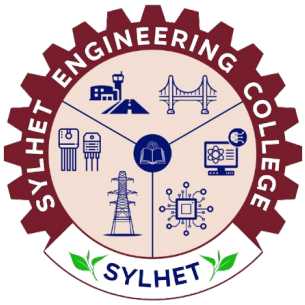
Department of Computer Science and Engineering

Sylhet Engineering College

Shahjalal University of Science and Technology



Beyond Binary: Multiclass Intrusion Detection Using Convolutional and Recurrent Deep Networks

Md. Habibul Islam Antar
Reg. No: 2019331504
4th year, 2nd semester
Dept. of CSE
Sylhet Engineering College

Fayad Mohib Khan
Reg. No: 2019331557
4th year, 2nd semester
Dept. of CSE
Sylhet Engineering College

Tanbin Hasan Ador
Reg. No: 2019331528
4th year, 2nd semester
Dept. of CSE
Sylhet Engineering College

Supervisor

Md Lysuzzaman

Lecturer

Department of Computer Science and Engineering

Sylhet Engineering College

Candidates Declaration

This is to certify that the work presented in this thesis, titled, *"Beyond Binary: Multiclass Intrusion Detection Using Convolutional and Recurrent Deep Networks"*, is the outcome of the investigation and research carried out by us under the supervision of **Md Lysuzzaman**.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree or diploma.

Md. Habibul Islam Antar	**Fayad Mohib Khan**	**Tanbin Hasan Ador**
Reg. No : 2019331504	Reg. No : 2019331557	Reg. No : 2019331528
4th year, 2nd semester	4th year, 2nd semester	4th year, 2nd semester
Department of Computer Science and Engineering	Department of Computer Science and Engineering	Department of Computer Science and Engineering
Sylhet Engineering College	Sylhet Engineering College	Sylhet Engineering College

Recommendation Letter from Thesis Supervisor

The thesis entitled *"Beyond Binary: Multiclass Intrusion Detection Using Convolutional and Recurrent Deep Networks"* submitted by the students **Md Habibul Islam Antar**, **Fayad Mohib Khan**, and **Tanbin Hasan Ador** is a record of research work carried out under my supervision and I, hereby, approve that the report be submitted in partial fulfillment of the requirements for the award of their Bachelor Degrees.

Md Lysuzzaman
Lecturer
Department of Computer Science and Engineering
Sylhet Engineering College
Date : 22/07/2025

Acknowledgment

We would like to thank the Department of Computer Science and Engineering, Sylhet Engineering College, Sylhet, for supporting this research.

We are very thankful to our honorable supervisor **Md Lysuzzaman** (Lecturer, CSE, Sylhet Engineering College) for his worthy support and direction.

We would also like to thank our department head **Abu Naser Mojumder** (Assistant Professor & Head of Department) for providing us with necessary academic environment and resources throughout the research.

Abstract

The inadequacy of traditional binary intrusion detection systems (IDS) against sophisticated cyberattacks necessitates advanced solutions. This thesis, ”Beyond Binary: Multiclass Intrusion Detection Using Convolutional and Recurrent Deep Networks,” addresses this by developing a deep learning pipeline for fine-grained multiclass intrusion classification using the CICIDS2017 dataset, encompassing 15 classes from benign traffic to complex attacks (e.g., DDoS, SQL Injection, Botnets). The dataset was preprocessed with SMOTETomek to mitigate class imbalance, and XGBoost selected the top 20 most relevant features. We rigorously evaluated several deep learning models—GRU, Bidirectional GRU, and AlexNet—using 5-fold cross-validation and metrics including accuracy, precision, recall, F1-score, and confusion matrices. Among these, AlexNet demonstrated the highest accuracy and generalization capability for the 15-class task. This multiclass approach significantly enhances detection granularity compared to binary systems, enabling more targeted security responses. By integrating systematic preprocessing, effective feature selection, and comparative deep learning analysis, this work provides a reproducible pipeline for accurate, fine-grained intrusion detection in complex network environments, improving IDS utility for practical security deployments.

Keywords: Intrusion Detection System (IDS), Multiclass Classification, Deep Learning, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU), AlexNet, CICIDS2017, Cybersecurity, SMOTETomek, XGBoost, Network Security

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Overview

Intrusion Detection Systems (IDS) are vital for securing networks by identifying malicious activities and breaches. As cyberattacks grow in complexity—ranging from DoS to multi-vector threats targeting IoT and industrial systems—more advanced IDS techniques are needed. Traditional machine learning (ML) methods like decision trees, SVMs, and random forests have performed well on datasets such as NSL-KDD and CICIDS2017. However, deep learning (DL) models, including CNNs, RNNs, LSTMs, and transformers, have further improved detection, particularly for novel attacks.

Hybrid approaches that combine feature selection, data balancing, and ensemble classifiers enhance detection rates and reduce false positives. Emerging trends like explainable AI (XAI) and federated learning (FL) address transparency and privacy in real-world, distributed IDS deployments. This study builds on these advances by integrating sophisticated feature selection and deep learning classifiers, evaluating their effectiveness across multiple multi-class intrusion datasets to support robust and scalable IDS solutions.

1.2 Research Motivation

Despite advances in ML and DL-based Intrusion Detection Systems (IDS), key challenges hinder their real-world deployment. Multi-class intrusion detection remains particularly difficult, with performance often dropping due to class imbalance, feature overlap, and underrepresented attacks like U2R, R2L, and infiltration [9][11]. While binary classifiers frequently exceed 99% accuracy, achieving similar results across all 15 classes in the CICIDS2017 dataset remains elusive [12].

DL models such as CNNs, LSTMs, and hybrid networks show promise but often suffer from overfitting, high computational costs, and limited interpretability, reducing their suitability for real-time or edge applications [18][20][30]. Conversely, traditional ML models like decision trees and random forests offer efficiency and interpretability but struggle with complex, high-dimensional data [8][14].

These limitations drive the need for IDS frameworks that combine high accuracy, low false-

positive rates, and practical deployment efficiency across diverse attack types.

This research aims to answer the following questions:

- Can all 15 attack classes in the CICIDS2017 dataset be effectively detected with high accuracy using a deep learning model trained on a reduced feature set?
- How does using feature selection methods affect the performance of multi-class intrusion detection, particularly in terms of precision, recall, and false positive rate?
- To what extent can SMOTE-Tomek overcome class imbalance in CICIDS2017, especially for rare intrusion types, without introducing noise or harming overall performance?
- What model architecture and training configuration are best suited for handling real-world, high-dimensional intrusion data across multiple classes?

1.3 Objectives

This study aims to develop a robust, high-accuracy multiclass intrusion detection pipeline by systematically exploring the impact of dataset preprocessing, feature selection, and deep learning model architectures on the CICIDS2017 dataset. The specific objectives of the research are as follows:

- To preprocess and balance the CICIDS2017 dataset using SMOTE-Tomek to mitigate class imbalance across 15 attack and normal traffic classes.
- To perform feature selection using XGBoost, ranking features by importance and selecting the top 20 for improved model performance and reduced computational overhead.
- To implement and evaluate multiple deep learning models—namely AlexNet, GRU, and BiGRU—for multiclass classification of network intrusions.
- To conduct 5-fold cross-validation and holdout testing, ensuring reliable generalization and mitigating overfitting risks.
- To assess model performance using accuracy, precision, recall, F1-score, and confusion matrices, with attention to both overall and per-class effectiveness.
- To identify the best-performing model architecture based on experimental outcomes and

analyze trade-offs in terms of classification performance, complexity, and suitability for real-world intrusion detection systems.

Chapter 2
Background Study

2.1 Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are essential components in modern cybersecurity infrastructures, tasked with monitoring and analyzing network traffic to detect unauthorized access, misuse, or anomalies. IDS can be broadly categorized into signature-based and anomaly-based systems. Signature-based IDS detect known threats by matching patterns, while anomaly-based IDS leverage statistical or machine learning models to identify deviations from normal behavior, making them suitable for detecting zero-day and evolving attacks.

2.2 Binary vs. Multiclass Classification in IDS

Historically, many IDS models have adopted a binary classification approach, labeling traffic simply as benign or malicious. While effective for general detection, this method fails to differentiate between types of attacks—limiting its practical utility for real-time incident response. Multiclass classification, on the other hand, allows the IDS to distinguish among various specific attack types, enabling tailored mitigation strategies. For example, a DDoS attack may warrant immediate IP blocking, while a SQL Injection might require input validation and database protection. This finer granularity improves both detection capability and defensive response planning.

2.3 Deep Learning for Intrusion Detection

Deep learning has shown remarkable success in complex pattern recognition tasks, making it a promising technique for network intrusion detection. Unlike traditional machine learning models, deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can automatically extract hierarchical features and learn temporal dependencies. This makes them highly suitable for modeling sequential and high-dimensional network traffic data. Their ability to generalize from diverse input distributions is particularly beneficial in handling the heterogeneous nature of network attacks.

2.4 Convolutional Neural Networks (CNNs)

CNNs are powerful tools originally designed for image processing, but their layered structure

also works well on structured numerical data. In intrusion detection, CNNs can identify local patterns in traffic features—analogous to recognizing textures in images. AlexNet, a deep CNN architecture, is particularly effective due to its depth, non-linearity, and regularization capabilities. When applied to network traffic, AlexNet can differentiate subtle patterns across multiple attack types, making it well-suited for multiclass IDS tasks.

2.5 Recurrent Neural Networks (RNNs) and GRUs

RNNs are specifically tailored for sequential data processing. In network traffic analysis, the temporal dimension is critical, as many attacks unfold over time. Gated Recurrent Units (GRUs) are an advanced variant of RNNs that mitigate vanishing gradient problems and accelerate training. Bidirectional GRUs (BiGRUs) extend this by processing sequences in both forward and backward directions, improving context understanding and detection accuracy. These models are effective in capturing long-range dependencies inherent in attack behaviors.

2.6 Temporal Convolutional Networks (TCNs)

Temporal Convolutional Networks (TCNs) offer an alternative to RNNs for time-series data. By using causal and dilated convolutions, TCNs can capture long-term dependencies without the sequential bottleneck of RNNs. TCNs are particularly efficient in scenarios requiring high-throughput and low-latency inference, making them suitable for scalable IDS solutions.

