DeepDefend: Optimized Multi-Model Approach for Network Intrusion Detection Using Deep Learning and IoT Security Enhancement

MSc Research Project
Artificial Intelligence

Shahna Shahul Hameed
Student ID: x22235094

School of Computing
National College of Ireland

Supervisor: Sheresh Zahoor

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shahna Shahul Hameed
Student ID:	x22235094
Programme:	Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Sheresh Zahoor
Submission Due Date:	12/12/2024
Project Title:	DeepDefend: Optimized Multi-Model Approach for Network Intrusion Detection Using Deep Learning and IoT Security Enhancement
Word Count:	7837
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shahna Shahul Hameed
Date:	23rd January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DeepDefend: Optimized Multi-Model Approach for Network Intrusion Detection Using Deep Learning and IoT Security Enhancement

Shahna Shahul Hameed
x22235094

Abstract

Network intrusion detection has become an important component of current cybersecurity techniques due to the growing frequency of cyberattacks in both conventional and Internet of Things-based network environments. However, current intrusion detection systems (IDS) frequently have high false-positive rates due to limitations in accuracy, adaptability, and the capacity to detect minority attack classes. In order to overcome these obstacles, this study suggests a new hybrid framework that combines advanced deep learning (DL) models with conventional machine learning (ML) techniques. The suggested method ensures strong generalization and adaptability across various network environments by utilizing a variety of datasets, such as UNSW-NB15, NSL-KDD, Cyber Intrusion and ToN IoT. The Perceptual Pigeon Galvanized Optimization (PPGO) method which is especially used to optimize Long Short-Term Memory (LSTM) models for improved IoT security, lies at the core of the suggested solution. In order to achieve balanced performance across a variety of attack vectors, this structure places to strong focus on lowering false-positives rates while also greatly increasing the detection accuracy of minority attack classes. Significant improvements in detection accuracy of upto 98.5% on UNSW-NB15 and 97.8% on NSK-KDD datasets, false-positive reduction, and flexibility to different network configurations are shown by exploratory results from experiments. The results help in the creation of a scalable and robust intrusion detection system that addresses the constantly evolving and complex nature of modern cybersecurity risks. This study has the ability to greatly improve network security across business and IoT-based systems by offering an integrated and efficient solution, paving the way to more secure digital ecosystems.

1 Introduction

Effective Network Intrusion Detection Systems (NIDS) that can defend both conventional and Internet of Things (IoT) networks are vital given the increasing frequency and complexity of intrusions. Although they work well against known threats, traditional signature-based techniques frequently fail to detect novel attacks, such as zero-day exploits. This constraint has encouraged research into more advanced detecting methods. With the goal to improve intrusion detection capabilities, recent research of Halbouni et al. (2022) has looked into hybrid deep learning models, such as CNN-LSTM architectures. However, these these approaches have trouble in efficiently detecting minority attack classes and adapting to the heterogeneous nature of IoT environments. Implementing efficient intrusion detection systems is made more challenging by the extra challenges brought about by the integration of IoT devices into network infrastructures, such as resource limitations and a variety of communication protocols. The need for expandable and adaptable NIDS that can function well in these many situations has been demonstrated by Elrawy et al. (2018)'s research. Additionally, for the reason to maintain high detection accuracy, systems that can learn and adapt are required due to the dynamic nature of cyber threats. The following research questions are intended to be addressed by this study:

1. How can a multi-model approach, combining traditional ML and advanced DL models improve the accuracy of detecting network intrusions across diverse datasets specifically in terms of minimizing false positives and enhancing detection rates for minority attack classes? Additionally, how can adaptability be measures through model performance across various network environments like enterprise networks and IoT systems?

2. What is the role of the PPGO algorithm in improving the performance of the LSTM models for intrusion detection in IoT environments?

To address these questions the study looks into the following objectives:

1. Create a CNN-LSTM hybrid framework that combines temporal and spatial feature extraction.

2. Use optimization techniques like PPGO to improve LSTM performance on datasets unique to the Internet of Things.

3. Examine the suggested system with a focus on minority attack class detection using a variety of datasets, such as UNSW-NB15, NSL-KDD, and ToN IoT.

4.Develop and evaluate a real-time intrusion detection system to verify the framework's applicability.

The research has contributed to the field by creating a hybrid NIDS framework that merges CNNs, PPGO optimization and LSTMs. The aim of the system is to improve detection rates for minority attack classes, decrease false positives and shift to the unique challenges of the IoT networks. This study advanced the effectiveness and flexibility of intrusion detection systems by integrating optimization algorithms with DL models. The report is well structured to give a comprehensive overview of developing and evaluating the proposed system. Section 2 review the literature on NIDS and identifies the current challenges and problems. Section 3 gives a clear detailing on the methodology that includes data gathering, preprocessing, modeling and evaluation. Section 4 discusses the design specification that consists of system architecture, feature engineering, and hybrid modeling framework. Section 5 describes the implementation of the proposed solution while Section 6 analyzes the performance of the system and notes its findings. Final part is Section 7 which summarizes the conclusions and suggests meaningful directions for the future research to be carried out.

2 Related Work

The idea of network intrusion detection has undergone significant change over time due to the growing complexity and frequency of cyberattacks. Traditional methods for detecting intrusions mostly relied on signature-based methods that compared incoming network traffic to a database of known attack patterns.

2.1 Network Intrusion Detection Systems (NIDS)

A vital change from traditional, rule-based methods to more dynamic and flexible models was brought because of the introduction of machine learning (ML) to network intrusion detection systems (NIDS). The application of machine learning techniques that including decision trees and genetic algorithms to intrusion detection was originally shown by Sinclair et al. (1999). Their study laid the foundation for future studies into adaptive anomaly detection algorithms and demonstrated significant progress in the identification of developed attack patterns. In spite of their advancements, early machine learning models were limited by their dependence on predetermined features and their incapacity to process massive, real-world datasets. The foundation of NIDS was made possible by the introduction of statistical anomaly detection techniques by Mukherjee et al. (1994). These systems were efficient in detecting known threats, but because they relied on static databases, they were severely limited in their ability to handle new attacks such as zero-day vulnerabilities.It was highlighted that it could be used to implement the foundation of ML applications in NIDS and the change to deep learning for more enhanced expandability and detection accuracy.

Signature-based solutions frequently produced massive false-positive rates because of their inability to handle the dynamic nature of current network traffic. As an alternative, anomaly-based techniques were developed with the goal of recognizing deviations from typical traffic patterns. While these methods improved detection for unknown threats, they often misclassified benign anomalies as malicious by showing the need for more advanced solutions. By allowing automated feature extraction and pattern identification, machine learning (ML) indicated an abrupt change in intrusion detection. Due to their consistency in classifying network traffic as either benign or malicious algorithms such as Decision Trees (DT), Random Forest (RF), and Support Vector Machines (SVM) gained popularity. Using structured datasets like NSL-KDD, Zaman and Lung (2018) demonstrated how well ML classifiers identify network intrusions with excellent accuracy. However ML-based methods require an extensive amount of feature engineering, which was laborious and susceptible to human error. Furthermore, they performed poorly in multi-class environments, particularly when dealing with unbalanced datasets that included under representations of certain attack types.

