

Module: Research Methods and Professional Development

**Title: Machine Learning-Based Network Intrusion
Detection System**

Student Name: MD. MASHRURUL HAQUE

Student Number: 2435257

Lecturer: Dr. Suresh Shanmugasundaram

List of Figures:.....	3
1. Introduction.....	4
1.1 Context and Background	4
1.2 Research Problem	4
1.3 Research Aim and Objectives	5
1.4 Significance of the Research	5
2. Literature Review.....	7
2.1 Machine Learning in IDS.....	7
2.2 Dataset Challenges.....	7
2.3 Model Performance and Optimization	7
2.4 Evaluation Metrics and Practical Challenges	7
2.5 Real world Application and Limitations	8
2.6 Literature Gap.....	8
3. Research Design and Methodology	9
3.1 Research Design	9
3.2 Research Methodology Workflow.....	9
3.3 Dataset Collection.....	10
3.4 Data Preprocessing	11
3.5 Machine Learning Models	11
3.6 Model Training and Testing	12
3.7 Evaluation Metrics	12
3.8 Model Comparison and Final Selection.....	12
3.9 Research Materials	12
3.10 Ethical Consideration	12
4. Expected Outcome	13
5. Timeline	13
6. References	15
7. Appendices.....	17

List of Figures:

Fig.1: Research Methodology Workflow of the Proposed Study -----	10
Fig.2: Fig.2: Data Preprocessing Flow-----	11
Fig.3: Project Expected Timeline Grantt Chart -----	16

1. Introduction

1.1 Context and Background

In today's digital world, our reliance on the internet and connected services has exploded. While this brings incredible benefits, it also means a bigger target for cyberattacks. Organizations depend heavily on their networks for daily operations, making them prime targets for threats like malware, Distributed Denial of Service (DDoS) attacks and unauthorized intrusions represent significant cybersecurity threats, as they have the potential to cause substantial harm to systems and services.

Intrusion Detection Systems (IDS) serve a critical function in observing network activity and identifying potential threats, yet conventional methods largely rely on signature-based detection mechanisms which means they can only detect known threats and often fail against unknown attacks [1]. To combat these dangers, Intrusion Detection Systems (IDS) were developed. Their job is to keep an eye on network traffic and flag anything suspicious. Traditional IDS, especially those that rely on signatures, are great at catching attacks they already know about. However, they struggle to identify new or evolving threats, which limits their effectiveness in our increasingly complex and dynamic network environments.

This is where machine learning (ML) comes in. ML offers a powerful solution to enhance intrusion detection by enabling systems to automatically learn patterns from data and recognize anomalies. Unlike conventional methods, ML-based IDS can learn from past data to identify previously unseen attacks, significantly improving overall security. Machine learning (ML) has become a viable way to overcome these constraints. ML-based intrusion detection systems (IDS) are better at spotting attacks that haven't been observed before since they can recognize anomalies and learn patterns from network data [2]. ML's relevance in cybersecurity has been reinforced by recent developments in data analytics and artificial intelligence.

This study aims to design and evaluate a network intrusion detection system based on machine learning techniques. Our primary goals are to accurately identify malicious activity in network data, improve the system's ability to adapt to modern cyberthreats, and reduce the number of false alarms, ultimately leading to better detection performance.

1.2 Research Problem

Despite notable improvements in Network Intrusion Detection Systems (IDS) achieved through machine learning techniques, a number of critical challenges that limit their effectiveness in real world environments. Many existing ML-based IDS models show very high accuracy when tested on standard datasets, but this performance does not always translate well to practical network settings.

Many ML-based IDS models perform poorly in real-world situations despite achieving great accuracy in controlled settings. One significant problem is the high false positive rate, which causes warning fatigue in security operations by mistakenly classifying legitimate traffic as hostile [3].

Another major issue is the high number of false positives, where normal network activity is mistakenly flagged as malicious. In real-world security operations, this can overwhelm analysts with unnecessary alerts and reduce trust in the system. At the same time, most machine learning models are trained using known attack patterns, which makes them less effective at identifying new or unknown threats, often referred to as zero-day attacks.

