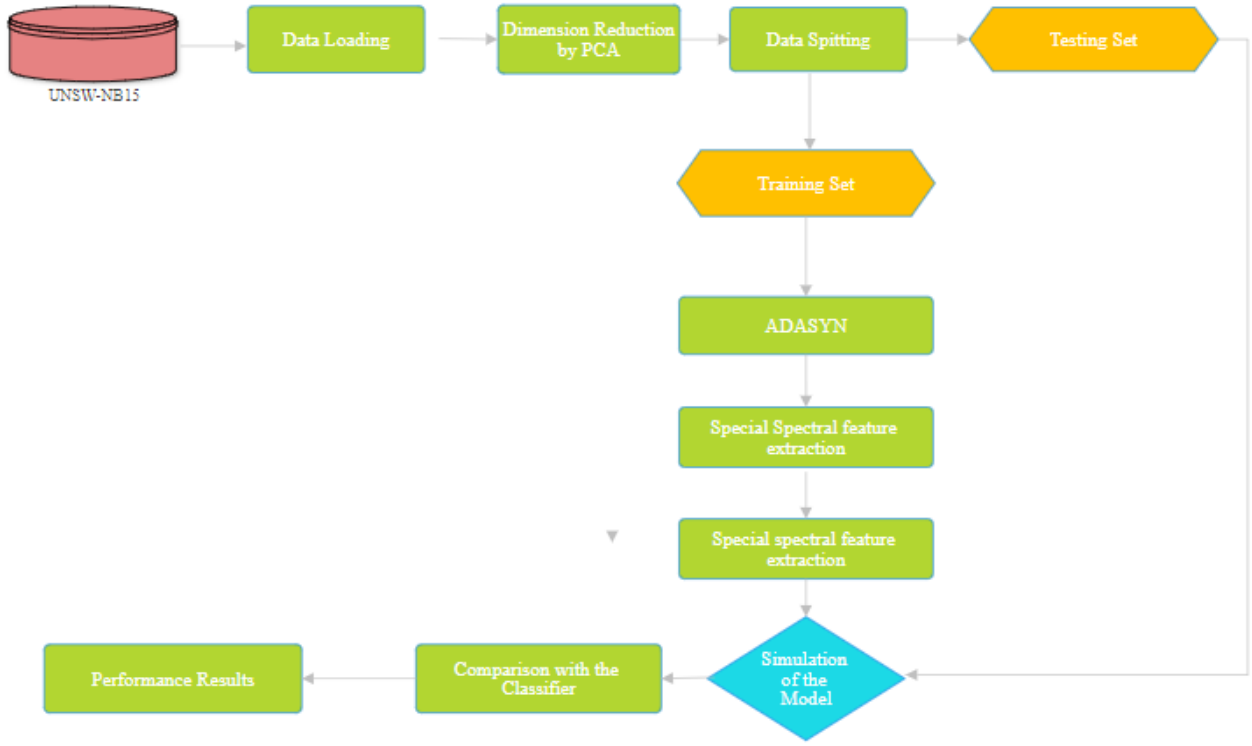

Fig 1: The overview of the proposed methodology

3.1 Method of Hybrid Sampling Integrating RENN with ADASYN

This study proposes a hybrid sampling strategy that combines ADASYN and RENN. The basic concept involves creating two sets of samples—one for the majority class and one for the minority class—from the original dataset. The DBSCAN clustering method is then employed to filter out noise from the new sample set. Subsequently, the two datasets are merged to construct a balanced database. The RENN method is utilized to represent the majority of the population, while the ADASYN algorithm is employed to augment minority samples. A unique approach is adopted in this combined sampling strategy, wherein the algorithm takes the initial sample from the minority class (P), along with the desired number of samples and their proportions. The algorithm generates two sets of examples: one containing pairs from the majority class (newN) and the other containing an equal number of examples from the minority class (newP).

The steps involved in this sampling strategy are as follows:

1. Assess the degree of imbalance in the database.
2. Calculate the proportion by identifying each sample in the majority class (N) and determining its $k1$ nearest neighbors. Normalize the proportions and determine the number of samples to synthesize for each minority sample.
3. Generate gi samples for each sample in the majority class (N) to create a new set of minority samples.
4. For each sample in the minority class (P), select $k22$ nearest neighbors from newN.
5. Determine the number of minority samples within the $k2$ nearest neighbors of each sample and remove the sample if the total exceeds a specified threshold (e.g., e = 1).
6. Repeat steps 4 and 5 to create a new sample collection for the majority class.
7. Deduct the samples from newP and newN to obtain the final newN and newP sets.