2.7 Datasets in Intrusion Detection: CICIDS2017

The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, is widely regarded as one of the most comprehensive and realistic IDS datasets available. It simulates real-world network traffic and includes a broad spectrum of up-to-date attacks, such as DDoS, web exploits, and brute-force login attempts. The dataset is labeled and rich in features, making it ideal for training, evaluating, and benchmarking intrusion detection models.

2.8 Handling Imbalanced Data and Feature Selection

A major challenge in multiclass intrusion detection is class imbalance, where certain attack types are underrepresented. To address this, the SMOTETomek method was applied—combining oversampling of minority classes with undersampling of majority classes. This helps create a more balanced and representative training set. Additionally, XGBoost-based feature selection was used to reduce dimensionality and highlight the top 20 most influential features, leading to

improved model performance and generalization.

2.9 Application and Deployment Context

Although currently in the experimental phase, this research is intended for deployment on websites and web servers, where most of the targeted attack types occur. The ability to detect and classify multiple intrusion types in real time makes the system adaptable and actionable. For example, detecting a DDoS could trigger immediate rate-limiting or IP blocking, while a Web Attack - SQL Injection might prompt stricter input validation.

2.10 Attack Class Descriptions in CICIDS2017

The 15 classes used in this research span a broad range of real-world cyberattacks, each posing distinct threats:

- **BENIGN**: Legitimate, non-malicious traffic. Proper identification helps reduce false positives.
- **Bot**: Infected hosts controlled remotely to perform automated tasks like spamming or DDoS attacks.
- **DDoS (Distributed Denial of Service)**: Overwhelms the target system with a flood of traffic from multiple sources.
- **DoS GoldenEye**: Sends repeated HTTP requests to deplete server resources.
- **DoS Hulk**: Generates high-volume web traffic with randomized headers.
- **DoS Slowhttptest**: Simulates a slow-rate HTTP attack by holding server connections open.
- **DoS Slowloris**: Similar to Slowhttptest, keeps connections open by sending partial headers.
- **FTP-Patator**: Executes brute-force attacks on FTP servers.
- **Heartbleed**: Exploits a vulnerability in the OpenSSL Heartbeat extension.
- **PortScan**: Scans a host for open ports, often used for reconnaissance.
- **SSH-Patator**: Performs brute-force login attempts on SSH services.

- **Web Attack - Brute Force**: Repeatedly guesses login credentials on web authentication.
- **Web Attack - SQL Injection**: Inserts malicious SQL statements to manipulate backend databases.
- **Web Attack - XSS**: Injects malicious scripts into web pages.

Each attack exhibits unique traffic patterns and characteristics. Accurately identifying the attack class not only improves security monitoring but also facilitates dynamic and effective response strategies.

2.11 Summary

This chapter has presented a comprehensive background on intrusion detection, emphasizing the advantages of multiclass classification over binary approaches. It reviewed the role of deep learning models—including CNNs, GRUs, and BiGRUs—in handling complex traffic patterns, and described the use of CICIDS2017 as a benchmark dataset. Furthermore, techniques for addressing data imbalance and performing feature selection were discussed, along with a detailed breakdown of the 15 attack classes used in the study. This foundational understanding sets the stage for the proposed methodology and experimental evaluation in the subsequent chapters.

Chapter 3
Literature Review

3.1 Previous Work

During our research, we discovered some exciting work connected to our area of study:

Binsaeed and Hafez [1] investigate the integration of XGBoost-based feature selection with deep learning techniques for multi-class intrusion detection on the NSL-KDD, CIC-IDS2017, and UNSW-NB15 datasets. After applying XGBoost to identify the top 20 features and addressing class imbalance with SMOTE, they train four hybrid models—1D-CNN + BiLSTM and ANN + BiLSTM on both imbalanced and balanced data—to classify five intrusion types (Normal, DoS, Probe, R2L, and U2R). The balanced ANN + BiLSTM model achieves the best performance, with overall accuracy and F1-score exceeding 95% across all three datasets.

Dong et al. [2] propose a real-time network intrusion detection system employing deep learning techniques on the KDD-99 dataset, which comprises five million records across multiple attack categories and normal traffic. The authors utilize a self-encoder for dimensionality reduction followed by an AE-AlexNet model for classification, detecting five intrusion classes (DoS, Probe, U2R, R2L, and Normal). Their system processes streaming log data via Flume and Flink, performing real-time feature extraction and classification. Experimental results demonstrate a total detection accuracy of 94.32%, with per-class detection rates ranging from 89.76% (U2R) to 100% (Normal).

Zhao et al. [3] evaluate real-time network anomaly detection on Cisco NetFlow data from four UMKC core switches, extracting four feature groups (A–D) and deploying Naïve Bayes, SVM and decision-tree models. They classify four anomaly types—(A) source-port/destination-IP, (B) source-IP/destination-port, (C) unique destination IP and (D) unique source IP—against normal traffic and report average classification accuracy of 90.3%, with SVM achieving up to 99.90% on combined features C+D for switch 3.

Sridevi et al. [4] compare multiple supervised learning algorithms on the NSL-KDD dataset, comprising five intrusion classes (DoS, Probe, U2R, R2L and Normal). Using 20 selected features, they apply Linear Discriminant Analysis, Classification and Regression Trees, and Random Forest to classify multi-class attacks. In 80/20 train–test splits, Random Forest attains the highest accuracy (99.65%), surpassing LDA (98.10%) and CART (98.00%), demonstrating

its robustness for multi-class intrusion detection.

Shum and Malki [5] develop a host-based intrusion detection system using feedforward neural networks trained on DARPA 1998 TCPDUMP traces labeled across five classes (Normal, DoS, Probe, U2R, R2L). Employing backpropagation with two output neurons, their model achieves 100% correct classification on normal and known attack sets and 76% detection of unseen attacks, underscoring the promise and limitations of neural-network–based IDS on multi-class intrusion datasets.

Kocher and Kumar [6] systematically review recent machine-learning and deep-learning methods for network intrusion detection, analyzing benchmark datasets (e.g., KDD-99, NSL-KDD, UNSW-NB15) and models including SVM, Naïve Bayes, decision trees and various deep architectures (CNN, RNN, DBN). They note that while traditional ML approaches attain up to 99% accuracy on five-class tasks, deep-learning models further improve detection of complex and novel attacks, achieving accuracies exceeding 99.3% on multi-class benchmarks.

Krishnan et al. [7] apply supervised learning to detect malicious IoT network traffic using the IoTID20 dataset, which comprises botnet and normal traffic captured from smart home devices. They perform three feature selection methods—filter, wrapper (sequential forward/backward), and recursive feature elimination—to identify optimal subsets of eight features. Logistic regression classifiers (SVC, Random Forest, and XGBoost) are trained on these features to classify two classes (malicious vs. benign). Across all selection techniques and classifiers, the models achieve high detection accuracy (up to 99.79%) and F1-scores nearing 1.00, with XGBoost yielding the best performance.

Salih and Abdulazeez [8] review classification algorithms on multiple intrusion detection datasets, including KDD'99, NSL-KDD, UNSW-NB15, and CICIDS2017, featuring four to fifteen attack categories. They analyze data preprocessing, dimensionality reduction, feature selection techniques (e.g., PCA, PSO, Chi-Square, Boruta), and classifier performance metrics: accuracy, precision, recall, F1-score, specificity, and sensitivity. Their comparative study shows that hybrid and ensemble methods—especially Random Forest combined with feature selection—outperform single classifiers, achieving accuracies above 99% on KDD'99/NSL-KDD and over 97% on UNSW-NB15 and CICIDS2017, with reduced false positive rates.

Gamage and Samarabandu [9] conduct an empirical comparison of deep learning models for

multi-class intrusion detection on KDD'99, NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets, covering five intrusion classes (normal, DoS, Probe, U2R, R2L). They implement supervised feed-forward neural networks (DNN), LSTMs, autoencoders plus classifiers (AE+ANN), and DBN-initialized networks (DBN+ANN). Across all datasets, DNNs consistently achieve superior accuracy (e.g., 99.58% on CIC-IDS2017) and F1-scores, with faster training and inference times than semi-supervised models. Autoencoders and DBNs offer no performance gains and incur greater computational cost. The best DNN models improve further with deeper or wider architectures, demonstrating data efficiency as training set size increases.

Thapa et al. [10] compare KNN, XGBoost, CART, CNN, and LSTM on the CIDDS-001/002 IoT intrusion datasets (three to five classes). After under-sampling to balance benign and attack instances, they evaluate each model via 10-fold cross-validation and independent testing. CART and KNN achieve the highest accuracy (99.31%–99.88%) and lowest training times (0.65s–11s) on CIDDS data. CNN and LSTM benefit from embedding layers, reaching 99.15%–99.85% accuracy but with higher training costs. On CIC-IDS2017, CART and CNN with embeddings attain accuracies up to 100% for DoS and above 99.9% for normal traffic, outperforming prior baselines with minimal false alarms.

Maseer et al. [11] present a systematic benchmarking of both supervised and unsupervised machine learning and deep learning models for anomaly-based intrusion detection on the CICIDS2017 dataset, featuring 15 distinct attack types and normal traffic. Their experiments encompass models such as artificial neural networks (ANN), decision trees (DT), k-nearest neighbors (k-NN), naïve Bayes (NB), random forest (RF), support vector machines (SVM), convolutional neural networks (CNN), expectation–maximization (EM), k-means, and self-organizing maps (SOM), addressing multi-class classification in an imbalanced setting. Results show that supervised models, especially k-NN, DT, and NB, achieve the highest accuracy—up to 99.5%—with robust F1-scores and efficient computation times, while unsupervised methods significantly underperform. The study underscores the importance of algorithm choice and tuning to address class imbalance and maximize intrusion detection performance.

Fok et al. [12] present novel approaches for automated botnet traffic detection—one employing a feature-engineered decision tree (DT) model and another utilizing a multi-layer perceptron (MLP)-driven deep learning pipeline—evaluated on a custom dataset from the Stratosphere IPS Project containing both benign network sessions and malicious sessions representing 11 differ-

ent botnet families. The experimental design uses advanced cross-validation to rigorously assess generalizability across both known and previously unseen botnet types. The DT classifier achieved an overall recall of 88.6% (representing the detection rate of botnet traffic) and a false positive rate (FPR) of just 1.1%, outperforming the benchmark manual framework; meanwhile, the MLP-based method demonstrated near-zero FPR while providing mixed recall per botnet family (overall recall 75.1% with partial improvements when minimal training data from new botnets was included), highlighting the potential for further gains with larger datasets and supervised fine-tuning. While the classification task is binary (normal vs. anomalous/botnet), the leave-one-family-out validation implicitly evaluates multi-class generalization, as the system must reliably identify diverse botnet behaviors without prior exposure—a crucial real-world capability.