Another concern is that models trained on specific datasets may not perform consistently across different network environments. Networks are dynamic, and traffic patterns can vary widely, which makes it difficult for a single model to generalize well in all situations. In addition, many machine learning approaches are not designed with real-time performance in mind, leading to delays in detecting attacks—something that can be critical in fast-moving cyber incidents.

These limitations make it necessary to find out how machine learning might be applied to intrusion detection systems more successfully. Thus, the following issue is the main focus of this study is How can a machine learning-based network intrusion detection system be built to detect attacks reliably, minimize false alarms, spot threats that haven't been seen before, and function effectively in real-time network environments. Furthermore, because zero-day attacks don't follow established patterns, identifying them is still quite difficult [4]. The reliance on particular datasets, which restricts models' capacity to generalize across various network contexts, is another problem [5]. Furthermore, the practical utility of many systems is diminished because they are not built for realtime detection.

1.3 Research Aim and Objectives

Research Aim:

The goal of this research is establishing and evaluate a machine learning-based network intrusion detection system that can reliably detect malicious network activity while minimized false alarms, enhancing the detection of previously undetected attacks, and guaranteeing effective operation in real-time settings.

Research Objectives:

To achieve this aim, the study will focus on the following objectives:

1. To review existing IDS and machine learning techniques
2. To select and preprocess relevant intrusion detection datasets
3. To develop multiple machine learning models
4. To evaluate model performance using standard metrics
5. To compare and optimise models for best performance
6. To reduce false positives and improve detection of unknown attacks
7. To assess real-time feasibility of the proposed system

1.4 Significance of the Research

The fact that this study tackles important shortcomings in existing intrusion detection systems makes it noteworthy. The suggested solution can improve cybersecurity operations, especially in Security Operations Centers (SOC), by increasing detection accuracy and decreasing false

positives. The research's conclusions can advance both theoretical understanding and real-world implementations. From a scholarly standpoint, it offers suggestions for enhancing ML-based IDS models. Practically speaking, it can help organizations create security systems that are more dependable and effective.

2. Literature Review

This review of the literature looks at previous studies on machine learning IDS, demonstrating significant advancements, popular methods, extant constraints. Finding significant trends and pointing out gaps that support this research is the goal.

2.1 Machine Learning in IDS

By empowering computers to identify patterns in network traffic, machine learning has greatly enhanced intrusion detection. ML and deep learning models perform better than conventional signature-based systems, especially when it comes to identifying unknown threats, according to studies like Liu et al. [1] and Xin et al. [9].

DL approaches [12], [13] advanced further by feature extraction automatically from data. Though, these kinds of models need high computational resources and large volume of datasets, which limits their practical use. Additionally, Sommer and Paxson [3] argue a key limitation is that many machine learning models provide strong results in controlled environments, yet fail to generalise effectively when applied to real-world scenarios.

2.2 Dataset Challenges

Datasets are essential to IDS functionality. While older datasets like NSL-KDD [7] are still used for benchmarking, more recent datasets like CIC-IDS2017 [6] offer realistic traffic patterns. Nevertheless, research [14], [19] emphasizes that out-of-date datasets do not accurately reflect the behavior of attacks today. Model performance and generalization may also be impacted by dataset bias [15]. This demonstrates how the reliability of IDS research is directly impacted by dataset selection.

2.3 Model Performance and Optimization

ML models including Logistic Regression, Decision Trees, Random Forest, and SVM, have been used for intrusion detection [17]. Performance has been demonstrated to be enhanced by ensemble approaches [8] and feature selection methods [18]. Pinto et al. [2] and Berman et al. [4] point out that many models continue to have large false positives and poor minority attack class recognition despite their high stated accuracy. This emphasizes the need for a more impartial assessment.