3.2 ALGORITHM

In this part, we provide a multitask-based, two-stage alternating optimisation method that can solve the two aforementioned subproblems.

ADASYN

Start

Input: - Samples
- Labels
- Target label
- N (number of synthetic samples to generate)
- Beta (parameter for synthetic sample generation)

Step1: Retrieve minority samples and majority samples based on target label

Step2: For each minority sample:
Find k nearest neighbors

Step2: Compute G (ratio of minority samples in k-nearest neighbors)

Step3: For each synthetic sample to generate (N times):
Randomly select a neighbor index

Step4: Generate synthetic sample using neighbor and Beta

Step5: Add synthetic sample to the list Output synthetic samples

End

An advancement of the SMOTE algorithm is the ADASYN Adaptive Synthetic Sampling technique. He (2008) established the notion of the Adaptive Synthetic (ADASYN) sampling process for unbalanced learning. To improve the identification of positive, ADASYN attempts to produce more synthetic cases on the area with less positive occurrences than one containing more positive instances. This approach creates a distribution function by counting the number of instances that are negative neighbours in each positive instance's K-nearest neighbours. The number of synthetic instances produced from this positive instance is determined by the distribution function [8]. When minority samples are harder to learn than easier minority sample classes, ADASYN generates more computer-generated points of information and observations. The fact that ADASYN generates a set number of instances for each minority instance using weighted distributions of its neighbours makes it ultimately a pseudo-probabilistic algorithm [4]. While ADASYN utilises a density distribution as an indicator parameter for automatically assessing the quantity of synthetic data specimens to be created for every case of minority sample group information, SMOTE gives each minority specimen class an equal chance of being picked during the development of the artificial information samples.

Thus, by shifting the classification determination boundary to the difficult cases and mitigating the bias caused by the class disparity, the ADASYN technique enhances data distribution learning (He et al., 2008). It is best to deal

with aberrations during data preparation before using the ADASYN technique since ADASYN is highly sensitive to outliers. By producing artificial points of data for these challenging and challenging to grasp min samples, ADASYN, in contrast to SMOTE, focused more on the minority specimens that are challenging to learn.

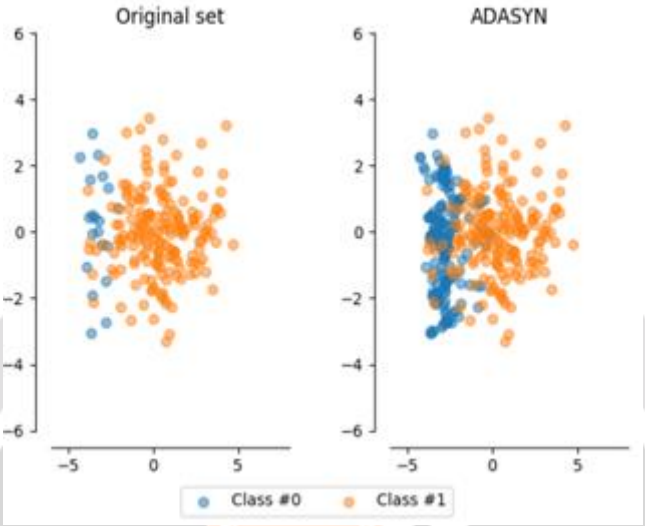

Fig 2: Schematic diagram of ADASYN algorithm application

3.3 Residual Networks

Moving forward, evaluations were performed on 18-layer and 34-layer residual networks (ResNets). The basic design is exactly the same as the direct network mentioned above, including a short connection for each pair of 3x3 filters. Unlike its simple counterparts, there are no additional features.

After repeated training, there is an interesting reversal in the order of magnitude: 34-layer ResNet outperforms 18-layer ResNet by about 2.8%. In addition, the 34-layer ResNet shows improved evaluation results and significantly reduced learning. This means that the degradation problem is effectively solved in this context and allows using a deeper architecture for better accuracy.