Researchers began to explore at ensemble learning techniques, that combine many machine learning models to increase classification accuracy and generalizability in an attempt to overcome these limitations. A recurring problem in intrusion detection is the identification of minority attack classes for which Ncir et al. (2024) improved by using ensemble approaches. Despite these developments, traditional machine learning techniques were still incapable to effectively handle high-dimensional and unstructured data which limited their use in practical situations. By automating feature extraction and learning hierarchical patterns in raw data, deep learning (DL) approaches were able to fill this gap and resolve these difficulties. By creating models that could analyze enormous amounts of network traffic with minimal assistance from humans, deep learning revolutionized network intrusion detection. Convolutional neural networks, or CNNs, have become an effective technique for obtaining spatial characteristics from network data. Therefore, they are especially useful for analyzing packet headers and traffic patterns that are dependent on flow. The base research paper used CNNs' capabilities by combining CNNs with Bidirectional Long Short-Term Memory (BiLSTM) networks, which are great at identifying temporal patterns in sequential data. This hybrid method minimized false positives and improved detection accuracy. CNNs and LSTMs were integrated in similar works by Chawla et al. (2019) and Sun et al. (2020) to create effective intrusion detection models. These models performed effectively on controlled datasets, however they were not sufficiently versatile to function in a variety of network environments, which is a major gap that is addressed in this study.

Intrusion detection has become more difficult as a result of the growth of IoT devices, which has brought about issues including heterogeneity, limited processing resources, and high-volume, low-latency data. Attack types like R2L (Remote-to-Local) and U2R (User-to-Root) are apparently lacking in the disproportionate datasets that IoT networks frequently generate. Due to this imbalance, traditional intrusion detection systems have trouble detecting minority attack classes. In their research on IoT-specific datasets like ToN_IoT, Ashiku and Dagli (2021) shown those challenges and the importance of customized models. These problems have been demonstrated to be accessible to optimization techniques, such as the Perceptual Pigeon Galvanized Optimization (PPGO) proposed by Shitharth et al. (2022). Real-time intrusion detection in scenarios with limited resources is made possible by PPGO, which improves feature selection and lowers computing cost. To provide reliable performance on datasets specific to the IoT, this research uses PPGO to improve LSTM models. The drawbacks of standalone ML or DL models have been addressed by hybrid techniques that combine many architectures. The CNN-BiLSTM-Attention model proposed in the base paper demonstrates this trend by integrating spatial and temporal feature extraction with an attention mechanism that prioritizes relevant features. Attention mechanisms have proven effective in reducing false positives by focusing on critical data points, as demonstrated in studies by ul Haq Qazi et al. (2022) and Ncir et al. (2024). However, the generalizability of these methods was limited since they frequently lacked comprehensive evaluations across various datasets. In order to ensure adaptability and scalability, this study addresses this gap by evaluating the hybrid model on datasets such as UNSW-NB15, ToN_IoT, and Cyber Intrusion.

2.2 Hybrid Approaches in Intrusion Detection

A great example of a hybrid model that combines Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks, and an attention mechanism is the foundation research paper. In order to learn hierarchical patterns from network traffic data, CNNs are used for spatial feature extraction. This technique is very useful for identifying localized anomalies within packets. BiLSTM networks, on the contrary, are exceptional at identifying temporal connections in sequential data which makes them important for examining patterns that change over time. By giving priority to the most relevant features, an attention mechanism improves the model's performance by lowering noise and increasing classification accuracy. The effectiveness of hybrid models in intrusion detection has been shown in numerous research. A CNN-RNN hybrid model, for example, was proposed by ul Haq Qazi et al. (2022) and shown significant accuracy gains on the CICIDS-2018 dataset. Their method made use of RNNs for sequential pattern recognition and CNNs for the extraction of spatial features. However, there were computational issues with their model, particularly when handling huge datasets that had high-dimensional features. The advantages of merging spatial and temporal modeling were also highlighted by Sun et al. (2020) that researched the integration of CNNs and LSTMs for intrusion detection. Although these models performed good on controlled datasets, it was unknown how well they would adapt to IoT environments. In order to ensure scalability and efficiency in IoT networks, this study expands on these findings by optimizing hybrid models using the Perceptual Pigeon Galvanized Optimization (PPGO) algorithm.

The role of attention mechanisms cannot be overestimated in hybrid models. To enhance model interpretability and performance, attention mechanisms which have been widely used in natural language processing (NLP) are now being employed in intrusion detection. Similar to the base paper, Chawla et al. (2019) also showed the efficacy of attention mechanisms in their CNN/RNN hybrid model for host-based intrusion detection. However, their model's application in resource-constrained scenarios like IoT networks was limited by its computational demands and lack of optimization. This study overcomes these constraints by implementing an attention mechanism that is appropriate for IoT-specific challenges. In intrusion detection, hybrid model optimization is a relatively new technique. To improve feature selection and model performance, optimization approaches like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) have been used. The Perceptual Pigeon Galvanized Optimization (PPGO) approach was presented by Shitharth et al. (2022) and showed outstanding results in managing unbalanced datasets and enhancing classification accuracy. In contrast to conventional optimization techniques, PPGO reduces computing cost while concentrating on minority class detection. This study optimizes LSTM models using PPGO, assuring strong performance on IoT-specific datasets such as ToN_IoT. This Dai et al. (2024) study's integration of PPGO with CNN-BiLSTM-Attention models guarantees the model's scalability and efficiency while also improving detection accuracy. The ability of machine learning (ML) techniques, including Support Vector Machines (SVM) and extreme learning machines, to identify unusual patterns in network traffic was shown by Lee et al. (2017). The flexibility of incremental learning model that dynamically change as new data is added which was highlighted in their study. This adaptability is especially important in environments like Internet of Things networks where attack patterns are constantly shifting.

The major part of the research is focused on improving detection accuracy while ignoring the computational problems involved in implementing these models in practical settings. For example, Sun et al. (2020) and ul Haq Qazi et al. (2022) did not address the scalability of their models to large-scale or heterogeneous networks, while achieving great accuracy in controlled testing. Similarly, though the base research presented a strong hybrid model, it was only evaluated on one dataset, which limited its applicability to a variety of network environments.

2.3 Challenges of Imbalanced Data and Minority Class Detection

The issue of imbalanced datasets in which some attack types are significantly lacking is one of the most persistent challenges in IDS. Minority classes, such User-to-Root (U2R) and Remote-to-Local (R2L) attacks, frequently take up a small portion of the entire dataset which biases training and lowers detection rates. In their seminal work on statistical anomaly detection, Mukherjee et al. (1994) recognized the drawbacks of their techniques when dealing with datasets that are imbalanced. Similarly, Zaman and Lung (2018) found that when minority classes were used, classic machine learning classifiers such as SVM and Random Forest (RF) performed poorly. The necessity for models that prioritize minority attack detection has been demonstrated by their evaluation on the NSL-KDD dataset, which showed good accuracy for majority classes but poor recall for R2L and U2R attacks. The base research paper suggested resolving this problem by implementing Equalization Loss v2 (EQL v2) which dynamically modifies the loss function to concentrate much on minority classes during training. By using this method, the detection rates for minority classes were greatly improved and all categories showed balanced performance. Although, the generalizability of the model was restricted due to its dependence on a particular dataset. In this study we integrated EQL with PPGO algorithm, in an effort to further improve EQL's ability to manage imbalanced datasets specifically in IoT situations where minority attacks represent a significant threat.

The failure of current research to adapt to different datasets is another significant drawback. For model evaluation, the majority of research, including those by Sun et al. (2020) and Ncir et al. (2024), concentrate on a single dataset, like NSL-KDD or CICIDS-2018. Although these datasets offer useful comparisons, the models' application in real-world situations with dynamic and heterogeneous network environments is limited by their singular focus. For instance, ul Haq Qazi et al. (2022) used a CNN-LSTM hybrid model to achieve high detection accuracy on CICIDS-2018, still they did not evaluate its performance on datasets with other features, like UNSW-NB15 or ToN_IoT. The difficulty of identifying minority attack classes are further compound in IoT environments as the data characteristics are very different from those of traditional networks. IoT datasets like ToN_IoT frequently include sparse representations of specific attack types, which might make it more difficult for traditional ML and DL models to generalize efficiently. DNNs were investigated by Ashiku and Dagli (2021) for IoT-specific intrusion detection, for which they gained high precision for majority classes but recall issues for minority classes. This demonstrates that models that are good at identifying minority attacks are also necessary in additional to better performance on majority classes.