2.4 Evaluation Metrics and Practical Challenges

Evaluation metrics are essential for assessing the effectiveness of IDS models. Precision, accuracy, recall, and F1-score are commonly used metrics [5]. However, several researchers argue that accuracy alone is insufficient, particularly in imbalanced datasets where normal traffic dominates. One of the most critical challenges is the high false positive rate. Axelsson [10] and Scarfone and Mell [11] highlight that excessive false alerts can overwhelm security analysts, reducing the practical usability of IDS systems.

Another key issue is detecting zero-day attacks. Many models struggle to identify attacks that do not follow known patterns, as discussed by Berhili [20]. This limitation highlights the need for more adaptive and intelligent detection approaches.

2.5 Real world Application and Limitations

Although machine learning-based IDS have shown promising results, their real-world deployment remains limited. Yang et al. [5] introduced an open-source IDS framework, but its evaluation was primarily conducted in controlled environments. Similarly, Kim et al. [16] and Dua and Du [17] emphasise that real-world network environments are highly dynamic, making it difficult for models to maintain consistent performance. Computational complexity, scalability, and real-time processing also remain major challenges.

These limitations suggest that current IDS research focuses more on experimental performance rather than practical implementation.

2.6 Literature Gap

In conclusion, current research shows that machine learning greatly improves intrusion detection capabilities. But there are still a few significant obstacles to overcome:

- High false positive rates
- Poor detection of zero-day attacks
- Strong dependency on specific datasets
- Limited generalisation across environments
- Lack of real-time performance evaluation

These limitations highlight a clear research gap between high experimental accuracy and real-world applicability.

3. Research Design and Methodology

3.1 Research Design

This research follows a design-oriented and experimental approach, where a machine learningbased intrusion detection system will be developed and evaluated. The design science approach is suitable for this study because the main goal is to build a practical system that can detect network intrusions effectively. At the same time, an experimental method is used to test and compare different machine learning models using benchmark datasets. This combined approach allows both system development and performance evaluation, ensuring that the results are meaningful for realworld cybersecurity applications.

3.2 Research Methodology Workflow

This study's whole methodology is organized and systematic, beginning with the selection of datasets and concluding with the identification of the best machine learning model. Every step is designed to ensure that the models are properly trained, the data is appropriately prepared, and the outcomes are methodically assessed.

The following flowchart illustrates step-by-step process used in this study to develop and assess the machine learning-based intrusion detection system:

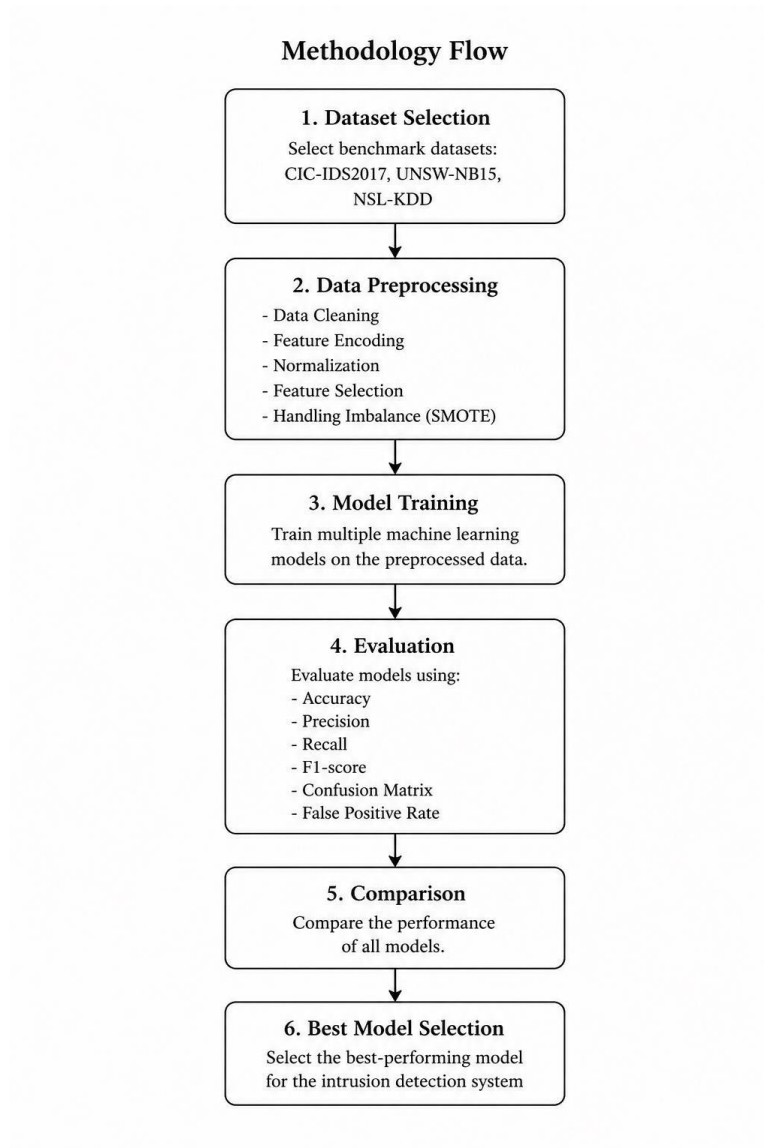


Fig. 1: Research Methodology Workflow of the Proposed Study

3.3 Dataset Collection

The study will use well-known and widely accepted intrusion detection datasets:

- CIC-IDS2017 (Primary Dataset)
- UNSW-NB15 (Secondary Dataset)
- NSL-KDD (Baseline Dataset)

These datasets are selected because they contain a wide range of modern attack types and realistic network traffic patterns. Using multiple datasets also helps to improve the generalisation capability of the model and avoids overfitting to a single dataset.

3.4 Data Preprocessing

Due to the inherently noisy and unstructured nature of raw network data, data preparation represents a critical stage in machine learning–based intrusion detection systems. The preprocessing process includes several key steps: data cleaning, which involves removing missing values, duplicate records, and irrelevant information; feature encoding, where categorical variables are transformed into numerical representations; data normalisation, which scales features to ensure consistency across inputs; feature selection, aimed at identifying the most relevant attributes to reduce model complexity and enhance performance; and handling data imbalance through the application of the Synthetic Minority Oversampling Technique (SMOTE) to achieve a more balanced dataset.

A data preprocessing flowchart is given below:

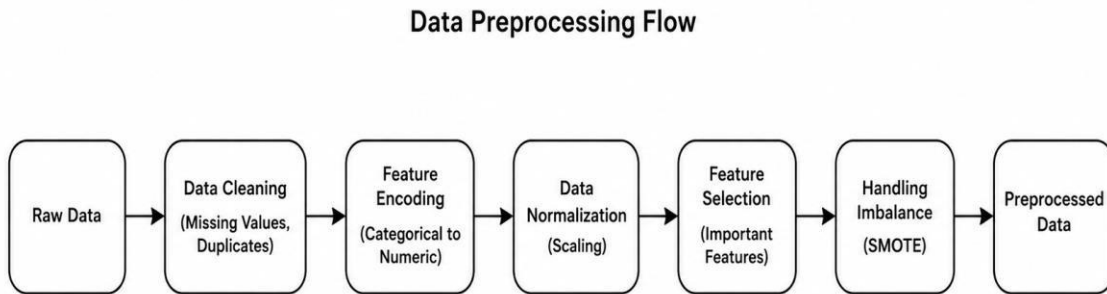


Fig.2: Data Preprocessing Flow

3.5 Machine Learning Models

To assess the effectiveness of different approaches, multiple machine learning models will be implemented. Logistic Regression will be used as a simple and efficient classification method that estimates the probability of class membership and performs well on linearly separable data. Decision Trees will be applied as an interpretable model that divides data into branches based on decision rules, although it may be prone to overfitting if not properly controlled. Random Forest, an ensemble technique, will be employed to combine multiple decision trees in order to enhance accuracy and reduce overfitting, making it suitable for complex datasets. Support Vector Machine (SVM) will also be utilised as a powerful classification method that identifies optimal decision boundaries, particularly effective in high-dimensional spaces, though computationally intensive. These models have been selected to represent diverse learning paradigms, including linear, tree-based, and kernel-based approaches, enabling a comprehensive and reliable comparison of their performance.