Furthermore, due to the effective reduction of the training error, the 34-layer ResNet reached a maximum of 1 error of 3.5%, showing a significant improvement compared to its simple counterpart. This comparison shows the effectiveness of the residual training in the deep network.

Finally, although the 18-layer ResNet converges quickly, the smoothness of the 18-layer and the remaining networks show similar accuracy. Even with 18 layers and extreme depth, the Stochastic Gradient Descent (SGD) solver can now find optimal solutions for simple networks. In this scenario, ResNet helps in optimization by facilitating early convergence earlier.

4. Experimental Setup

4.1 Parameters used for evaluation

TP is the number of attack traffic detected by the network model, and the result is correct. FN is the total number of wrong attacks detected.TN indicates the normal traffics that is detected, and the detection result is correct. FP indicates normal traffic detected, and the detection result is wrong, the traffic is nothing but an attack flow. **Accuracy** is the proportion of the overall sample that the classifier correctly classifies the sample, higher the accuracy value, the better is the performance of the network intrusion detection model. Accuracy is calculated as follows

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

Recall is the proportion of the number of samples predicted as positive to the total positive samples. Higher the Recall value then higher is the performance. Recall is calculated using below equation

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

Precision represents the proportion of positive samples in the positive examples classified by the classifier. Higher the value better the false positive performance of the network intrusion detection model. Calculated as equation

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

F1-score is nothing but the weighted average of precision and recall. Higher the value, the closer the precision and recall are to 1, and the better the detection performance of the model for the traffic that is normal. Calculated as equation

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

4.2 Dataset

The UNSW-NB15 dataset, created in 2015 by the Australian Cyber Security Center (ACCS) Cyberspace Intelligence Laboratory, is generated using IXIA's PerfectStorm software to emulate real-world cyber environments. Comprising 47 features categorized into two groups, the dataset encompasses nine attack methods, including spoofs, probes, Trojans, denial-of-service, generic, exploration, shellcode, and worms. The performance of the model in this task is assessed using the initial training and test sets.

4.3 Data Pre-Processing

This paper's pre-processing steps include: (1) converting the original flow data's non-numerical features into numerical characteristics and normalizing them; (2) sampling using a combining algorithm that combines ADASYN & RENN; (3) using the technique of feature selection to select features; and (4) converting the resulting data into a grey-scale map. Fig 1 illustrates this stage's particular procedure.

5. Experiment Results

Both numerical and non-numeric data are included in the set of information that is loaded below; however, as algorithms are limited to processing data that is numerical, we have to process the data with the goal to convert it to numeric.

In the, Chart 1, the y-axis displays the total number the records for both the normal as well as attack categories, while the x-axis indicates 0 overall NORMAL and 1 for ATTACK. Chart 2 shows the Training Validation Accuracy and Loss. Once we upload the dataset the details are given as shown in the Figure 3. Once we close the graph above, we use the "Preprocess Dataset" option to process this data set and get the output that is displayed in the figure 4.