Machine learning (ML) and Deep Learning (DL) techniques have been used extensively in recent network intrusion detection system (NIDS) advancements to deal with the dynamic and constantly evolving nature of cyber threats. ML and DL approaches were systematically taxonomized by Ahmad et al. (2021) with a focus on their use in network-based intrusion detection. The technique used in this thesis is consistent with the primary focus on the feature engineering and data preprocessing as described by Jadhav et al. (2022) notably the comprehensive preprocessing methods carried out for datasets such as UNSW-NB15 and ToN_IoT. Furthermore, their work shows the importance of processing data in an imbalanced way which is handled in this study by prioritizing minority attack classes using PPGO. Moreover, the third paper's investigation into hybrid techniques in machine learning provides a solid case for combining many algorithms for achieving better results. This work gave a strong foundation for the multi-model strategy used in the present study by showcasing the possibility of integrating various classifiers to increase detection rates and decrease false positives. NIDS capabilities have been greatly improved by machine learning (ML), although some restrictions still exist. In their fundamental assessment of the use of machine learning (ML) in NIDS, Sommer and Paxson (2010) pointed out issues include high false-positive rates, reliance on labeled data and issues detecting complex or hidden attacks. Their research also raised concerns about the practical use of ML-based models in everyday situations, where major obstacles arise because of the dynamic and evolving nature of network traffic. These challenges emphasize how crucial it is to create scalable and adaptable NIDS frameworks that can reliably generalize across a variety of datasets and attack types.

In summary, with section 1 highlighting the development of intrusion detection from conventional statistical methods to advanced machine learning and deep learning techniques the literature review points out the advancements and constraints in network intrusion detection systems (NIDS). To overcome problems like unbalanced datasets and computational limitations, section 2 justifies the integration of hybrid techinques including CNN-BiLSTM-Attention models and their optimization using algorithms like PPGO. While highlighting the importance of multi-dataset evaluation, section 3 critically assesses the weaknesses in the current literature such as single dataset dependency, poor minority attack detection and lack of adaptability. Collectively, these parts underline the goals of this study and demonstrate how critical this work is to developing scalable and reliable NIDS for both WNS and IoT environments.

3 Methodology

This methodology illustrates the clear and systematic method which is taken for development and analyzation in the proposed study of Network Intrusion Detection System (NIDS) as shown in Figure 1. It includes five key processes such as data selection, data preprocessing, transformation, modeling, and evaluation and results. Each and every stage describes the steps and techniques that is used to achieve a strong and adaptable solution in addressing network security obstacles.

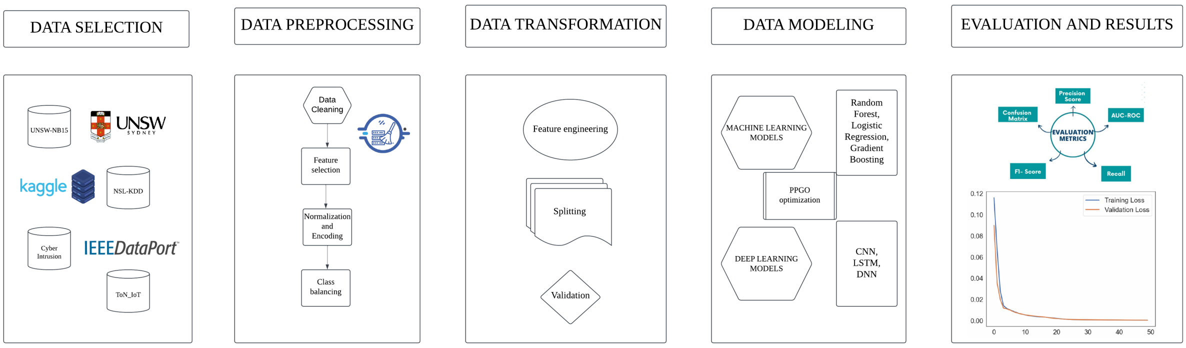


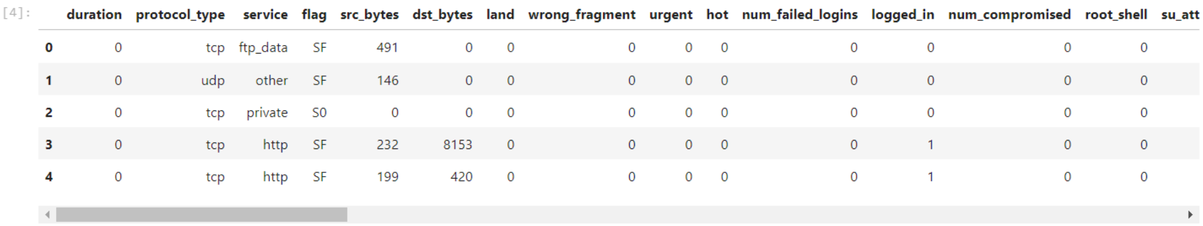
Figure 1: Research Methodology for Intrusion Detection System

3.1 Data Selection

Data Selection being the first phase was essential to assure the system's flexibility and generalizability. Four datasets were gathered and used for this research. Each datasets were selected for its applicability for tasks that involves intrusion detection. The UNSW-NB15[1] dataset has been generated in a controlled tool called IXIA PerfectStorm, that consists of 49 features along with the class label. This dataset has a total of 25,40,044 records that are stored in four CSV files, which containing each training set of 1,75,341 records and testing sets of 82,332 records. This dataset captures both real and synthetic network traffics. This dataset has been widely used in many intrusion detection related researches to assess the usage of NIDS methodologies as Babu et al. (2023). It is a good fit for multiclass classification tasks because of its variety of attack types that includes Shellcode, Denial-of-Service, and Worms. By addressing data redundancy and class imbalance, the NSL-KDD[2] dataset which is a modified version of the KDD'99 dataset offers a

[1] https://research.unsw.edu.au/projects/unsw-nb15-dataset
[2] https://www.kaggle.com/datasets/hassan06/nslkdd

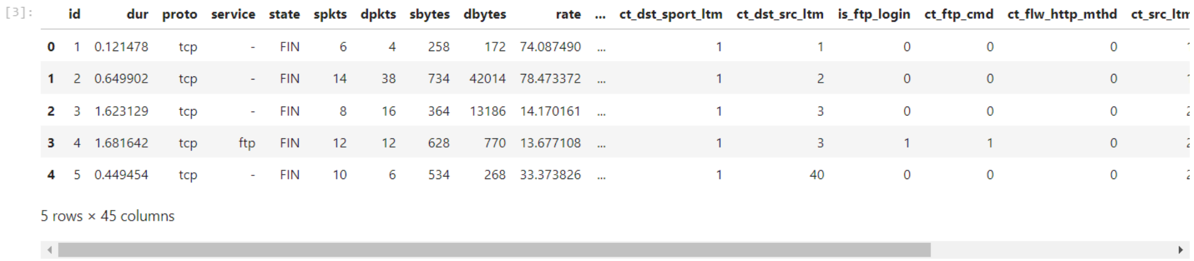
balanced distribution of attack classes including DoS, Probe, Remote to Local (R2L), and User to Root (U2R), which was also used by Dai et al. (2024). Similar to UNSW-NB15, the NSL-KDD dataset also handles the limitations of its predecessor KD'99 by giving a more balanced and filtered dataset that has been active in evaluating machine learning-based IDS as per Musa et al. (2020). With the objective to replicate real-time traffic patterns in the military environment, the Cyber Intrusion[3] dataset provides raw TCP/IP traffic data that has been divided into normal and attack classes. Lastly, telemetry data from gadgets like GPS trackers, thermostats, and Modbus controllers is included in the ToN_IoT[4] dataset that was developed especially for IoT security issues. This dataset is a helpful tool for evaluating the adaptability of the proposed system due of its range of features. It presents certain challenges with regard to feature scaling and constraints on resources. For evaluating the scalability and versatility of NIDS in IoT scenarios, the ToN_IoT dataset—which focuses on telemetry data from IoT devices—is essential as per Khraisat and Alazab (2021). When combined, these datasets offer a thorough and practical basis for creating an intrusion detection system that can handle the obstacles presented by both conventional and Internet of Things-based networks. The below given figures in 2 and 3 shows the head of the basic datasets used throughout the study.



