Table of Contents

Students' Property Right Declaration and Anti-Plagiarism Statement i

Table of Contents ii

List of Tables iii

List of Figures iii

Abstract iv

Chapter 1: Introduction 1

1.1 Problem Statement and Motivation 1

1.2 Background 1

1.3 Aims and Objectives 1

1.4 Current Solutions 2

1.5 Full Overview of Dataset. 2

Chapter 2: Background 4

2.1 Importance and Context 4

2.2 The Target Market and Its Needs 4

2.3 Ethical and Environmental Issues 5

2.4 Summarization 6

Chapter 3: Design 7

Chapter 4: Implementation 10

4.1 Methods and Tools 10

4.2 Infrastructure 10

4.3 Trade-offs in Design/Implementation 10

4.4 Dependencies/Assumptions 10

Chapter 5: Results and Discussion 11

Chapter 6: Economical, Ethic, and Contemporary Issues 12

6.1 Cost Estimation and Justification 12

6.2 Relevant Codes of Ethics and Moral Theories 12

6.3 Ethical Dilemmas and Justification of the Proposed Solution 13

6.4 Relevant Environmental Considerations 13

6.5 Relevance to Jordan and Region-Social, Cultural, and Political 14

Chapter 7: Project Management 15

7.1 Project Schedule and Time Management 15

7.2 Resource and Cost Management 15

7.3 Quality Management 15

7.4 Risk Management 15

7.5 Project Procurement 16

Chapter 8: Conclusion and Future Work 17

8.1 Algorithmic Performance Evaluation 17
8.2 Further Future Work to Enhance the Solution/System. 17
8.3 Lessons Learned 17
References 18

List of Tables

Table 1. Comparison of Intrusion Detection Techniques: Advantages and Limitations ………………..2
Table 2. Design Considerations …..……………………………..……………………………………..9
Table 3. Results Table Title …..…… …………………………. …………..………………………10

List of Figures

Figure 1. Label Distribution Chart…..…………………………….....…………………….....………….. 3
Figure 2. Traffic Type Distribution Chart …..…………………………..…………………………..…….4
Figure 3. Traffic Subtype Distribution chart …..…………………………...………...……..……... 4
Figure 4. Two-Phase diagram Preparation and Machine Learning Modeling ...…..……...……..……... 5
Figure 5: Distribution of NIDS Market Across Various Sectors ..……….……………….....………….. 6

Abstract

The project aims to improve the detection and prevention of cyber threats using machine learning in network security. Thus, the TII-SSRC-23 dataset is considered, which includes benign and malicious network traffic. Therefore, the work will look forward to developing an enhanced NIDS capable of detecting real-time attacks with unprecedented accuracy. Traditional intrusion detection methods cannot keep pace with newly emerging attack techniques and hence require more adaptive and intelligent solutions. This work presents several machine learning models, including Decision Tree, Random Forest, Gradient Boosting, KNN, SVM, and XGBoost, to determine the most effective model that can give high detection accuracy with low false positives. The dataset is prepared for model training through data preprocessing, class balancing, and feature engineering. It contributes to building a robust real-time detection system, enhancing network security by precisely identifying and mitigating cyber threats, hence enhancing the efficiency of detection compared to traditional systems.

Keywords: Network Security, Intrusion Detection system, Machine Learning.

Chapter 1: Introduction

1.1 Problem Statement and Motivation

Within the last ten years, personal, corporate, and government activities have depended a lot on the use of the internet and network systems. [1] As much as these dependences have brought immense revolutions in the way we communicate and do business, they have equally increased the rate of cyber threats. These include, but are not limited to, information leakage, unauthorized access, and malicious activities [2] within networks. It has therefore become evident, with the ever-increasing sophistication and complexity of these cyber-attacks, that conventional security measures based on signature-based detection methods cannot stand effectively against such modern network threats[7]

Traditional IDS relies on attack signatures predefined and previously known patterns. However, due to the various limitations mentioned above, traditional systems cannot detect unknown and emerging threats using signatures that are outdated and old. There is an emerging need and necessity for developing advanced systems capable of detecting and mitigating a cyber-attack in real time[1]

1.2 Background

IDS are designed to monitor network traffic for suspicious activities or violation of policies. Traditional IDS systems, though effective in detecting attack patterns already known, usually fail in the identification of new unknown threats[2]. Since cyber-attacks are becoming more and more developed, new and innovative attack vectors keep coming out; similarly, the development of IDS needs to keep pace with the detection of emerging patterns[5].