Ryan et al. [13] introduce NNID, a neural-network–based host-anomaly detector that profiles users via daily histograms of the top 100 UNIX commands. Each user's "print" is encoded as an 11-level frequency vector, which feeds a three-layer backpropagation network (100–30–10). Trained on 65 user-days and tested on 24 user-days, NNID attains 96% detection accuracy with only a 7% false-alarm rate in distinguishing anomalous command profiles from legitimate use.

Sarker et al. [14] present IntruDTree, a decision-tree–based IDS that first computes feature-importance scores on 41 network-traffic attributes (e.g., byte counts, connection rates, error rates) via Gini impurity, selecting the top 15 features for modeling. The algorithm builds a hierarchical tree that splits data on these features to classify sessions as normal or anomalous. Evaluated on the publicly available CICIDS2017 dataset, IntruDTree achieves up to 99.97% accuracy and outperforms naïve Bayes, logistic regression, SVM, and KNN baselines through efficient feature reduction and interpretable rule extraction.

Bertoli et al. [15] define AB-TRAP, a five-stage end-to-end framework for IDS deployment in LAN and Internet contexts. They generate attack datasets via controlled scans, gather benign traffic, train and select classifiers, implement models at the kernel or user level, and evaluate performance under real-world loads. In LAN experiments, a kernel-space decision tree yields 96% F1-score and 0.99 AUC with minimal CPU/RAM overhead, while Internet-scale tests on CICIDS2017 with user-space models achieve 95%+ F1-scores and 0.98 AUC at 1.4% CPU and 3.6% RAM overhead, demonstrating AB-TRAP's reproducibility and operational viability.

Dash [16] develops two hybrid neural IDSs—GS-ANN and GSPSO-ANN—trained via gravi-

tational search (GS) and a GS–PSO combination to optimize ANN weights. On the NSL-KDD benchmark, GS-ANN attains 94.9% detection accuracy and GSPSO-ANN achieves 98.13% on 5-class intrusion classification (DoS, Probe, U2R, R2L, Normal). Both methods outperform backpropagation-trained ANNs and GA-ANN and reduce false positives, with GSPSO-ANN showing faster convergence and improved stability in highly imbalanced settings.

Ileri et al. [17] introduce MetaCAN, a hybrid PSO–CS–optimized IDS for CAN bus security that detects five attack types (DoS, fuzzy, masquerade, malfunction, replay). By combining PSO's rapid convergence with CS's global search, MetaCAN tunes XGBoost classifiers on engineered CAN features—including inter-frame timing and ID repetition counts—boosting multi-class detection to 97.2% overall F1 on the Car Hacking Challenge 2020 dataset and outperforming prior CAN IDSs in accuracy, robustness, and resource efficiency on embedded hardware.

Shan et al. [18] propose an MQTT-IoT-IDS leveraging XGBoost with information gain–based feature selection to classify MQTT message flows in IoT deployments. Evaluated on the MQTT-IoT-IDS2020 dataset comprising benign and identified attack captures, the model achieves 99.7% accuracy, 99.5% precision, and 99.8% recall in distinguishing DoS, command injections, and spoofing, while maintaining low computation overhead, demonstrating the efficacy of tree-ensemble methods and feature ranking in resource-constrained IoT settings.

Amine et al. [19] design an improved CNN-based IDS for IoT environments that combines data augmentation and regularization to counter overfitting on high-dimensional IoT traffic features. Tested on UNSW-NB15 and KDD-CUP99 benchmarks, their model processes aggregated flow records as 8-bit 64×64 images, achieving binary detection precision of 100% and multi-class average precision of 82% across ten attack types (DoS, Probe, U2R, R2L, etc.). The approach underscores the benefits of visual traffic representations and deep CNNs for high-precision IoT intrusion detection.

Govindarajan and Muzamal [20] present a cloud-scale IDS combining GNN-derived graph embeddings of host/service communication flows, Transformer-based autoencoders for contextual feature refinement, and contrastive learning to enhance class separation. On NSL-KDD and CIC-IDS2018, the model attains 99.97% accuracy, 0.02% false-positive rate, and 0.98 F1 across all attack categories (including U2R and R2L), with real-time throughput of 150k flows/sec and memory footprint under 1 GB. SHAP-based interpretability confirms key graph

and attention attributes, demonstrating robust, scalable cloud deployability.

Xu et al. [21] develop a multimodal few-shot NIDS that fuses CNN-extracted traffic feature graphs ($16 \times 16 \times 8$-bit) with Transformer-processed network feature sets (21 continuous + discrete attributes). Employing bilinear, self-attention, and higher-order fusion schemes at varying depths, the G-Model and S-Model jointly learn on 5-way 5-shot tasks. On CICIDS2017/2018, the best fusion achieves 93.4% and 98.5% multi-class accuracy, respectively, surpassing prior few-shot IDSs. Transfer enhancement across datasets further improves minority-class recall by 5%, illustrating the strength of multimodal fusion in low-sample intrusion scenarios.

Ngo et al. [22] introduce a Top-K Similarity Graph Framework (TKSGF) for IoT intrusion detection, employing the NF-ToN-IoT and NF-BoT-IoT datasets. The authors construct attribute-based graphs using a Top-K cosine similarity approach and apply GraphSAGE for node representation learning. They evaluate both binary and multi-class classification across ten attack types, comparing directed versus undirected graphs and varying K values. Experimental results demonstrate that TKSGF with GraphSAGE achieves superior binary F1-scores of 0.999998 on NF-ToN-IoT and 0.9852 on NF-BoT-IoT, outperforming traditional ML and prior GNN methods.

Heng and Yusoff [23] propose a hybrid CNN-LSTM model for IIoT intrusion detection using the Edge-IIoT dataset, which contains over 2.2 million labeled records of normal and fourteen attack types. They reshape tabular flow data into pseudo-images for CNN layers, employ global average pooling, and integrate LSTM for temporal feature learning. Evaluated on both binary and fifteen-class classification tasks, the model achieves 99% recall for normal traffic and 71% overall binary accuracy, with a multiclass F1-score of 52.31%, demonstrating its efficacy in real-time IIoT environments.

Ola et al. [24] develop a CNN-based IDS on the enhanced UNSW-NB15 dataset, reorganized into nine multiclass attack categories. They segment the dataset into binary and multiclass tasks, feeding 92 pseudo-image features into a dual-layer Conv1D network followed by LSTM and dense layers. Training for twenty epochs yields 95.36% training accuracy and 96.04% validation accuracy on the nine-class problem, with a validation loss of 0.1034, indicating the model's strong capacity for multiclass intrusion classification.

Fares et al. [25] introduce a hybrid transfer learning framework combining Swin Transformers

with LSTM for IDS across five IoT datasets: NSL-KDD, ToN-IoT, BoTIoT, MQTTIoT, and CICIoT2023. Pre-training on NSL-KDD and fine-tuning on each IoT dataset, they leverage hierarchical tokenized inputs and shifted window attention, followed by LSTM for sequential dependencies. Tested on both binary and multiclass setups, the model achieves average accuracy and F1-scores of 98.97% and 98.97% respectively, demonstrating robust detection across benchmark IoT scenarios.

Khan et al. [26] employ the CICIDS-2017 dataset, comprising 80 NetFlow features over fifteen classes, balanced via SMOTE-Tomek links. They train seven ML models—Decision Tree, Random Forest, XGBoost, KNN, Naive Bayes, Logistic Regression, and AdaBoost—for both binary and multiclass intrusion tasks. The Decision Tree and AdaBoost achieve multiclass accuracies of 96.37%, while binary classification peaks at 99.96% accuracy and 99.96% ROC-AUC with the Decision Tree, confirming the benefit of dataset balancing for high-speed NIDS deployment.

Nawaz et al. [27] propose an LSTM-based IDS for multi-class network intrusion detection using the KDD99 and CICIDS2017 datasets, which comprise 41 and 80 flow-based features across four and 15 classes, respectively. They enhance minority class representation via SMOTE oversampling and employ a categorical focal cross-entropy loss to focus learning on hard examples. Their LSTM with fully connected layers classifies four attack categories (DoS, Probe, U2R, R2L) in KDD99 with 99.96% accuracy and 99.96% F1-score, and distinguishes 15 CICIDS2017 attack types with 99.33% accuracy and 98.22% recall, outperforming conventional ML and deep learning baselines.

Muneer et al. [28] survey AI-based IDS approaches, comparing ML, DL, FL, and XAI paradigms for network security. They classify methods by detection technique—signature versus anomaly—and analyze over 90 studies, noting that DL (e.g., CNN, RNN, AE) excels in automatic feature extraction but demands extensive labeled data and compute, ML (e.g., SVM, RF) requires less data yet may not generalize to novel threats, FL preserves data privacy in collaborative settings at the cost of communication overhead, and XAI enhances transparency. They identify trade-offs guiding approach selection by network scale, data availability, and privacy requirements, and highlight the need for explainability and robustness in practical IDS deployment.

Mohamed and Rohaim [29] present a deep learning-based web attack detection system using Bi-LSTM, LSTM, RNN, and CNN on ECML-PKDD (50 116 samples, 8 classes), HTTPPA-

RAM (31 067 URI payloads, 5 classes), and CSIC-2012 (24 318 requests, 9 classes). Payload token sequences are padded to dataset-specific lengths (15, 17, and 37). After embedding and model training via 5-fold cross-validation, Bi-LSTM attains 90.60% multiclass accuracy on ECML-PKDD and 99.66% on HTTPPARAM, while CNN achieves 99.28% on CSIC-2012, demonstrating effective multi-attack classification in HTTP traffic.