3.6 Model Training and Testing

The dataset will be partitioned into training and testing subsets to ensure an unbiased evaluation of model performance. The training set will be utilised for model development, while the testing set will be used to assess performance. Additionally, cross-validation techniques may be employed to enhance reliability and minimise overfitting.

3.7 Evaluation Metrics

The performance of the models will be assessed using standard evaluation metrics, including accuracy, precision, recall, F1-score, confusion matrix, and false positive rate (FPR). Accuracy reflects the overall correctness of the model, while precision measures the proportion of correctly identified attacks. Recall evaluates the model's ability to detect actual attack instances, and the F1-score provides a balanced measure between precision and recall. The confusion matrix offers a visual representation of prediction outcomes, and FPR is particularly important for minimising false alarms. Collectively, these metrics provide a comprehensive evaluation of model performance.

3.8 Model Comparison and Final Selection

After evaluation, all models will be compared based on their performance metrics.

The final model will be selected based on:

- Highest accuracy
- Lowest false positive rate
- Better detection of unknown attacks

This step ensures that the most effective and practical model is chosen for the proposed intrusion detection system.

3.9 Research Materials

Software Tools to be used:

- Python
- Scikit-learn
- Pandas and NumPy
- Jupyter Notebook
- Matplotlib / Seaborn

3.10 Ethical Consideration

To make sure that no private or sensitive data is used, this study will make use of publically accessible databases. As a result, there are minimal ethical risks. However, during the study, appropriate data handling and responsible use of the findings will be upheld.

4. Expected Outcome

Instead of depending only on accuracy, this research is anticipated to yield a reliable and empirically validated machine learning-based intrusion detection system that exhibits better performance across various assessment dimensions. The creation of a model that strikes a better balance between detection capability and false alarm reduction is one of the main anticipated results. This balance is crucial in real-world cybersecurity settings, especially in Security Operations Centers (SOC), since too many false positives can overwhelm analysts and overlooked attacks might result in major security breaches.

Additionally, the study is anticipated to demonstrate how dataset selection and data preprocessing techniques affect model performance and generalization. The research attempts to offer insights into the dependability and transferability of machine learning models in various network contexts by assessing the system across several benchmark datasets.

Another key outcome of this study is a comparative assessment of various machine learning algorithms, highlighting the trade-offs among simplicity, interpretability, computational cost, and detection performance. This analysis aims to identify models that are more suitable for practical deployment rather than those that only perform well in theoretical settings.

Furthermore, the research is expected to demonstrate the feasibility and limitations of applying machine learning to near real-time intrusion detection, limitations such as processing overhead and scalability.

From a broader perspective, the findings are anticipated to contribute to the ongoing effort to bridge the gap between high experimental performance and real-world applicability of intrusion detection systems. This includes providing guidance on designing IDS solutions that are not only accurate but also operationally effective.

5. Timeline

The project is planned over a period of approximately five months, with each phase structured to ensure a logical progression from theoretical understanding to practical implementation and evaluation.

In the first month, the focus will be on conducting an in-depth literature review and refining the research proposal. This phase involves analysing existing studies on machine learning-based intrusion detection systems, identifying research gaps, and clearly defining the research objectives. Establishing a strong theoretical foundation at this stage is essential for guiding the later stages of the project.

During the second month, attention will shift towards dataset selection and initial data exploration. Relevant datasets such as CIC-IDS2017, UNSW-NB15, and NSL-KDD will be analysed to understand their structure, features, and limitations. At the same time, preliminary preprocessing steps, including data cleaning and feature understanding, will begin.

The third month will primarily focus on data preprocessing and model development. This includes applying techniques such as feature encoding, normalization, feature selection, and handling data imbalance using methods like SMOTE. Following this, multiple machine learning models will be implemented and prepared for training.