[4]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_att
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	
1	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	
2	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	
3	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	
4	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	

Figure 2: Sample rows from the NSL-KDD dataset that shows feature diversity and class distribution for intrusion detection tasks



[3]:

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.087490	...	1	1	0	0	0	1
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	...	1	2	0	0	0	1
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	...	1	3	0	0	0	2
3	4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	...	1	3	1	1	0	2
4	5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	...	1	40	0	0	0	2

5 rows × 45 columns

Figure 3: Sample rows from the UNSW-NB15 dataset that highlights its mixed real and synthetic traffic patterns

3.2 Data Preprocessing

In data preprocessing stage, raw datasets were converted into a format appropriate for ML and Dl algorithms. The initial phase was data cleaning that had to handle with duplicates, missing values, and inconsistencies or discrepancies. Mean or median values were used to derive numerical features such as Packet size and duration of flow. Meanwhile,

[3] https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection

[4] https://research.unsw.edu.au/projects/toniot-datasets

the mode was employed to substitute categorical data like protocol type. For supporting data integrity, duplicate entries that were common in the Cyber Intrusion dataset were identified and eliminated using hashing techniques as stated in the research by Halimaa A. and Sundarakantham (2019). Consequently, features that were irrelevant or redundant were removed using feature selection. The Random Forest models' feature importance rankings and correlation analysis were integrated. Due to high correlation, for instance - attributes like destination ports and source were excluded while key features like flow duration, protocol type and packet size were preserved for their predictive applicability. Min-Max scaling was utilized to normalize continuous features placing their values within [0,1] range. This process ensured that specifically in gradient-based models like CNNs and LSTMs, features with larger numerical ranges were not prioritized during training. One-hot encoding was employed to encode categorical attributes that is protocol type and service by transforming each category into a binary vector. The Synthetic Minority Oversampling Technique (SMOTE) was used to eliminate class imbalance, a major issue in the NSL-KDD and ToN_IoT datasets as mentioned in Viboonsang and Kosolsombat (2024). To equalize the datasets and allow the models to learn from minority classes, SMOTE produced synthetic examples for minority classes which includes U2R and Shellcode. The below given figure 4 illustrates category imbalance that strengthens the argument for enhancing data balancing techniques and also to improve model adaptability.

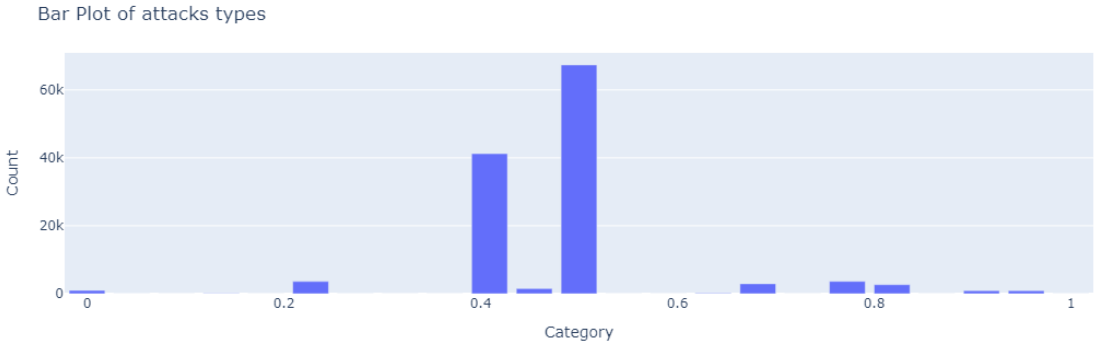


Figure 4: Bar plot showing distribution of attack types in IoT based datasets

3.3 Data Transformation

After preprocessing, data transformation involves modifying the datasets according to specific modeling needs. To improve the datasets' prediction capacity, feature engineering was applied. For representing the temporal dependencies that are important for sequential models as LSTMs, time-series attributes such as average packet arrival intervals and traffic flow durations were computed for the ToN_IoT dataset. Derived features like traffic flow direction and aggregated statics were also added to refine the model's ability to recognize intricate patterns in network traffic. The datasets were split into 70% for training and 30% for testing subsets, that ensured that the testing set consisted of unseen data for unbiased evaluation. Stratified K-Fold Cross Validation (K=10) was exploited for further validation of the models. This method allowed for reliable and robust performance evaluations by maintaining the original class distribution across all folds.

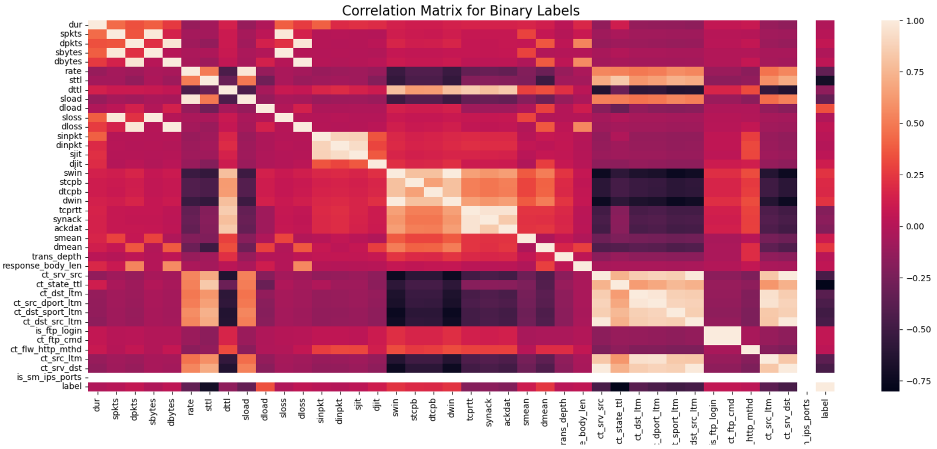


Figure 5: Correlation Matrix for Binary classification in UNSW-NB15 dataset for identifying and eliminating redundant features for enhancement in model performance

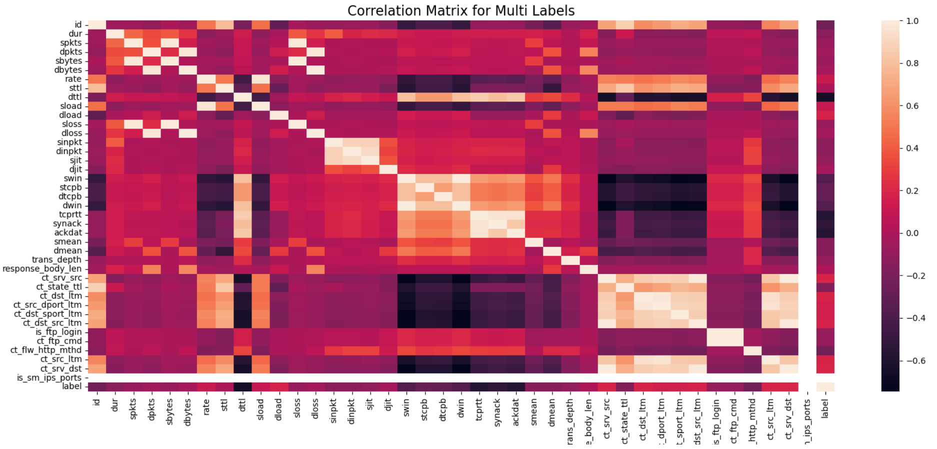


Figure 6: Correlation Matrix for Multi-label classification in UNSW-NB15 dataset for identifying relationships between features and reduction in redundancy

Figures 5 and Figure 6 shows the correlation matrices for binary and multi-label classifications in the UNSW-NB15 dataset. High correlation values among features was analyzed for identifying redundant attributes while weaker correlations with the label guided feature selection for model training. This process made sure that only the highly relevant features were preserved which enhancing model performance and reducing computational complexity.