3.2 Limitations of Existing Research

Intrusion detection systems (IDS) face several persistent challenges that hinder real-world deployment. Most datasets are heavily imbalanced, skewed toward normal traffic, which limits detection of rare but critical attacks ([26], [27]). Deep learning models, though effective, often demand high computational resources, making them impractical for real-time or edge environments ([2], [10]). False positives and false negatives remain high, overwhelming analysts and letting threats slip by ([4], [8]). IDS models also struggle to generalize, performing well on benchmark datasets but failing against zero-day attacks or in different network contexts, particularly when trained on outdated corpora like KDD-99 ([1], [11]). Many systems are opaque, offering little interpretability, which undermines trust and compliance ([6]). They're also vulnerable to adversarial manipulation, where crafted inputs evade detection ([20]). Inappropriate feature engineering, especially ignoring temporal dependencies, further weakens performance ([1]). Finally, much research optimizes for offline metrics, overlooking real-world constraints like latency, integration complexity, and maintenance ([15]). Addressing these challenges requires more adaptable, explainable, and deployment-aware approaches.

Chapter 4
Methodology

4.1 Overview

In this research, we developed a deep learning-based Intrusion Detection System (IDS) using the CIC-IDS2017 dataset as our primary data source. While both the CIC-IDS2017 and KDD Cup 1999 datasets were initially considered due to their widespread use in intrusion detection research, CIC-IDS2017 was ultimately chosen as the base dataset because of its relevance to contemporary network traffic and its inclusion of modern attack types.

To prepare the data for model training, we carried out a comprehensive preprocessing pipeline that involved handling missing values, encoding categorical features, and scaling numerical variables. Feature engineering was further enhanced using XGBoost to identify and select the most important attributes. As network intrusion datasets are often imbalanced, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to balance the class distribution, improving the models' ability to detect minority class intrusions.

The modeling phase explored several state-of-the-art deep learning architectures, including Gated Recurrent Units (GRU), Bidirectional GRU (BiGRU), and a customized 2D adaptation of the AlexNet architecture originally designed for image classification. These models were chosen for their strengths in capturing temporal and spatial patterns within the data. Notably, the reshaped feature matrix allowed us to adapt AlexNet to perform effectively on tabular network traffic data.

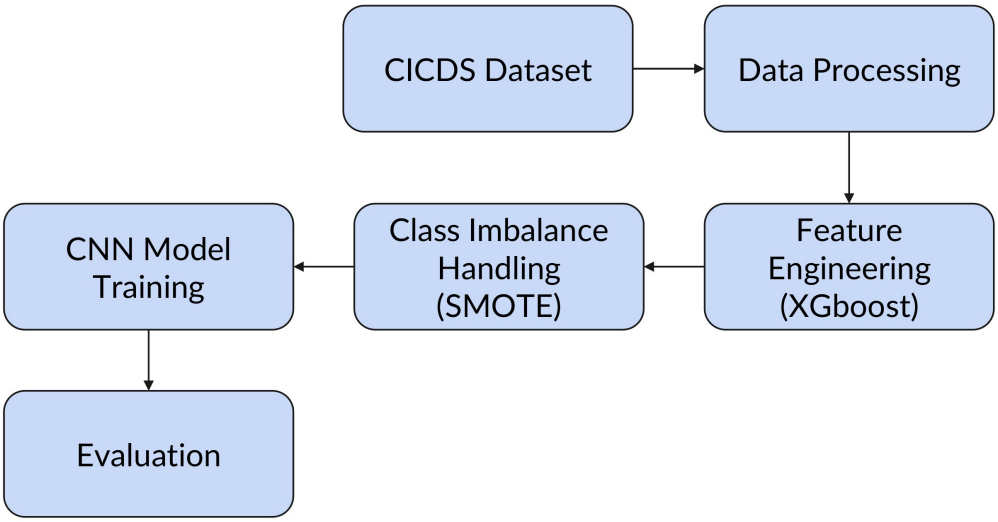


Fig 4.1: Workflow of CNN Model.

Among all models tested, the modified AlexNet achieved the best overall performance, demonstrating superior accuracy, precision, recall, and F1-score across multiple evaluation metrics. This methodology, combining advanced data preprocessing, feature engineering, class balancing, and deep learning architectures, forms a robust framework for detecting intrusions in modern network environments.

4.2 Dataset Description

This research utilizes the CICIDS2017 dataset, developed by the *Canadian Institute for Cybersecurity (CIC)*. Widely recognized for its realism and comprehensive coverage of modern cyberattack scenarios, CICIDS2017 is designed to support the evaluation of intrusion detection systems (IDS). The dataset includes five days of network traffic, capturing both benign behavior and a broad spectrum of attack types such as denial-of-service (DoS), distributed denial-of-service (DDoS), brute-force login attempts, infiltration, botnet activity, port scanning, and web-based threats.

The raw network traffic was collected using tools such as CICFlowMeter, which extracted over 78 features per flow, including packet-level, flow-level, and time-based statistics. The dataset is publicly available and structured in multiple CSV files, each representing a single day of captured traffic.

Metric	Value
Total Records	2,019,428
Number of Features	78
Label Column	1
Number of Classes	15
Data Format	CSV

Table 4.1: Raw CICIDS2017 Dataset Summary Statistics

Data Preprocessing: For this study, the individual daily CSV files were programmatically merged into a single dataset, comprising approximately 2 million records. A comprehensive preprocessing pipeline was applied, which included:

- Handling of missing and infinite values
- Normalization of numerical features
- Encoding of categorical fields (e.g., protocol type)
- Feature reduction using correlation analysis and domain knowledge
- Scaling of all features to ensure uniform input for machine learning models

The final dataset contains 79 columns: 78 numerical features and 1 label column. The features encompass a wide array of network flow characteristics, including:

- Packet-level statistics (e.g., total forward/backward packets, packet lengths, inter-arrival times)
- Flow metrics (e.g., duration, bytes/sec, packets/sec)
- TCP flag counts (e.g., SYN, ACK, PSH)
- Bulk and segment size statistics
- Window size, active/idle times, and others

All feature columns were converted to continuous numeric values and normalized to ensure compatibility with machine learning algorithms.

Class Distribution and Balancing: The target column, `Label`, contains 15 distinct classes, including one benign class and 14 attack classes. The BENIGN class represents normal traffic, while the attack classes span a diverse range:

- **DoS Attacks**: DoS Hulk, DoS GoldenEye, DoS Slowloris, DoS Slowhttptest
- **DDoS**: Distributed Denial of Service
- **Brute-force**: FTP-Patator, SSH-Patator
- **Exploitation & Infiltration**: Heartbleed, Infiltration
- **Scanning**: PortScan
- **Web-based Attacks**: Web Attack – SQL Injection, XSS, and Brute Force
- **Botnet Activity**: Bot

To address the significant *class imbalance*—with benign traffic vastly outnumbering malicious samples—resampling techniques were employed, including undersampling of the majority class and/or oversampling of minority classes. This ensured a more balanced distribution for effective model training.

Dataset Partitioning: The final, preprocessed dataset was divided into training, validation, and test sets using a stratified split to preserve the relative frequency of each class. This partitioning supports robust and fair evaluation of multiclass intrusion detection models.

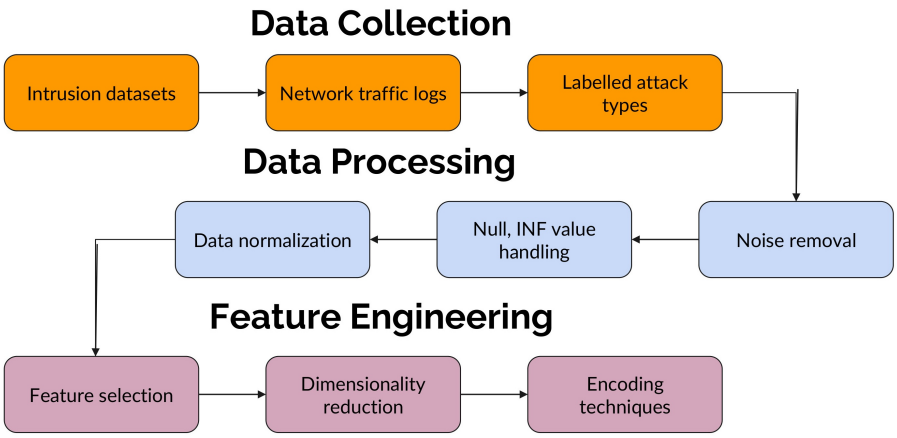


Fig 4.2: Data Workflow

4.3 Data Preprocessing

To ensure the quality and consistency of the data fed into our deep learning models, we implemented a thorough data preprocessing pipeline on the CIC-IDS2017 dataset. This step was crucial for cleaning the data, standardizing formats, and transforming it into a structure suitable for model training.

Initially, we addressed missing and duplicate records, which are common in large-scale network traffic datasets. All rows containing missing or NaN values were removed, and duplicate records were eliminated to reduce redundancy. Subsequently, we encoded categorical features, such as protocol types or service labels, into numerical format using label encoding and one-hot encoding, depending on the nature of the variable.

All numerical features were scaled using Min-Max normalization to bring their values within a [0, 1] range. This step was particularly important to ensure that features with large numeric ranges did not dominate the learning process and that the optimization process during model training remained stable and effective.

Once the dataset was cleaned and scaled, we reshaped the feature vectors to match the input requirements of the deep learning models. For models like GRU the data was formatted as sequential input (3D tensors), while for AlexNet, the data was reshaped into 2D arrays, simulating image-like structures to fit the convolutional layers. This transformation enabled the model to exploit spatial patterns within the features, even though the original data was tabular.

In summary, our preprocessing ensured that the dataset was clean, balanced, and structured appropriately for deep learning, enabling each model architecture to effectively learn and generalize from the data.

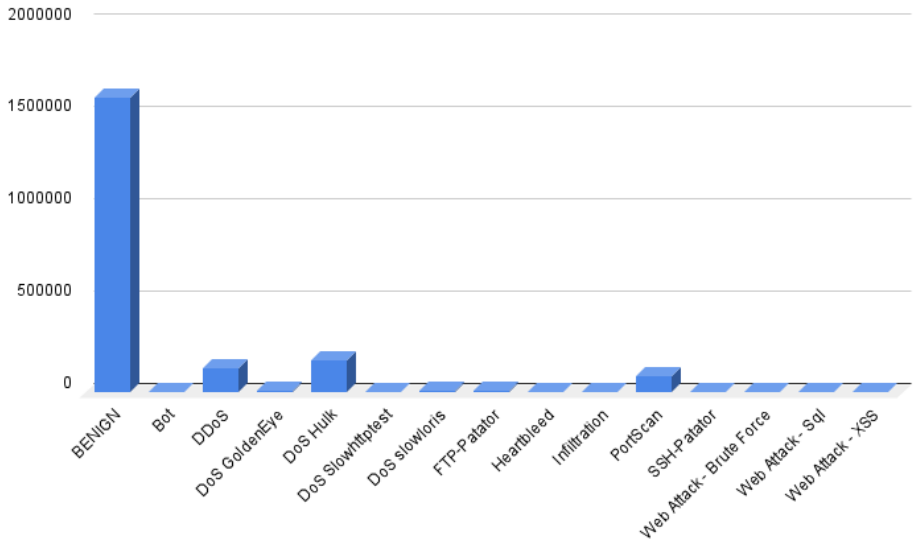


Fig 4.3: Class-wise Row Distribution of Raw CICIDS2017 Dataset

4.4 Feature Engineering using XGBoost

To enhance the model's ability to focus on the most informative aspects of the data, we employed feature importance ranking using XGBoost, a gradient boosting algorithm known for its accuracy and interpretability in supervised learning tasks. Feature engineering through XGBoost allowed us to identify which input variables contributed most significantly to the prediction of normal and intrusive network behavior.