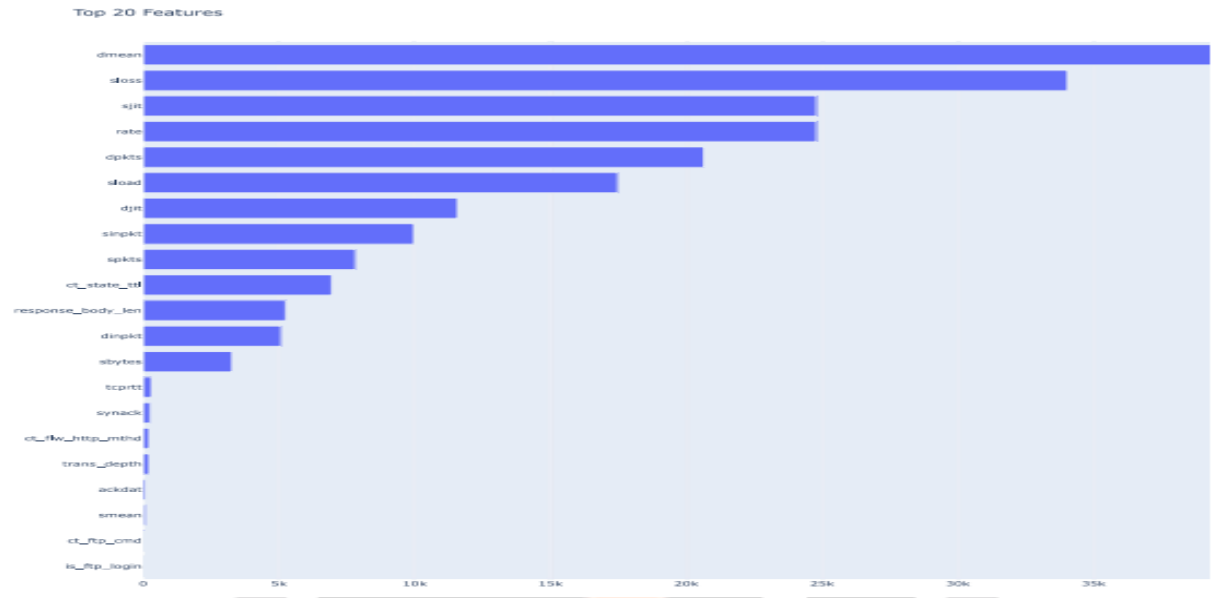

Chart 1: Different Features

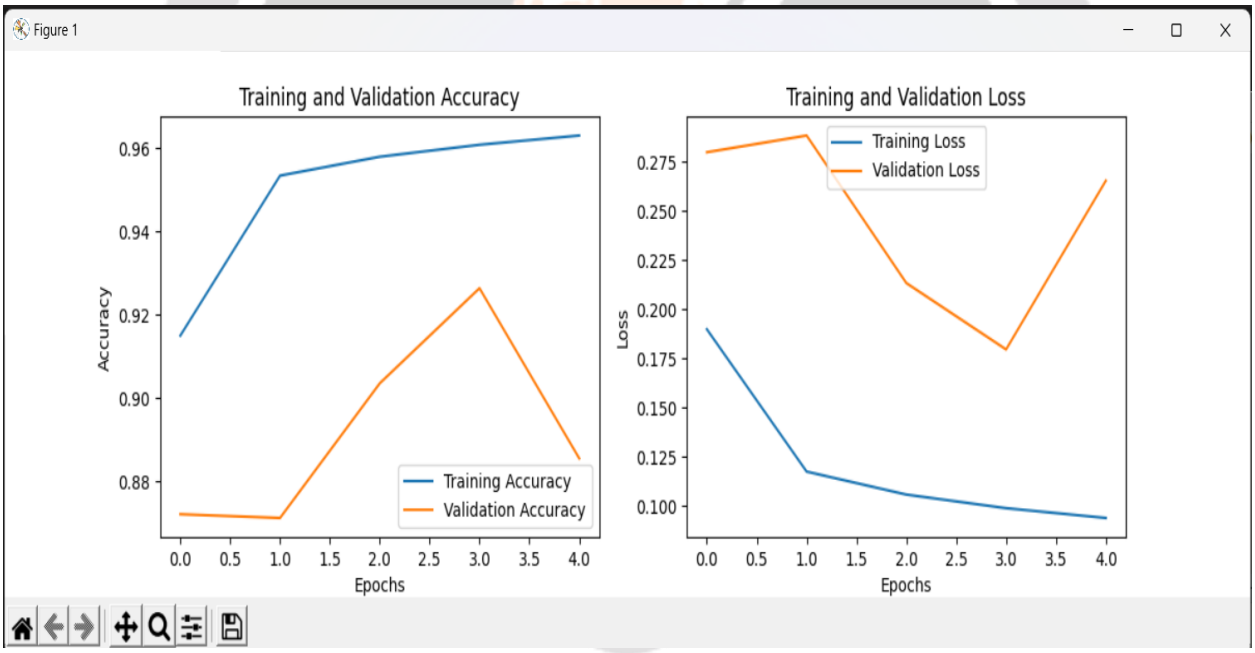

Chart 2: Validation accuracy and loss

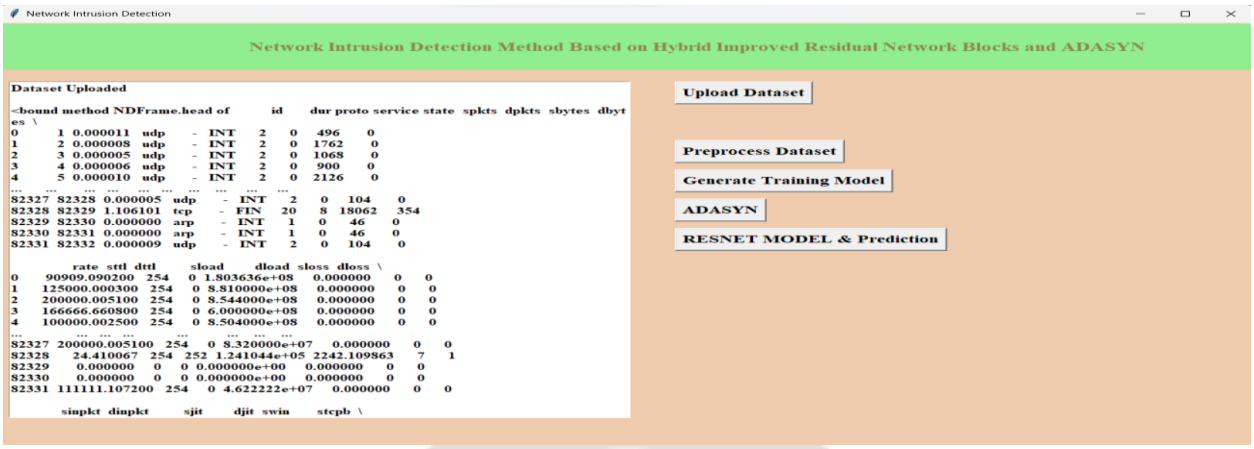

Fig 3: Data Uploading

In pre-processing the numerical characters are removed from dataset and it