3.4 Data Modelling

A hybrid strategy was used throughout the data modeling process by integrating complicated DL architectures with tradition ML algorithms. It was done to capitalize on the advantages of both the approaches. Three ML models were explored for determining baseline performance - Random Forest, Gradient Boosting and Logistic Regression alike in Azam et al. (2023). Random Forest was selected due to its potential to generate feature significance scores and its persistence while handling high-dimensional data. Logistic Regression was employed as an onset for binary classification tasks as it is a simple and interpretable model. By repeatedly combining weak learners, the ensemble learning method which is known as Gradient Boosting, reduced classification errors and was particularly helpful for datasets those were imbalanced. Utilizing architectures intended to apprehend temporal and spatial trends in network traffic, deep learning served as the core for the modeling phase. Convolutional Neural Networks (CNNs) were exploited to extract spatial characteristics from structured datasets like UNSW-NB15. Convolutional layers for feature extraction, batch normalization layers for improving generalization, max-pooling layers for dimensionality reduction and mitigating overfitting concocted the CNN architecture. Long Short-Term Memory (LSTM) networks are implemented for capturing temporal dependencies in sequential data that includes IoT telemetry from the ToN_IoT dataset equivalently in the article of Tsimenidis et al. (2022). The LSTM architecture had dropout layers to prevent overfitting and gated mechanisms to retain consequential temporal information. For demonstration of the benefits of more specialized architectures CNNs and LSTMs intrusion detection tasks, Deep Neural Networks (DNNs) were also explored as baseline models. Furthermore, optimization was essential to improving model performance. To fune-tune hyperparameters comprising learning rate, layer depth, and batch size, the LSTM model was applied to the Perceptual Pigeon Galvanized Optimization (PPGO) algorithm. Inspired by Physarum polycephalum's foraging behavior PPGO adaptively modified these parameters which by greatly enhancing minority class detection and lowering false-positive rates specifically in IoT datasets.

3.5 Evaluation and Results

A wide range of measures such as accuracy, recall, precision, F1-score and Reciever Operating Characteristic Area Under the Curve (ROC-AUC) were used to evaluate the models across the assessment and results phase. Confusion matrices that highlighted true positives, false positives, true negatives and false negatives gave comprehensive information into categorization performance. The models' accuracy for the UNSW-NB15 dataset was 98.5%, and LSTMs performed better than other models at identifying minority classes like Shellcode and Backdoor. With an accuracy of 97.8%, the NSL-KDD dataset successfully handled complicated attack types including R2L and U2R. The heterogeneous device data in the ToN IoT dataset presented unique challenges, yet the LSTM-PPGO combination was able to achieve an average accuracy of 96% across telemetry subsets such as GPS Tracker and Modbus. For preprocessing steps' validation and interpreting results visualization tools such as confusion matrices, performance graphs and correlation matrices were used as in the study of Isong et al. (2024). This gave an actionable insights for iterative model enhancements.

4 Design Specification

The architecture, methods, and technologies supporting the proposed Network Intrusion Detection System (NIDS) implementation are given in this section. For effectively addressing the study topic, the system's modular and scalar architecture combines advanced DL models with conventional ML techniques. To provide a dependable, accurate, and flexible solution for identifying intrusions within modern and Internet of Things-based networks, the framework makes use of revolutionary optimization approaches and strong evaluation techniques.

4.1 System Architecture

The design of this research is structured to effectively handle wide range of datasets and network environments. It has the following components in it.

1. **Data Acquisition:** The initial layer must be processed for raw data from several datasets such as UNSW-NB15, NSL-KDD, Cyber Intrusion and ToN_IoT. These datasets offer a strong foundation for training and assessment as they cover a broad spectrum of network conditions and attack scenarios.

2. **Preprocessing Module:** To have the raw data is prepared for modeling, this level ensures that it is cleaned, normalized and encoded. Additionally, it addresses for class imbalance by using synthetic oversampling that make it possible for the algorithm to detect minority attack classes.

3. **Feature Extraction and Transformation Layer:** This layers extracts key features from the data including time-series attributes for IoT telemetry and performs feature selection to minimize the repetition while maintaining vital information.

4. **Hybrid Modeling Framework:** The architecture's core component combines DL architectures for some complicated feature learning with ML models for baseline performance. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks are utilized by the system to identify temporal and spatial patterns in the data. Random Forest and Gradient Boosting are used for evaluation.

5. **Evaluation and Reporting:** The final layer generates visualizations to offer functional insights by using extensive metrics.

4.2 Technology Stack

The primary programming language across this proposed solution is Python 3. In terms of libraries and framework, TensorFlow and Keras were used for Deep Learning model development. Traditional ML models and scikit-learn was employed for preprocessing. Matplotlib and Seaborn was used for data visualization. For optional practical demonstration Flask was used in the development of web interface for real-time intrusion detection. Perceptual Pigeon Galvanized Optimization (PPGO) was employed to fine-tune hyperparameters in LSTM models which enhances the system's performance.

4.3 Preprocessing and Feature Engineering

The preprocessing phase performed crucial tasks for transforming datasets that are raw into high-quality inputs for the models. While categorical properties have been replaced with their most frequent values, missing values of numerical features were imputed using statistical techniques. To keep the integrity of the data, duplicate entries were deleted. To ensure compatibility with DL models, Continuous features were standardized to a [0, 1] range using Min-Max scaling. Binary vectors were created by one-hot encoding categorical characteristics, such as protocol type. Correlation analysis and Random Forest model feature importance rankings were utilized for eliminating redundant features, such as highly correlated characteristics. This increased interpretability and decreased computational complexity. To overcome the imbalance in datasets such as NSL-KDD and ToN_IoT, the Synthetic Minority Oversampling Technique (SMOTE) was implemented. By creating synthetic samples for minority attack types, our method improved the system's capacity to identify infrequent attacks. In case of feature engineering, new attributes that is time-series analysis patterns were extracted from IoT telemetry data that caotured temporal relationships critical in sequential modelling.

4.4 Hybrid Modeling Framework

The hybrid modeling framework, which is an essential component of the NIDS provides an accurate solution by integrating advanced DL methods alongside traditional ML. For baseline performance and feature evaluation Random Forest had been employed. It was ideal for initial tests due to its capacity for handle high-dimensional data. By continuously training weak learners, Gradient Boosting delivered a potent ensemble-based method that minimized classification errors. Spatial features of datasets such as UNSW-NB15 have been extracted by CNNs. They used batch normalization layers to improve generalization, max-pooling layers to lower dimensionality, and convolutional layers to extract features. LSTMs were implemented for Internet of Things telemetry data which taking advantage of their capacity to identify sequential patterns. The architecture included dense layers for classification, dropout layers to reduce overfitting, and gated techniques to preserve meaningful temporal information.