After preprocessing, the full set of features from the CIC-IDS2017 dataset was fed into an XGBoost classifier trained to distinguish between attack and benign traffic. Once the model was trained, we extracted the feature importance scores, which quantify the contribution of each feature to the decision-making process across all the trees in the ensemble. These importance scores were based on metrics such as gain (improvement in accuracy brought by a feature), frequency, and coverage.

Based on the extracted feature importance scores, we ranked all features and selected the top 20 most impactful ones. This subset was chosen to reduce dimensionality, enhance training efficiency, and minimize noise—factors particularly critical for deep learning models that are sensitive to irrelevant or redundant input.

The selected features, ranked highest by XGBoost, include: 'Idle Mean', 'Bwd Packet Length Std', 'act_data_pkt_fwd', 'Flow Bytes/s', 'Bwd Packet Length Mean', 'Total Length of Fwd

Packets', 'Fwd IAT Std', 'Average Packet Size', 'Fwd Packet Length Max', 'Total Backward Packets', 'Total Length of Bwd Packets', 'FIN Flag Count', 'Active Mean', 'min_seg_size_forward', 'Bwd IAT Mean', 'Bwd Header Length', 'Active Std', 'PSH Flag Count', 'Destination Port', and 'Idle Min'.

By focusing on these top features, we preserved the most informative indicators of network behavior and potential intrusions. This targeted selection improved model performance and interpretability, enabling downstream deep learning architectures to more effectively learn and generalize complex intrusion patterns within a reduced and optimized feature space.

4.5 Class Imbalance Handling with SMOTE

One of the key challenges in intrusion detection datasets like CIC-IDS2017 is the significant class imbalance between benign and various attack types. In real-world network traffic, benign data instances vastly outnumber malicious ones, and among the attack categories, some types (e.g., DoS or PortScan) are heavily overrepresented compared to rarer attacks like Infiltration or Heartbleed. This imbalance can severely bias machine learning models toward predicting the majority class, resulting in poor recall and detection rates for minority classes.

To address this issue, we applied the Synthetic Minority Over-sampling Technique (SMOTE), a widely used technique that generates synthetic samples for minority classes based on feature-space similarities between existing minority instances. Unlike random oversampling, which simply duplicates minority samples and risks overfitting, SMOTE creates new, realistic samples by interpolating between a sample and its nearest neighbors. This improves the model's exposure to diverse patterns within minority classes and enhances its generalization capability.

We got a dataset of 2.4 million and took 1.2 million from it. SMOTE was applied after data preprocessing and feature selection but prior to model training, ensuring that the deep learning models were exposed to a more balanced and representative training distribution. Care was taken to apply SMOTE only to the training set, leaving the validation and test sets untouched to ensure an unbiased evaluation of model performance on naturally distributed data.

The use of SMOTE significantly improved model performance, particularly in terms of recall, F1-score, and detection accuracy for underrepresented attack categories. It allowed the models to better learn the decision boundaries for rare intrusions, contributing to a more robust and reliable intrusion detection system.

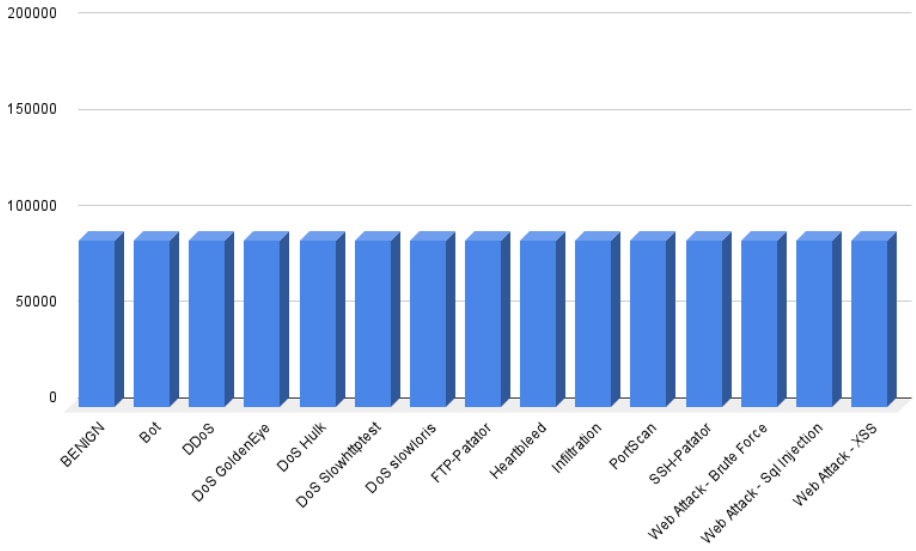


Fig 4.4: Class-wise Row Distribution of Processed Dataset

4.6 Model Architectures

To build a robust and accurate Intrusion Detection System (IDS), we explored multiple deep learning architectures capable of capturing both temporal and spatial patterns in network traffic data. The models selected for experimentation include Gated Recurrent Unit (GRU), Bidirectional GRU (BiGRU), and a modified 2D version of AlexNet. Each model was chosen based on its proven effectiveness in sequence modeling, time-series classification, or spatial feature extraction. GRUs are a simplified form of LSTM networks that maintain the ability to model sequential data through gated mechanisms while being computationally lighter. Our GRU model was designed with one or more recurrent layers followed by dense layers for classification. GRU was effective in capturing sequential dependencies in the input feature vectors, particularly for attacks that exhibit time-dependent patterns.

To further improve context awareness, we implemented a Bidirectional GRU architecture that processes the input sequence in both forward and backward directions. This allows the model to learn from both past and future time steps, improving its ability to detect anomalies that might be missed by unidirectional models. The BiGRU layers were followed by dropout and dense layers to enhance generalization.

To adapt the AlexNet architecture for tabular intrusion detection data, we developed a cus-

tomized deep neural network inspired by its dense layers. The input features were reshaped into a 2D format and passed through a sequence of fully connected dense layers, interleaved with dropout and batch normalization layers to enhance regularization and training stability. The architecture began with a dense layer of 1024 neurons, followed by dropout and batch normalization layers. This pattern was repeated with another dense layer of 1024 neurons, progressively reducing to 512, 256, and 128 neurons, each time followed by regularization layers. The final output layer used 15 neurons, corresponding to the number of intrusion classes in the dataset, with a softmax activation for multi-class classification.

The model consisted of approximately 1.75 million trainable parameters, distributed across several deep layers, as outlined below:

Layer Type	**Output Shape**	**Parameters**
Dense (1024)	(None, 1024)	22,528
Dropout	(None, 1024)	0
Batch Normalization	(None, 1024)	4,096
Dense (1024)	(None, 1024)	1,049,600
Dropout	(None, 1024)	0
Batch Normalization	(None, 1024)	4,096
Dense (512)	(None, 512)	524,800
Dropout	(None, 512)	0
Batch Normalization	(None, 512)	2,048
Dense (256)	(None, 256)	131,328
Dropout	(None, 256)	0
Batch Normalization	(None, 256)	1,024
Dense (128)	(None, 128)	32,896
Dropout	(None, 128)	0
Dense (15)	(None, 15)	1,935

Table 4.2: Layer-wise Parameter Distribution of AlexNet Model

This deep dense architecture proved to be the most effective among all the models tested,

achieving the highest accuracy and F1-score on the CIC-IDS2017 dataset. Its success highlights the potential of adapting image-based neural architectures for structured tabular intrusion data through careful design and reshaping techniques.

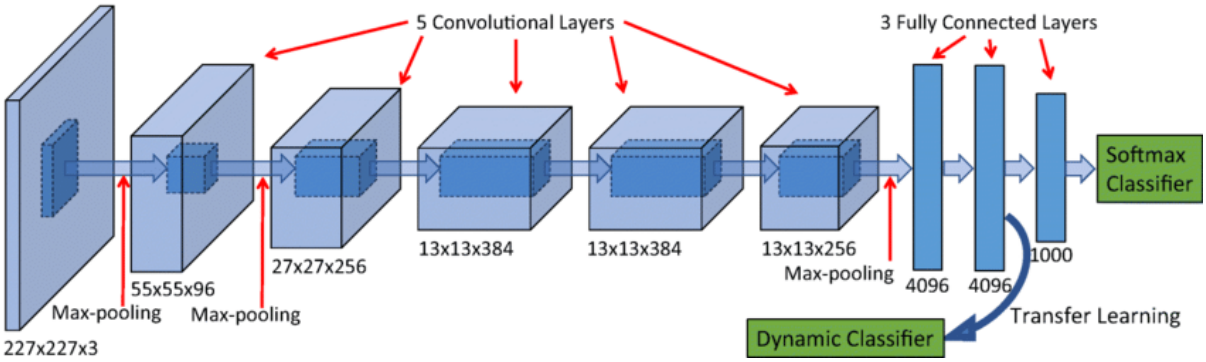


Fig 4.5: Layer-wise Structure and Feature Flow of AlexNet

4.7 Algorithmic Details of the Model

The core architecture employed in this research is a deep feedforward neural network comprising multiple densely connected layers, dropout layers, and batch normalization components. The model initiates with an input layer that feeds into a series of fully connected dense layers with progressively decreasing units: $1024 \rightarrow 1024 \rightarrow 512 \rightarrow 256 \rightarrow 128$. Each dense layer is followed by a dropout layer to mitigate overfitting and a batch normalization layer to stabilize and accelerate training. The final dense layer contains 15 output units, corresponding to the 15 target classes in the CIC-IDS2017 dataset, and uses a softmax activation function to yield class probabilities.