In the fourth month, the focus will be on model training, testing, and evaluation. The developed models will be trained using the prepared datasets and assessed using performance metrics, including accuracy, precision, recall, F1-score, and false positive rate. A comparative analysis will also be carried out to identify the strengths and limitations of each model.

Finally, the fifth month will be dedicated to finalising the research. This includes selecting the best performing model, interpreting the results, and completing the final report. Additionally, time will be allocated for reviewing the work, making necessary improvements, and preparing the presentation.

Overall, this timeline ensures a balanced distribution of tasks, allowing sufficient time for both technical implementation and critical analysis, while also maintaining flexibility to address potential challenges during the research process.

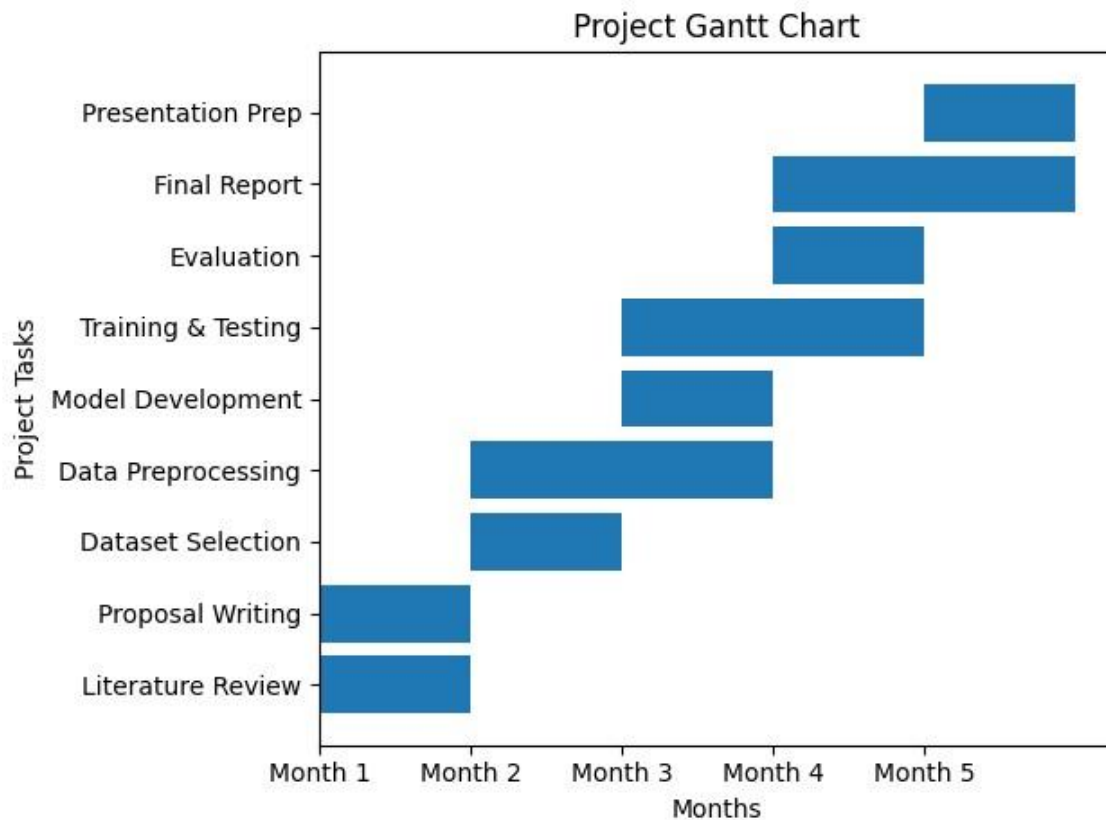


Fig.3: Project Expected Timeline Grantt Chart

6. References

- [1] H. Liu, B. Lang, M. Liu, and H. Yan, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," *Applied Sciences*, vol. 9, no. 20, pp. 1–27, 2019.
- [2] A. Pinto, R. M. S. Silva, and A. A. P. de Medeiros, "A Survey on Intrusion Detection Systems Based on Machine Learning," *Sensors*, vol. 23, no. 5, pp. 1–35, 2023.
- [3] M. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 305–316.
- [4] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A Survey of Deep Learning Methods for Cyber Security," *Information*, vol. 10, no. 4, pp. 1–35, 2019.
- [5] L. Yang, H. Wang, and X. Li, "IDS-ML: An Open-Source Machine Learning Framework for Intrusion Detection," *SoftwareX*, vol. 21, pp. 1–10, 2022.
- [6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. Int. Conf. Information Systems Security and Privacy*, 2018, pp. 108–116.
- [7] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Dataset," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.
- [8] S. M. Kasongo and Y. Sun, "A Deep Learning Method with Filter-Based Feature Engineering for Wireless Intrusion Detection System," *IEEE Access*, vol. 8, pp. 38597–38607, 2020.
- [9] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, and H. Zhu, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
- [10] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," *Technical Report*, Chalmers University, 2000.
- [11] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," *NIST Special Publication 800-94*, 2018.
- [12] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in *Proc. EAI Int. Conf. Bio-inspired Information and Communications Technologies*, 2016, pp. 21–26.

- [13] S. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Trans. Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [14] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of NetworkBased Intrusion Detection Data Sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.
- [15] M. Zolanvari, M. A. Teixeira, S. Aminanto, A. H. Lashkari, and A. A. Ghorbani, "Generating a New Intrusion Detection Dataset: CIC-IDS2017," *IEEE Int. Conf. Big Data*, 2020.