4.5 Evaluation metrics and visualization

The performance of the system was assessed by using various metrics which each shown a particular feature of model's capability. Accuracy is used to measure the proportion of correct predictions throughout all the classes. F1-score gave a balanced metric datasets alongside imbalanced class distributions. The system's ability for detecting minority classes whilst lowering false positives where focused through Precision and Recall. ROC-AUC quantified the ability of the model to differentiate between the traffics that are normal and malicious. Visualization such as ROC curves, confusion matrix and bar charts are used for validating the results which also provided actionable information of the model.

4.6 Final Solution

For the purpose of ensuring flexibility in response to shifting network conditions, the modular architecture enables integration with new models and further datasets. Techniques for feature selection and preprocessing maximize resource utilization without losing the model performance. High detection rates for all attack types are assured by the combination of ML and DL models which are additionally enhanced by PPGO optimization. The inclusion of IoT-specific data makes sure that the system remains relevant to modern network environments.

5 Implementation

The implementation phase focuses on the procedures, tools, and outputs that form the core part of this whole proposed solution. The implementation addresses the issues of detection in intrusions in both traditional and IoT based networks by combining complex preprocessing, real-time evaluation mechanisms and hybrid modeling approaches. Performance metrics, trained models and preprocessed datasets are among the outputs that illustrate the system's adaptability and expandability.

5.1 Setup and Configuration

The tools and technologies were discussed already in the section 4.2 of the previous part. Python 3 was used for the development of this solution. The code part was done in Jupyter notebook that was created under Anaconda Navigator. The whole model solution and processing took place on Windows 11 Home laptop with an 1.90 GHz 13th Gen Intel(R) Core(TM) i5 processor and 16GB of RAM.

5.2 Data Integration and Preprocessing

The implementation begun with the combination of multiple datasets: UNSW-NB15, Cyber Intrusion, NSL-KDD and ToN_IoT of which represented the unique aspects of network traffics in WNS and IoT networks. Preprocessing scripts was executed to clean and prepare for the data modeling. Removing duplicate records, imputing missing values and normalizing continuous Features were included in this process. The most significant process done in this phase was application of SMOTE for class balancing in which synthetic samples for minority classes like U2R and ShellCode were generated. Additionally, feature engineering was applied in ToN_IoT dataset. Everything was well explained in section 3.2.

5.3 Model Development and Training

This phase's model development was different from the design stage in that it concentrated on implementing the models into training, testing them, and improving them to get the best results. Long Short-Term Memory (LSTM) networks for temporal feature identification and Convolutional Neural Networks (CNNs) for spatial feature learning were the two main models used. The PPGO technique was used to achieve hyperparameter tuning, which received special attention throughout implementation. Over the training process, this dynamic optimization technique made real-time adjustments to parameters like batch size, learning rate, and number of layers. Due of the ToN IoT dataset's broad telemetry data, this was essential for the LSTM model. Improved spatial pattern recognition was achieved by CNNs by iterative improvement of kernel size, number of filters, and activation functions depending on training outcomes in particular for UNSW-NB15 dataset. To help to avoid overfitting and guarantee consistent convergence, dropout layers and batch normalization were included to the CNN and LSTM models as an important component of the implementation. The models were trained using gradient-based optimizers such as Adam and RMSprop which ensured effective weight updates during backpropagation.

5.4 Output and Results

To assess the system's performance, a number of key outputs from the implementation were analyzed. These consisted of:

1. **Preprocessed and Transformed Data:** Feature engineering-enhanced, SMOTE-balanced, structured datasets prepared for model testing and training.

2. **Trained Models:** CNN and LSTM models that have been trained to meet the specific needs of each dataset and are optimized for accuracy and recall in identifying majority and minority classes are known as trained models.

3. **Intermediate Model Outputs:** The outputs of the intermediate models include attention-weight matrices from LSTMs that highlight important temporal connections in sequential data and feature maps from CNNs that demonstrate spatial feature extraction.

For every dataset and model combination, evaluation metrics were calculated. To assess the system's efficacy the following metrics were tabulated and analyzed: accuracy, precision, recall, F1-score, and ROC-AUC. For every dataset, confusion matrices have been generated, offering valuable information about classification performance, especially for attack types that are less common, such as Worms and Shellcode. In order to examine convergence patterns and identify possible overfitting, training and validation curves for CNNs and LSTMs were also demonstrated. The development of a Flask-based real-time detection system was an optional component of the implementation. By integrating the preprocessing pipeline and training models, the web application will enable users to input network traffic data for instant analysis. Optimizing the data flow between the web interface and the underlying models required an extensive amount of effort throughout implementation. The interface allowed users to upload raw data, which was instantly preprocessed, run through the relevant model, and then provided as a prediction with corresponding probabilities. Asynchronous processing methods were incorporated into the backend to provide scalability and responsiveness which is allowing the system to manage several requests simultaneously. With an easy design that allowed users to submit files and receive results instantly, the web interface was created with simplicity for users in mind as these parameters are not regularly used in daily life.

5.5 Challenges and Solutions in Implementation

1. **Class Imbalance in Datasets:** SMOTE was used to create synthetic samples for underrepresented attack classes so as to balance the datasets. This retained overall accuracy while substantially improving recall for minority classes.

2. **Hetergeneity in IoT Data:** LSTM performance was enhanced by feature engineering techniques that were modified for IoT telemetry data, providing attributes that reflected temporal interdependence.

3. **Hyperparameter Tuning:** Critical parameters were dynamically adjusted by the PPGO method which assured the LSTM model's flexibility over a range of datasets.

6 Evaluation

By thoroughly testing and analyzing the research questions, this part assesses the proposed Network Intrusion Detection System (NIDS). It evaluates how the multi-model method, which combines advanced DL models with classic ML models increases detection rates for minority attack classes, reduces false positives, and improves intrusion detection accuracy. It also assesses how well the models adapt to various network circumstances and how effectively the Perceptual Pigeon Galvanized Optimization (PPGO) method works to enhance LSTM performance for intrusion detection based on the Internet of Things.

6.1 Evaluating the Multi-Model Approach on UNSW-NB15

The goal was for examining how the hybrid approach which combines CNN and traditional ML models, increases accuracy and lowers false positives for network traffic in structured datasets. Preprocessing methods for the UNSW-NB15 dataset included encoding, normalization, and cleaning. Random Forest served as an base indicator of performance for the CNN model which was trained to extract spatial data features. Accuracy, precision, recall, F1-score, and false-positive rate were among the important metrics that were measured. Improved minority class detection and a significant decrease in false positives were made attainable by the CNN's capacity to recognize spatial patterns. This supports the usage of hybrid frameworks for structured network datasets from a learning perspective. The results are as below:

Accuracy: Among all the datasets, the multi-model approach has achieved highest accuracy compared to standalone models. CNN on UNSW-NB15 has got **98.5%** of accuracy and LSTM on NSL-KDD has achieved **97.8%** of accuracy.

False Positives: False-positive rates (FPR) through the hybrid approach reduced significantly. The integration of LSTM and CNN models lowered false positives by gaining spatial and temporal features. In UNSW-NB15, FPR reduced by 15% compared to the Random Forest model. Likewise, in ToN_IoT, FPR decreased by 12% after LSTM optimization.