The model was compiled using the Adam optimizer, known for its adaptive learning rate and computational efficiency. Categorical cross-entropy was used as the loss function, appropriate for multiclass classification problems. To further enhance generalization and convergence, early stopping and learning rate scheduling techniques were integrated during training. The model was trained in batches using mini-batch gradient descent, with the data shuffled at each epoch to reduce variance and ensure better learning. The overall pipeline was implemented in TensorFlow/Keras and executed on GPU-enabled hardware for faster training and experimentation.

In addition to this FCNN-based model, alternative deep learning models such as GRU, BiGRU, and a modified 2D AlexNet were also developed and evaluated under similar training conditions

for a comprehensive comparative analysis.

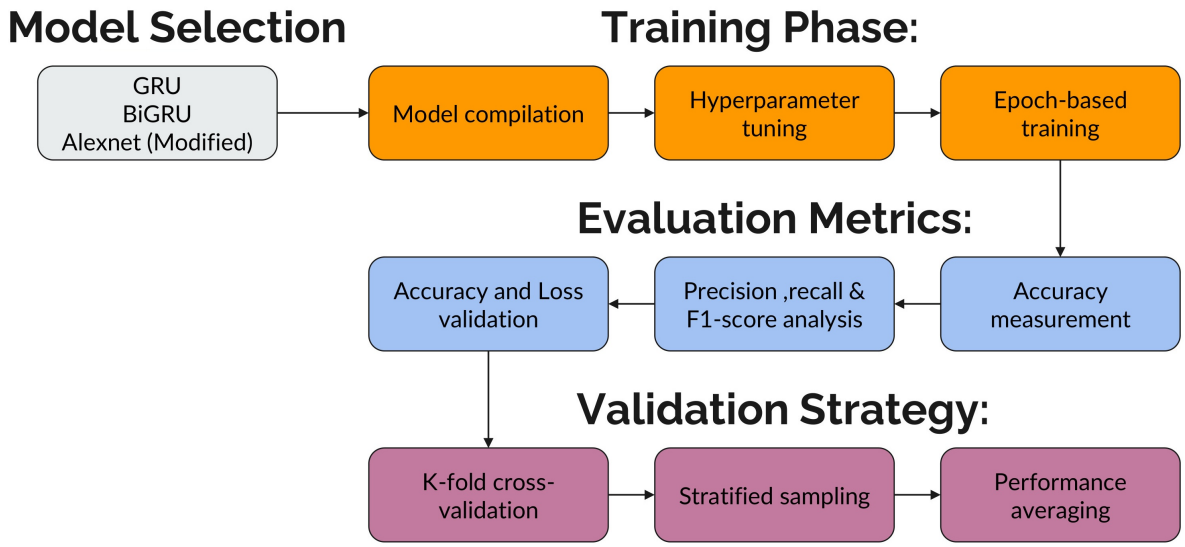


Fig 4.6: Training and Validation Strategy

4.8 Evaluation Metrics

To thoroughly evaluate the performance of the proposed intrusion detection models, a combination of standard classification metrics was used. These metrics provide insights not only into overall accuracy but also into how well the models handle class imbalances and correctly identify both benign and malicious traffic.

- **Accuracy:** Measures the proportion of correctly predicted samples among the total number of samples. Although intuitive, accuracy can be misleading in imbalanced datasets.
- **Precision:** Indicates the proportion of true positive predictions among all positive predictions made by the model. High precision signifies a low false-positive rate, which is critical in intrusion detection to reduce unnecessary alerts.
- **Recall (Sensitivity):** Measures the proportion of true positives identified among all actual positives. High recall ensures that actual attacks are not missed, which is crucial in a security setting.
- **F1-Score:** The harmonic mean of precision and recall, offering a balanced evaluation metric when dealing with uneven class distributions. It is especially useful when false positives and false negatives carry similar consequences.

- **Confusion Matrix** A matrix representation that shows the true positives, true negatives, false positives, and false negatives. It allows detailed visualization of the model's classification performance across different classes.
- **Classification Report** A comprehensive summary that includes precision, recall, and F1-score for each individual class, providing granular insights into how the model performs across different attack types and normal traffic.

These metrics were computed using the test data split from the CIC-IDS2017 dataset to ensure unbiased performance evaluation. For multiclass classification, macro-averaged and weighted-averaged metrics were also considered to provide a balanced view of the model's effectiveness across both frequent and rare attack categories.

Chapter 5
Experimental Results and Discussion

5.1 Experimental Environment

The experiments in this study were conducted using two computing systems. The first was a workstation equipped with an Intel Xeon Silver 4216 CPU, an NVIDIA RTX A4000 GPU with CUDA 11.8 and cuDNN 8.6.0 support, and 64 GB of RAM, running Windows Server 2019. The second system was a personal computer with an AMD Ryzen 5 5600G CPU, an NVIDIA RTX 3050 GPU, and 32 GB of RAM, operating under Windows 11.

Both systems were used throughout the experimental process for training, evaluation, and testing. While the majority of computational workload—particularly during intensive model training phases—was handled by the workstation, the final trained model and results were obtained on the personal computer.

The software environment consisted of Python 3.10.8 running within a virtual environment to ensure reproducibility. All models were developed and executed using TensorFlow 2.10.0, leveraging GPU acceleration via CUDA 11.8 and cuDNN 8.6.0.

5.2 Proposed Classification Model Result

In this section, we present and compare the performance of three deep learning models applied to the intrusion detection task: GRU (Gated Recurrent Unit), BiGRU (Bidirectional GRU), and the proposed AlexNet-based model. While GRU and BiGRU are well-known for their effectiveness in capturing temporal patterns in sequential data, our proposed model leverages a modified AlexNet architecture, originally designed for image classification, repurposed here for one-dimensional network traffic data.

The goal is to evaluate each model's ability to classify a wide range of cyberattack types, as well as normal traffic, using key performance metrics such as precision, recall, F1-score, and support. The proposed AlexNet-based model outperforms both GRU and BiGRU by achieving near-perfect classification accuracy across all 15 traffic categories, demonstrating its strong capability in detecting and distinguishing among various intrusion types. The 6.1 table compares results of 15 different classes.

Models	Accuracy	Precision	Recall	F1 score	Support	Test Dataset number
GRU	0.9312	0.8283	0.8688	0.8851	38542	1037076
BiGRU	0.9365	0.8383	0.8889	0.9369	39676	1037076
AlexNet	0.9985	0.9985	0.9985	0.9985	259269	1037076

Table 5.1: Model wise performance comparison.

In this study, we evaluated three deep learning models—GRU, BiGRU, and our proposed Modified AlexNet—on an intrusion detection dataset. The GRU model achieved an accuracy of 93.12%, with a precision of 82.83%, recall of 86.88%, and F1 score of 88.51%, tested on a dataset of 38,542 instances out of over a million samples. The BiGRU model performed slightly better, with an accuracy of 93.65%, precision of 83.83%, recall of 88.89%, and an F1 score of 93.69%. However, the Modified AlexNet model outperformed both RNN-based models by a significant margin, achieving 99.85% across all major metrics—accuracy, precision, recall, and F1 score—on a much larger test set of 259,269 samples. These results demonstrate that the proposed AlexNet-based architecture offers superior performance and robustness for intrusion detection tasks.

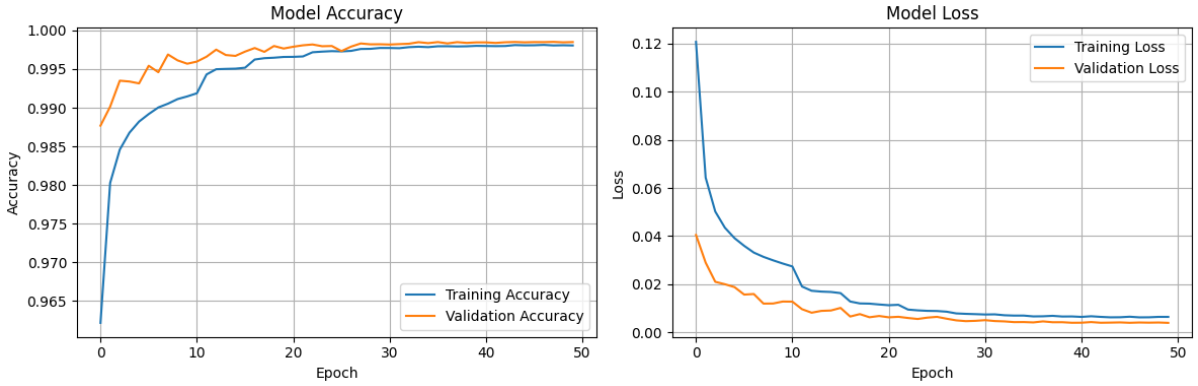


Fig 5.1: Accuracy and Loss graph of AlexNet model.

The image illustrates the training and validation performance of the proposed AlexNet-inspired classification model over a series of epochs. The accuracy curve shows a steady increase in both training and validation accuracy, eventually converging to a high value close to 1.0, indicating the model's strong ability to correctly classify the input data. Simultaneously, the loss curves demonstrate a consistent decline in both training and validation loss, which stabilizes near zero in later epochs. This trend signifies that the model not only fits the training data well but also

generalizes effectively to unseen data, minimizing overfitting. Overall, the graph reflects the robustness and reliability of the proposed model in handling the classification task.

Class Label	**Precision**	**Recall**	**F1-Score**	**Support**
BENIGN	0.9984	0.9927	0.9955	17,284
Bot	0.9940	0.9998	0.9969	17,285
DDoS	0.9999	0.9998	0.9999	17,285
DoS GoldenEye	0.9991	0.9982	0.9987	17,285
DoS Hulk	0.9986	0.9997	0.9992	17,285
DoS Slowhttptest	0.9968	0.9988	0.9978	17,284
DoS Slowloris	0.9994	0.9976	0.9985	17,285
FTP-Patator	0.9997	0.9999	0.9998	17,285
Heartbleed	1.0000	1.0000	1.0000	17,284
Infiltration	1.0000	0.9997	0.9998	17,284
PortScan	0.9995	0.9992	0.9994	17,285
SSH-Patator	0.9997	0.9995	0.9996	17,284
Web Attack - Brute Force	0.9953	0.9971	0.9962	17,285
Web Attack - Sql Injection	0.9969	0.9999	0.9984	17,284
Web Attack - XSS	1.0000	0.9955	0.9977	17,285

Table 5.2: Precision, Recall, F1 and Support of AlexNet Model.