- [16] J. Kim, H. Kim, and H. Kim, "Long Short-Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *IEEE Access*, vol. 4, pp. 1–10, 2016.
- [17] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Boca Raton, FL, USA: CRC Press, 2016.
- [18] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using Feature Selection for Intrusion Detection System," *IEEE Int. Symp. Communications and Information Technologies*, 2010.
- [19] S. S. Tripathy and B. Behera, "A Review of Datasets for Machine Learning-Based Intrusion Detection Systems," *arXiv preprint arXiv:2506.02438*, 2025.
- [20] M. Berhili, "Intrusion Detection Systems in IoT Based on Machine Learning: A Review," *Procedia Computer Science*, vol. 235, pp. 1–10, 2024.

7. Appendices

Appendix A: Dataset Overview

This research utilises three widely recognised datasets for intrusion detection:

- **CIC-IDS2017**
Contains realistic network traffic with modern attack types such as DDoS, brute force, and infiltration attacks. It is used as the primary dataset due to its relevance to real-world scenarios.
- **UNSW-NB15**
Includes both normal and malicious traffic with diverse attack categories. It is used to test the generalisation capability of the models.
- **NSL-KDD**
An improved version of the KDD Cup 1999 dataset. It is used as a baseline dataset for comparison with more recent datasets.

These datasets provide a comprehensive foundation for evaluating the performance of machine learning models in intrusion detection.

Appendix B: Data Preprocessing Steps

The preprocessing stage involves several key steps to prepare the data for machine learning:

- Elimination of missing and duplicate values
- Transferring categorical into numerical values
- Feature normalization to maintain data consistency
- Selection of relevant features
- Handling class imbalance using SMOTE

These steps ensure that the dataset is clean, balanced, and suitable for model training.

Appendix C: Machine Learning Models

The following machine learning models are used in this study:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)

Each model is selected based on its ability to handle classification problems and its relevance to intrusion detection tasks.

Appendix D: Evaluation Metrics

The performance of the models is evaluated using the following metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- False Positive Rate

These metrics provide a comprehensive evaluation of the system's effectiveness in detecting intrusions.

Appendix E: Tools and Software

The implementation of this research is carried out using:

- Python programming language
- Scikit-learn library
- Pandas and NumPy for data processing
- Jupyter Notebook for development
- Matplotlib and Seaborn for data visualization

Appendix F: Methodology Flow Description

The research follows a structured workflow consisting of:

1. Dataset Selection
2. Data Preprocessing
3. Model Training
4. Evaluation
5. Comparison and Final Model Selection

This workflow ensures a systematic approach to developing and evaluating the intrusion detection system.