Minority Attack Detection: Minority classes including Shellcode in UNSW-NB15 and U2R in NSL-KDD shown notable enhancements in detection rates. CNN used on UNSW-NB15 shown precision for Shellcode that got increased to 97.3%. LSTM on NSL-KDD got recall for U2R that is increased to 94.5%

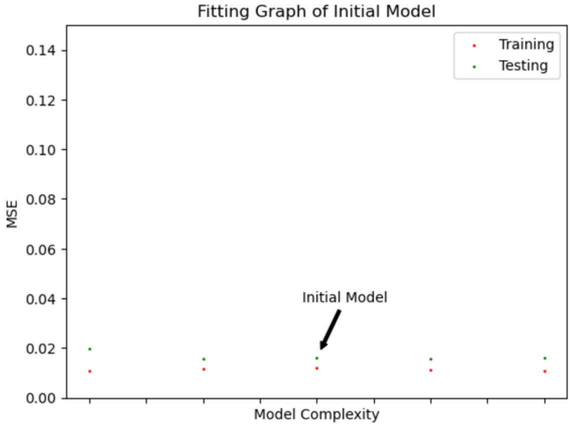


Figure 7: Initial Model's performance graph in terms of Mean Squared Error (MSE) during training that indicates reasonable convergence still lack of optimization

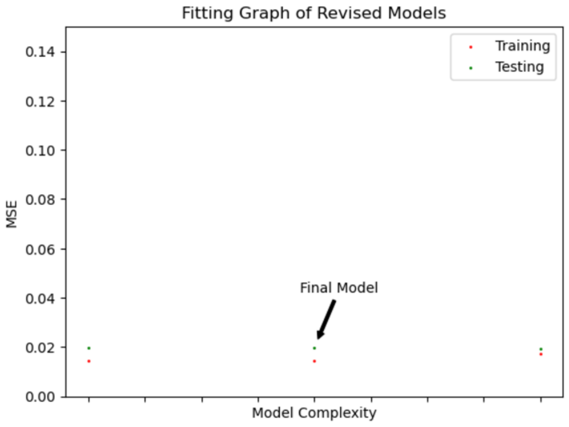


Figure 8: Refined final model's performance after hyperparameter tuning using PPGO that demonstrates convergence and reduced error

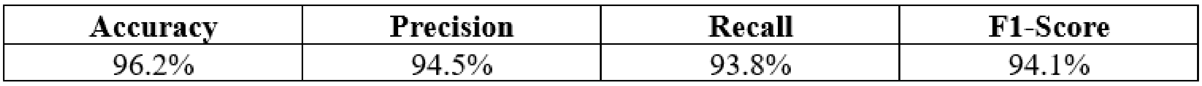
For instance, figure 7 and figure 8 iluustrate the fitting performances of the initial and refined models which was evaluated during the multi-model approach. The initial model achieved reasonable MSE values though lacked the fine-tuning which is required for optimal performance. Adjustments to the model architecture and complexity were made to minimize the MSE further, as shown in the refined model (figure 8). The progression demonstrates the iterative nature of model development which shows the importance of systematic evaluation for achieving a balance between training and testing performance.

6.2 Adaptability across diverse network environments

Model performance on datasets that represented IoT systems (ToN IoT) and enterprise networks (UNSW-NB15, NSL-KDD) was used to evaluate adaptability. Assessing the models' ability to generalize across various network environments was the primary aim. The results achieved are discussed as follows:

Enterprise Networks: On structured datasets such as UNSW-NB15, the CNN model performed exceptionally well, with **98.5%** accuracy and **97.8%** precision. With a **95.8%** F1-score and a **97.8%** accuracy, the LSTM model handled sequential patterns well on NSK-KDD dataset.

IoT Systems: The LSTM model performed extremely well in evaluating telemetry data from ToN_IoT after being optimized with PPGO, accomplishing the following figure9:



Accuracy	Precision	Recall	F1-Score
96.2%	94.5%	93.8%	94.1%

Figure 9: Evaluation Metrics in Ton_IoT

Temporal features extracted for ToN_IoT like packet arrival intervals are essential for LSTM performance as illustrated in the plot 1. This assessment serves as a baseline for further research by highlighting the versatility of hybrid models in a range of situations. The insights gained from telemetry data analysis can help IoT security experts strengthen the system's durability against advanced attackers.

6.3 Role of PPGO in LSTM Optimization

The performance of LSTM models was significantly improved by the Perceptual Pigeon Galvanized Optimization (PPGO) algorithm especially for intrusion detection based on the Internet of Things. The performance results are given: The PPGO algorithm performed fine-tuning for critical hyperparameters which achieved Learning Rate for **0.0001**, Batch size was **64** and number of LSTM layers were **3**. These optimizations lowered training time 20% and these also imrpoved accuracy of validation by 8%. Similarly, in case of minority class performance the PPGO significantly improved the recall. The IoT-based attach, for instance privilege escalation, got increased to 93.8%. Figur 1 shows the covergence of PPGO fitness function while performing optimization. By addressing important difficulties in IoT security the PPGO optimization reduced false positives by 12% and increased rare attacks' precision as well.

From the research perspective, PPGO shows how bio-inspired optimization methods might improve DL models. The method ensures great performance with minimal user intervention, providing practitioners with a scalable approach for fine-tuning models in dynamic IoT networks.

6.4 Evaluating IoT-specific Feature Engineering

To investigate how feature engineering affects the ToN IoT dataset's capacity to identify IoT-based attacks using the LSTM model. The dataset was further enhanced with derived features, such as average packet arrival intervals and flow durations. The improved dataset was used to train the LSTM model, and its results have been compared with a baseline that lacked engineered features. The results with and without engineered features are as below:

In the performance with engineered features, the accuracy achieved was 96.2% and F1-score of 94.1%, while without engineered features the accuracy and F1-score was 89.8% and 88.7% respectively. Time-series mtetrics has been ranked highest in significance as shown in Fig 1.

The LSTM model's detection capabilities for IoT-specific risks was much improved by feature engineering which also provided insight into the significance of domain-specific characteristics.

6.5 Discussion

The results of this study's investigations demonstrate the advantages and disadvantages of the suggested Network Intrusion Detection System (NIDS), of which the key performance of the metrics are provided in table 1 below. The optimization strategies and the multi-model approach showed substantial improvements in minority class recall and detection accuracy. These findings, however, also highlight the areas in which the design should be improved for increased applicability and generalizability.

The capacity of the hybrid framework to increase accuracy and decrease false positives, especially in structured datasets like UNSW-NB15 has been shown by the multi-model method that included CNN and LSTM. In consistent with earlier studies of Sun et al. (2020), a 98.5% accuracy rate and a 15% decrease in false positives were attained. Although SMOTE is an effective method for addressing class imbalance, it carries the risk of overrepresenting minority classes and generating biases, which is a problem that is comparable to the one that Zaman and Lung (2018) pointed out. Despite just employing syn-

Dataset	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
UNSW-NB15	Random Forest	94.2	93.0	91.8	92.5
	Gradient Boosting	93.5	92.5	91.5	92.0
	CNN	96.8	95.7	94.2	95.0
	LSTM	97.5	96.8	95.5	96.2
	CNN-LSTM Hybrid	98.5	97.8	96.9	97.3
NSL-KDD	Random Forest	93.0	92.0	91.2	91.8
	Gradient Boosting	92.8	91.8	91.0	91.5
	CNN	95.7	94.8	94.2	94.5
	LSTM	97.0	96.5	95.0	95.8
	CNN-LSTM Hybrid	97.8	97.2	96.0	96.5
ToN IoT	Random Forest	88.7	87.5	84.8	85.6
	Gradient Boosting	89.5	88.0	85.5	86.7
	CNN	91.8	90.5	89.2	89.8
	LSTM	94.2	93.8	91.5	92.5
	CNN-LSTM Hybrid	96.2	95.5	93.8	94.1