is saved inside clean.txt file. The Dataset information is displayed in the Fig 4.

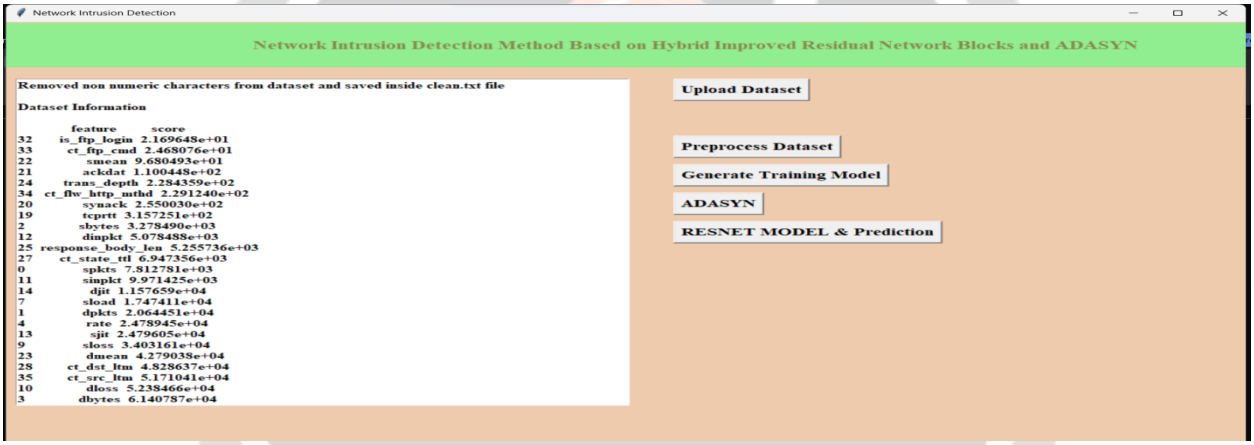

Fig 4: Data Pre-processing

After the pre-processing the data, the pre-processed data is divided into Training set and Testing set respectively as shown in the Fig 5.