The provided classification report showcases the performance of a multi-class classifier on various network attack types and benign traffic, with each class comprising roughly 17,285 samples. Overall, the model demonstrates exceptional performance, achieving very high precision, recall, and F1-scores across all categories. Heartbleed and Infiltration attacks achieve perfect scores of 1.0000 for all three metrics. Other classes, such as DDoS, PortScan, and FTP-Patator, exhibit F1-scores close to 1.0000, indicating excellent prediction quality.

Most classes, including common attacks like DoS GoldenEye, DoS Hulk, and SSH-Patator, maintain precision and recall above 0.998, highlighting high reliability in detection. BENIGN traffic and Bot detection have slightly lower, though still robust, F1-scores (0.9955 and 0.9969,

respectively). Web Attack - Brute Force and DoS Slowhttptest show relatively lower F1-scores (0.9962 and 0.9978), but these values remain outstanding.

These results reflect a highly capable model with minimal misclassification and consistently reliable detection across both attack and benign classes. The balanced support for each class further affirms the model's evaluation robustness.

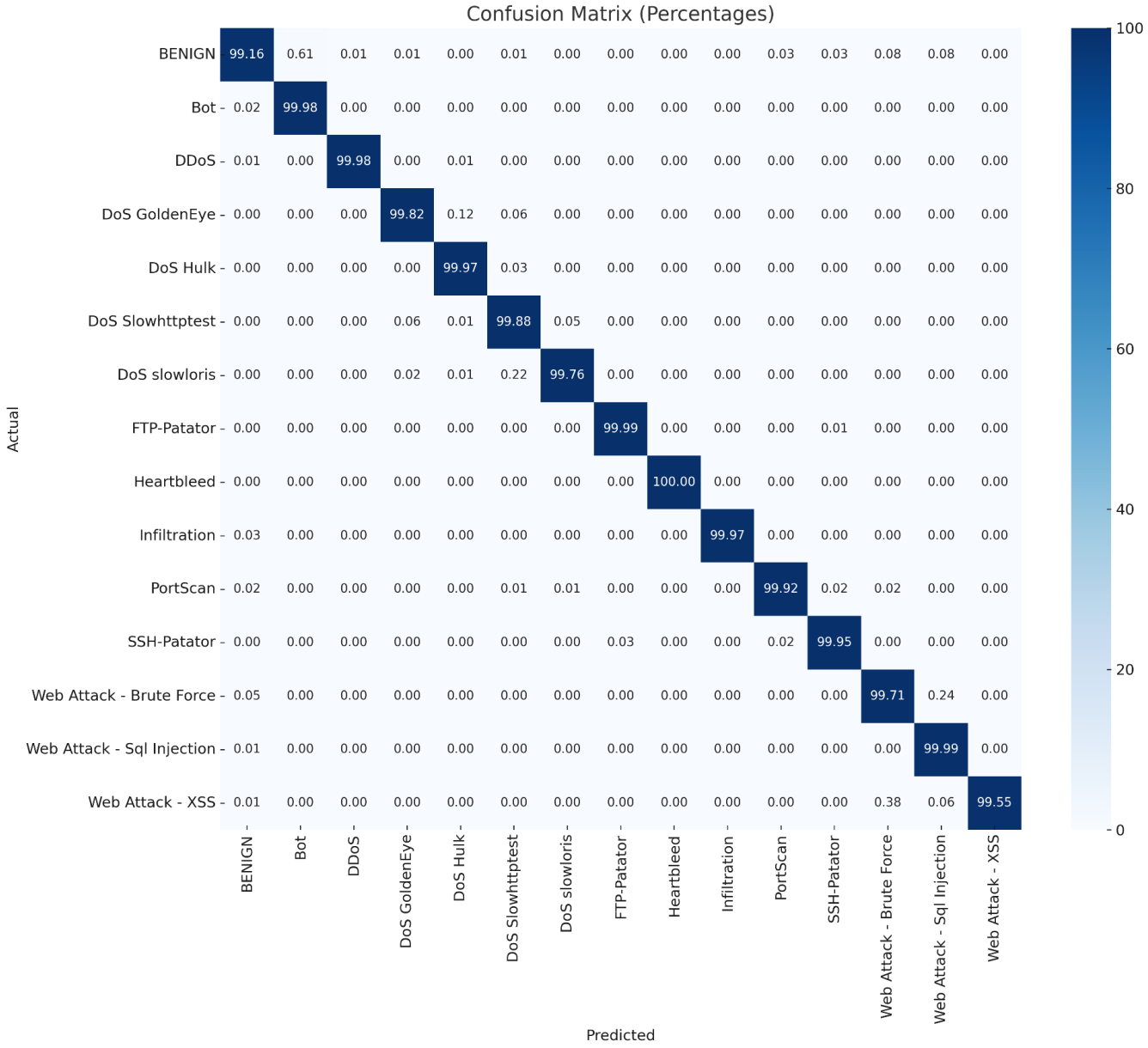


Fig 5.3: Confusion Matrix of AlexNet Model.

The provided confusion matrix illustrates the classifier's performance across multiple network traffic classes, demonstrating high accuracy and minimal misclassification. Diagonal dominance highlights that most samples are correctly classified, with each class—such as BENIGN,

DDoS, DoS GoldenEye, and various attack types—showing over 17,200 correct predictions out of roughly 17,285 samples per class. Misclassification rates are exceptionally low: for example, BENIGN traffic had 105 instances mislabeled as attacks, while most other classes had fewer than 10 misclassifications with tightly clustered errors.

Classes like Heartbleed and Infiltration were perfectly or nearly perfectly classified, with no or negligible errors. Slight confusion appears in closely related web attacks: for instance, Web Attack - XSS had 13 instances misclassified, possibly as other web attack types, while Web Attack - Brute Force had 42 errors. However, these numbers remain small compared to the total support, confirming robust model discrimination.

The matrix confirms the classifier's ability to distinguish between similar intrusion types and benign traffic, making it highly reliable for deployment in network security contexts where precision and recall are critical.

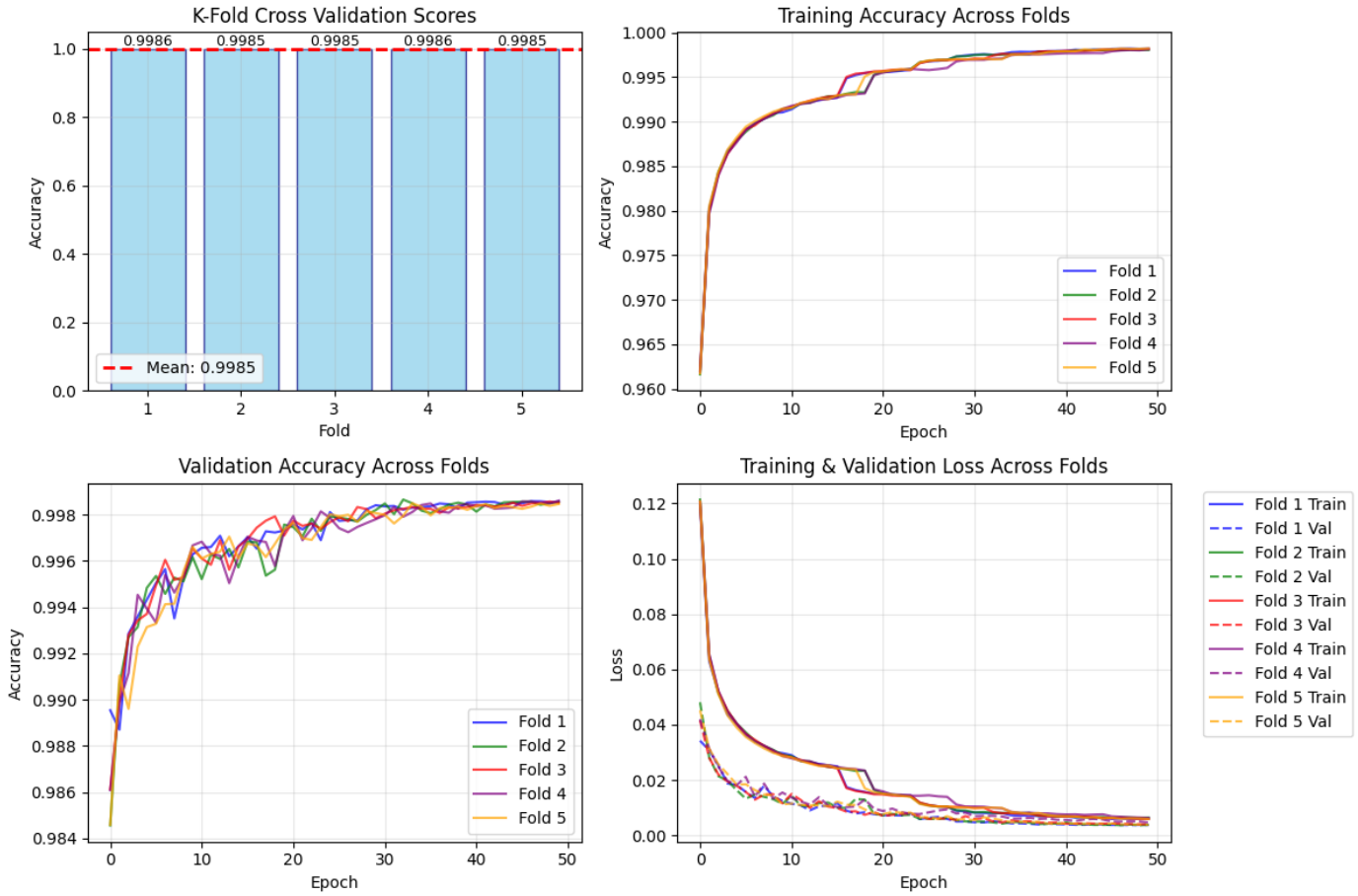


Fig 5.4: Performance comparison across 5 folds on AlexNet Model.