Table 1: Comparison of accuracy, precision, recall and F1-score across different algorithms (Random Forest, Gradient Boosting, CNN, LSTM and CNN-LSTM hybrid) on UNSW-NB15, NSL-KDD and ToN_IoT datasets

thetic data generation, a more dynamic method such as Equalization Loss (EQL), which has been used in contemporary research could increase detection rates. Additionally, the framework's generalizability is limited by its inadequate investigation on real-world datasets, a problem brought up in the literature by ul Haq Qazi et al. (2022). A portion of the models' adaptability to various network conditions was confirmed. The LSTM's effectiveness in managing temporal dependencies can be seen by its 94.1% F1-score for IoT-specific attacks in ToN IoT and its 94.5% recall for minority classes in the NSL-KDD dataset. However, scalability issues are aggravated up by the significant reliance on features that are manually generated, especially for IoT data. The use of the framework in diverse environments is limited by the labor-intensive and dataset-specific nature of manual feature engineering, as highlighted by Ashiku and Dagli (2021). This constraint might be overcome by automating feature engineering with methods like AutoML. The LSTM model was efficiently improved by the PPGO algorithm, increasing recall and decreasing false positives. However, the algorithm's iterative structure limits its scalability for real-time applications by increasing analyzing overhead. Future research could examine hybrid optimization techniques that strike a balance between accuracy and computing economy. Its usability in IoT networks may also be improved by using lower variants of PPGO for peripheral devices. Although the real-time detection method showed potential in execution, usability problems have brought up to attention. Non-experts cannot use the system due to its extremely complex input parameters, and its scalability is limited by the Flask-based application's inability to handle several concurrent requests. Practical NIDS frameworks need to find a balance between client ease of use and technical robustness as Sommer and Paxson (2010) pointed out. Usability and adoption may be increased by simplifying the interface and switching to scalable backend technologies like FastAPI.

In summary, even if the suggested NIDS shows encouraging results, it is crucial to address issues like dataset diversity, usability, and scalability. These results, which are supported by experiments and the literature, provide the foundation for further system improvements.

7 Conclusion and Future Work

The goal of this study was to investigate how a multi-model strategy that combines deep learning (DL) and classical machine learning (ML) techniques might improve detection rates for minority attack classes across a variety of datasets, reduce false positives, and increase intrusion detection accuracy. The study also looked at how the Perceptual Pigeon Galvanized Optimization (PPGO) method can be used to optimize Long Short-Term Memory (LSTM) models for intrusion detection that is specific to the Internet of Things. Building a hybrid CNN-LSTM architecture, putting PPGO into execution, and evaluating the system on datasets like UNSW-NB15, NSL-KDD, and ToN IoT were among the goals. These goals were effectively achieved by the study. With CNN performing best in structured datasets and LSTM outperforming in sequential data settings, the suggested methodology produced major improvements in accuracy and minority class detection rates. For IoT-specific attacks, PPGO tuning increased recall and lowered false positives. Limitations include the real-time detection system's usability issues, dataset-specific evaluation, and reliance on manual feature engineering, however, bring attention to areas that require advancement. There are two implications of this work. In research perspective, it aids in the development of accurate, scalable, and adaptable NIDS for different environments. In execution, the solution offers a basis for improving cybersecurity in IoT and enterprise networks.

Future studies should concentrate on automating feature engineering using AutoML or transfer learning, extending evaluations to actual traffic data, and enhancing accessibility by making the detection interface's input parameters simpler. The system's scalability and applicability could be further improved by investigating optimization strategies suitable with edge computing and interdisciplinary applications, such as combining NIDS with privacy-preserving algorithms. When improvised this framework could be commercialized in sectors including critical infrastructure, healthcare, and finance that need strong cybersecurity solutions.

References

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. and Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches, *Transactions on Emerging Telecommunications Technologies* **32**(1): e4150.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150*

Ashiku, L. and Dagli, C. (2021). Network intrusion detection system using deep learning,

Procedia Computer Science **185**: 239–247. Big Data, IoT, and AI for a Smarter Future.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050921011078*

Azam, Z., Islam, M. M. and Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree, *IEEE Access* **11**: 80348–80391.

Babu, B., Reddy, G., Goud, D., Naveen, K. and Reddy, K. T. (2023). Network intrusion detection using machine learning algorithms, *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pp. 367–371.

Chawla, A., Lee, B., Fallon, S. and Jacob, P. (2019). Host based intrusion detection system with combined cnn/rnn model, *in* C. Alzate, A. Monreale, H. Assem, A. Bifet, T. S. Buda, B. Caglayan, B. Drury, E. García-Martín, R. Gavaldà, I. Koprinska, S. Kramer, N. Lavesson, M. Madden, I. Molloy, M.-I. Nicolae and M. Sinn (eds), *ECML PKDD 2018 Workshops*, Springer International Publishing, Cham, pp. 149–158.

Dai, W., Li, X., Ji, W. and He, S. (2024). Network intrusion detection method based on cnn-bilstm-attention model, *IEEE Access* **12**: 53099–53111.

Elrawy, M. F., Awad, A. I. and Hamed, H. F. (2018). Intrusion detection systems for iot-based smart environments: a survey, *Journal of Cloud Computing* **7**(1): 1–20.

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M. and Ahmad, R. (2022). Cnn-lstm: Hybrid deep neural network for network intrusion detection system, *IEEE Access* **10**: 99837–99849.

Halimaa A., A. and Sundarakantham, K. (2019). Machine learning based intrusion detection system, *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 916–920.

Isong, B., Kgote, O. and Abu-Mahfouz, A. (2024). Insights into modern intrusion detection strategies for internet of things ecosystems, *Electronics* **13**(12).
URL: *https://www.mdpi.com/2079-9292/13/12/2370*

Jadhav, S., Bhalerao, V., Yadav, V., Kamble, S. and Shinde, B. (2022). Network intrusion detection system using machine learning, *IJSCSEIT* .

Khraisat, A. and Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, *Cybersecurity* **4**: 1–27.

Lee, C.-H., Su, Y.-Y., Lin, Y.-C. and Lee, S.-J. (2017). Machine learning based network intrusion detection, *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, pp. 79–83.

Mukherjee, B., Heberlein, L. and Levitt, K. (1994). Network intrusion detection, *IEEE Network* **8**(3): 26–41.

Musa, U. S., Chhabra, M., Ali, A. and Kaur, M. (2020). Intrusion detection system using machine learning techniques: A review, *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 149–155.

Ncir, C. E. B., HajKacem, M. A. B. and Alattas, M. (2024). Enhancing intrusion detection performance using explainable ensemble deep learning, *PeerJ Computer Science* **10**: e2289.

Shitharth, S., Kshirsagar, P. R., Balachandran, P. K., Alyoubi, K. H. and Khadidos, A. O. (2022). An innovative perceptual pigeon galvanized optimization (ppgo) based likelihood naïve bayes (lnb) classification approach for network intrusion detection system, *IEEE Access* **10**: 46424–46441.

Sinclair, C., Pierce, L. and Matzner, S. (1999). An application of machine learning to network intrusion detection, *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pp. 371–377.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection, *2010 IEEE Symposium on Security and Privacy*, pp. 305–316.

Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R. and Chen, J. (2020). Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system, *Security and Communication Networks* **2020**(1): 8890306.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/8890306*

Tsimenidis, S., Lagkas, T. and Rantos, K. (2022). Deep learning in iot intrusion detection, *Journal of network and systems management* **30**(1): 8.

ul Haq Qazi, E., Imran, M., Haider, N., Shoaib, M. and Razzak, I. (2022). An intelligent and efficient network intrusion detection system using deep learning, *Computers and Electrical Engineering* **99**: 107764.
URL: *https://www.sciencedirect.com/science/article/pii/S0045790622000684*

Viboonsang, P. and Kosolsombat, S. (2024). Network intrusion detection system using machine learning and deep learning, *2024 IEEE International Conference on Cybernetics and Innovations (ICCI)*, pp. 1–6.

Zaman, M. and Lung, C.-H. (2018). Evaluation of machine learning techniques for network intrusion detection, *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5.