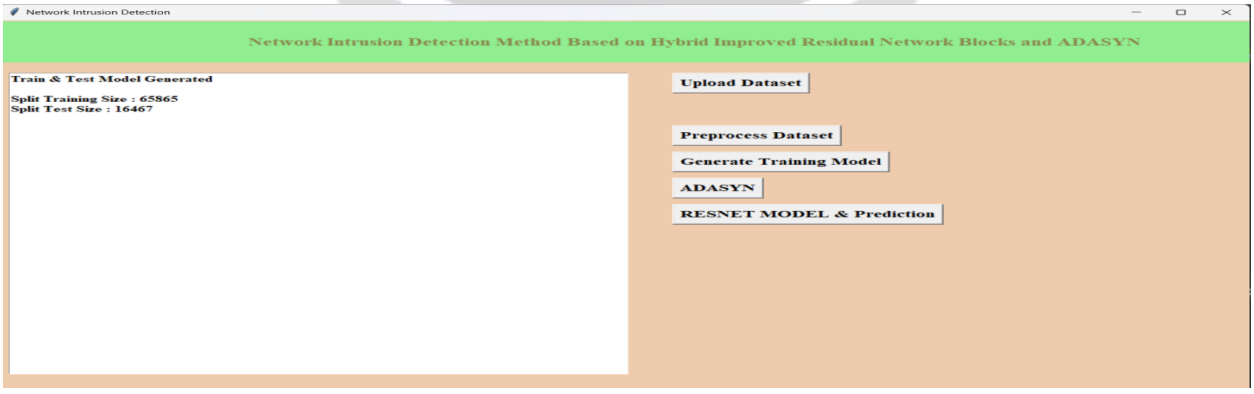

Fig 5: Splitting of Data

The data imbalance problem is overcome by "Adaptive synthetic sampling' method. In this method the data is balanced by creating the synthetic samples of the minority class data. This prevents model being biased towards the majority samples ignoring the minority samples. The dataset before and after ADASYN, which uses Smote to eliminate dataset imbalance are depicted in the following Chart 3.

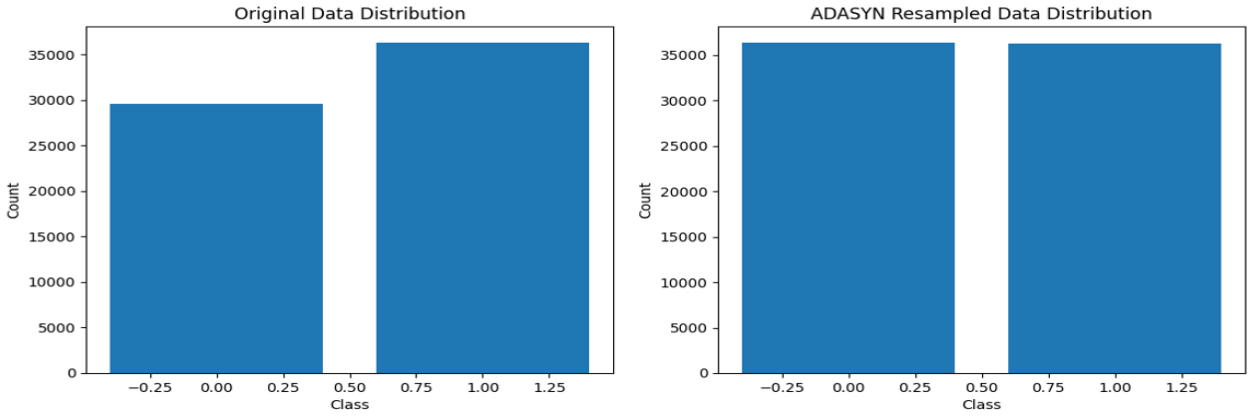

Chart 3: ADASN Resampling

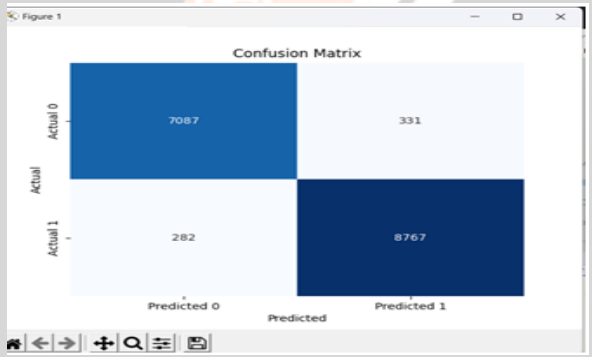

Chart 4: Confusion Matrix

The confusion matrix corresponding to the ResNet neural network modelling method is shown in the above Chart 4.

Table 1: Comparison

Algorithm	Accuracy	Precision	Recall	F1 score
ADASYN+SPCNN	92.15%	92.85%	92.69%	92.15%
Nave Byes	85.35%	85.15%	85.96%	85.12%
SVM	78.11%	69.25%	73.15%	70.20%
ADASYN+ResNet	96.25%	96.10%	96.75%	95.85%