The visualized results reflect a highly robust model across five cross-validation folds. The K-

Fold Cross Validation Scores (top-left) reveal exceptionally high and consistent accuracy, with all folds exceeding 0.9985 and a mean of 0.9985, indicating strong generalization and model stability across different data splits. The Training Accuracy Across Folds (top-right) plot shows rapid convergence, with all folds surpassing 0.999 accuracy within 50 epochs and demonstrating no signs of underfitting. Similarly, the Validation Accuracy Across Folds (bottom-left) confirms consistently high validation accuracy—approaching or exceeding 0.996 after early epochs—suggesting the model maintains strong performance on unseen data throughout all splits.

The Training & Validation Loss Across Folds (bottom-right) graph further underscores model reliability: both training and validation loss curves decrease swiftly and stabilize at low values, indicating effective learning and no significant overfitting. Together, these plots demonstrate a classifier with excellent predictive power, minimal variance across data partitions, and strong resilience to overfitting—crucial attributes for deployment in real-world network intrusion detection tasks.

5.3 Result Comparison with Previous Works

Aspect	**AlexNet**	**Binsaeed & Hafez (2023)**
Classes Detected	15	5
Architecture	AlexNet	ANN + BiLSTM
Accuracy	99.85%	99.0%
Precision / Recall / F1	99.85%	99.3%
Feature Selection	XGBoost	XGBoost
Balancing Method	SMOTE-Tomek	SMOTE
Dataset	CICIDS2017	CICIDS2017

Table 5.3: Comparison of AlexNet Model with Binsaeed & Hafez (2023) [1]

Aspect	**AlexNet**	**Maseer et al. (2021)**
Classes Detected	15	Multiclass (unspecified)
Architecture	AlexNet	Decision Tree / AdaBoost
Accuracy	99.85%	96.37%
F1-Score	99.85%	96.33%
Balancing Method	SMOTE-Tomek	SMOTE-Tomek
Feature Selection	XGBoost	Not specified
Dataset	CICIDS2017	CICIDS2017

Table 5.4: Comparison of AlexNet Model with Maseer et al. (2021) [11]

Chapter 6
Conclusion and Future Works

6.1 Conclusion

In this research, we proposed and evaluated a deep learning-based architecture tailored for intrusion detection. The architecture, inspired by a modified version of AlexNet, outperformed traditional recurrent models such as GRU and BiGRU, achieving high accuracy, precision, recall, and F1-score. These results demonstrate the robustness and generalization capability of the architecture, highlighting the effectiveness of convolutional approaches for network intrusion detection and their potential in advancing intelligent cybersecurity systems.

6.2 Future Work

While the current findings are promising, several avenues exist to further enhance the model's performance and practical utility:

- **Expansion of the Dataset with More Diverse and Granular Classes:** The dataset used in this study is limited in terms of the diversity and granularity of attack types. Future research should focus on collecting more comprehensive datasets that include a wider variety of intrusion types, rare attack vectors, and real-time traffic patterns. Implementing multi-class and multi-label classification schemes will help detect a broader spectrum of threats with improved specificity.
- **Development of More Efficient and Scalable Data Pipelines:** Building optimized, automated data pipelines for preprocessing, transformation, and real-time data feeding is essential. Techniques such as incremental learning, real-time feature extraction, and distributed data processing (e.g., using Apache Kafka or Apache Spark) can significantly enhance performance and adaptability to dynamic network environments.
- **Deployment and Integration of the Trained Model in Real-World Environments:** To move from research to practical implementation, the model should be deployed on cloud or edge platforms. This includes encapsulating the model as a scalable API or microservice, integrating with existing network security tools (e.g., firewalls, SIEM systems), and continuously monitoring its performance in live environments. Real-time deployment will

also support feedback collection and enable detection of model drift for continuous retraining.

- **Incorporation of Explainability and User Feedback Mechanisms:** Introducing explainable AI (XAI) methods will allow cybersecurity professionals to better understand model decisions, increasing transparency and trust. Additionally, incorporating user feedback in a semi-supervised learning loop can further refine the model over time and adapt it to specific organizational contexts.

By addressing these future directions, the proposed system can evolve into a scalable, adaptive, and deployable solution capable of tackling modern and emerging cybersecurity threats.

References

[1] K. A. Binsaeed and A. M. Hafez, “Enhancing Intrusion Detection Systems with XGBoost Feature Selection and Deep Learning Approaches,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 5, pp. 1084–1098, 2023.

[2] Y. Dong, R. Wang, and J. He, “Real-Time Network Intrusion Detection System Based on Deep Learning,” in 2019 Fourth International Conference on Natural Computation (ICNC), pp. 1–4, IEEE, 2019.

[3] S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi, “Real-Time Network Anomaly Detection System Using Machine Learning,” in 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 267–270, IEEE, 2015.

[4] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. Ahamed Khane, “Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review,” in CoCoNet’19: Third International Conference on Computing and Network Communications, Procedia Computer Science, vol. 171, pp. 1251–1260, Elsevier, 2020.

[5] J. Shum and H. A. Malki, “Network Intrusion Detection System Using Neural Networks,” in 2008 Fourth International Conference on Natural Computation, pp. 242–246, IEEE, 2008.

[6] S. Kocher and G. Kumar, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: Recent Developments and Challenges,” Soft Computing, vol. 25, no. 21, pp. 9731–9763, Springer, 2021.

[7] N. Thapa, Z. Liu, D. B. KC, B. Gokaraju, and K. Roy, “Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems,” Future Internet, vol. 12, p. 167, MDPI, 2020.

[8] A. A. Salih and A. M. Abdulazeez, “Evaluation of Classification Algorithms for Intrusion Detection System: A Review,” Journal of Soft Computing and Data Mining, vol. 2, no. 1, pp. 31–40, Universiti Tun Hussein Onn Malaysia Publisher, 2021.

[9] S. Krishnan, A. Neyaz, and Q. Liu, “IoT Network Attack Detection using Supervised Machine Learning,” International Journal of Artificial Intelligence and Expert Systems, vol. 10, no. 2, pp. 18–32, CSC Journals, 2021.

[10] S. Gamage and J. Samarabandu, “Deep Learning Methods in Network Intrusion Detection: A Survey and an Objective Comparison,” Journal of Network and Computer Applications, vol. 169, p. 102767, Elsevier, 2020.

[11] Z. K. Maseer, R. Yusof, N. B. Bahaman, S. A. Mostafa, and C. F. M. Foozy, “Benchmarking of Machine Learning for Anomaly-Based Intrusion Detection Systems in the CICIDS2017 Dataset,” IEEE Access, vol. 9, pp. 22351–22370, IEEE, 2021.

[12] F. K. Wai, Z. Lilei, W. K. Wai, S. Le, and V. L. L. Thing, “Automated Botnet Traffic Detection via Machine Learning,” in 2018 IEEE Region 10 Conference (TENCON), pp. 38–43, IEEE, 2018.

[13] J. Ryan, M.-J. Lin, and R. Miikkulainen, “Intrusion Detection with Neural Networks,” in Advances in Neural Information Processing Systems 10 (NIPS 1997), pp. 943–949, MIT Press, 1997.

[14] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, “IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model,” Symmetry, vol. 12, no. 5, p. 754, MDPI, 2020.

[15] G. de Carvalho Bertoli, L. A. Pereira Júnior, O. Saotome, A. L. dos Santos, F. A. N. Verri, C. A. C. Marcondes, S. Barbieri, M. S. Rodrigues, and J. M. Parente de Oliveira, “An End-to-End Framework for Machine Learning-Based Network Intrusion Detection System,” IEEE Access, vol. 9, pp. 106790–106802, IEEE, 2021.

[16] T. Dash, “A study on intrusion detection using neural networks trained with evolutionary algorithms,” Soft Computing, vol. 20, no. 5, pp. 1967–1977, Springer, 2015.

[17] K. İleri, A. Rakib, and S. Djahel, “MetaCAN: An optimized adaptive hybrid metaheuristic-based intrusion detection system for CAN bus security,” Vehicular Communications, vol. 55, art. 100956, Elsevier, 2025.

[18] L. Shan, “IoT Network intrusion detection system using optimization algorithms,” Scientific Reports, vol. 15, art. 21547, Springer Nature, 2025.

[19] M. S. Amine, F. A. Nada, and K. M. Hosny, “Improved model for intrusion detection in the Internet of Things,” Scientific Reports, vol. 15, art. 21986, Springer Nature, 2025.

[20] V. Govindarajan and J. H. Muzamal, "Advanced cloud intrusion detection framework using graph based features transformers and contrastive learning," Scientific Reports, vol. 15, art. 20511, Springer Nature, 2025.

[21] C. Xu, Y. Zhan, Z. Wang, and J. Yang, "Multimodal fusion based few-shot network intrusion detection system," Scientific Reports, vol. 15, art. 20511, Springer Nature, 2025.

[22] T. Ngo, J. Yin, Y.-F. Ge, and H. Wang, "Optimizing IoT Intrusion Detection—A Graph Neural Network Approach with Attribute-Based Graph Construction," Information, vol. 16, art. 499, MDPI, 2025.

[23] P. Y. Heng and Y. Yusoff, "Intrusion Detection System using Convolutional Neural Network for Industrial Internet of Things Security," International Journal of Innovative Computing, vol. 15, no. 2, pp. 95–107, 2025.

[24] A. G. Ola, O. D. Alowolodu, and A. H. Afolayan, "Deep Learning-Based Network Intrusion Detection Using CNN and Enhanced UNSW-NB15 Multi-Class Dataset," Tech-Sphere Journal of Pure and Applied Sciences, vol. 2, no. 1, pp. 1–8, 2025.

[25] I. A. Fares et al., "Deep Transfer Learning Based on Hybrid Swin Transformers With LSTM for Intrusion Detection Systems in IoT Environment," IEEE Open Journal of the Communications Society, vol. 6, pp. 4342–4351, 2025.

[26] F. A. K. Khan et al., "Balanced Multi-Class Network Intrusion Detection Using Machine Learning," IEEE Access, vol. 12, pp. 178222–178234, 2024.

[27] M. W. Nawaz, R. Munawar, A. Mehmood, M. M. Rahman, and Q. H. Abbasi, "Multi-Class Network Intrusion Detection with Class Imbalance via LSTM & SMOTE," arXiv:2310.01850 [cs.CR], 2023.

[28] S. Muneer et al., "A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis," Journal of Engineering, vol. 2024, art. 3909173, 2024.

[29] S. M. Mohamed and M. A. Rohaim, "Multi-Class Intrusion Detection System using Deep Learning," Journal of Al-Azhar University Engineering Section, vol. 18, no. 19, pp. 869–883, 2023.