Due to this, the research has started shifting towards machine learning-based IDSs as a potential solution. They have advantages in utilizing big data for training the models to recognize new, unseen attack patterns. However, such models strongly depend on the quality and balance of the data used in their training. Datasets like TII-SSRC-23 have been quite useful in carrying out cybersecurity research, yet they are replete with some challenges such as large size and class imbalance[3]. In this regard, preprocessing of data plays a significant role in building a good machine learning model for intrusion detection [4].

1.3 Aims and Objectives

The main objective of this project is the design and development of an advanced machine learning-based Intrusion Detection System using the TII-SSRC-23 dataset. Specific objectives of this study are as follows:

- **Data Cleaning and Reduction**: Remove duplicate records and irrelevant columns to enhance the manageability and improve the speed of data processing.
- **Class Balancing**: The dataset is imbalanced; balancing must be done either through under-sampling or over-sampling to make the model learn effectively from all classes of traffic.
- **Feature Preprocessing**: Scale continuous features and then encode categorical variables in order to prepare the dataset for machine learning modeling[4]
- **The models to be trained and evaluated:** include Random Forest, Gradient Boosting, and XGBoost, among others, using accuracy, precision, recall, and F1-score performance metrics.

By the end of the project, the goal is to identify the most effective machine learning model for real-time detection and classification of malicious network traffic.

1.4 Current Solutions

There are several intrusion detection techniques, each with certain strengths and weaknesses. The different techniques are as follows:

- **Signature-based Detection:** It is very effective in detecting the known attacks cataloged in the signature database. However, the limitation is in its inability to detect zero-day exploits or new attack strategies. The signature database has to be updated regularly for the detection of new threats [1].
- **Anomaly-Based Detection:** These anomaly detection systems learn the "normal" behavior of network traffic and detect deviations from this. Such systems can usually detect previously unknown threats but often result in a higher rate of false positives due to the challenges of defining "normal" network behavior accurately [2].
- **Hybrid Detection Systems**: These systems use both signature-based and anomaly-based detection methods to extend the coverage and reduce false positives. While they offer more flexibility, hybrid systems are often more resource-intensive and require more computational power [3].

Detection Type	Advantages	Limitations
Signature-based	- Effective for known threats. - Low false positives.	- Inability to detect new threats. - Requires continuous database updates.
Anomaly-based	- Can detect unknown threats. - Adaptable to evolving attack patterns.	- High false positive rate. - Requires large amounts of training data.
Hybrid Systems	- Combines strengths of both approaches. - Reduces false positives.	- Resource-intensive and complex. - Requires significant investment in hardware and expertise.

Table 1. Comparison of Intrusion Detection Techniques.

1.5 Full Overview of Dataset.

The TII-SSRC-23 dataset is pretty huge and diverse, having both benign and malicious network traffic. It has a total of 86 feature dimensions, covering almost every type of traffic variation, and hence can be very well used for training any machine learning model related to network intrusion detection. Following are the key characteristics of this dataset:

- **Size**: 8,656,767 rows × 86 columns , 5.5 GB.
- **Data Types:** 78 numeric features, 7 categorical features, and 1 integer feature.
- **Class Distribution**: This dataset is highly imbalanced, as the majority of the classes are occupied by malicious traffic, while benign ones make up only a small portion. The imbalanced classes pose a problem with model training and evaluation.
- **Traffic Types**: The following are some of the traffic types that compose this dataset: Denial of Service, Information Gathering, and Mirai Botnet. The distribution of these types of traffic is important in understanding the nature of the dataset and the types of attacks that need to be detected.[3]

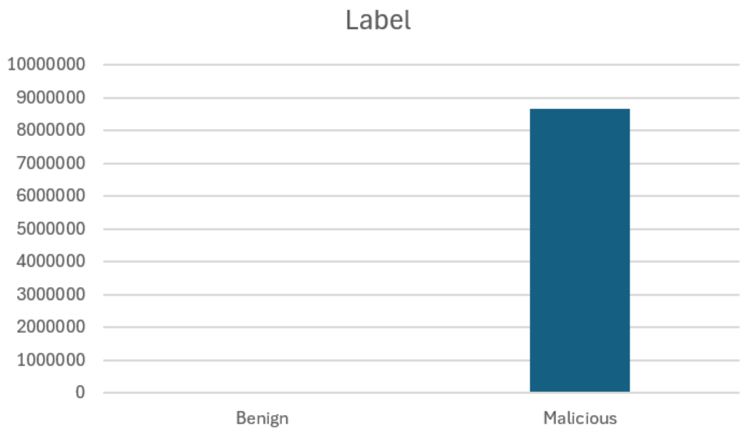


Figure 1. Label Distribution Chart

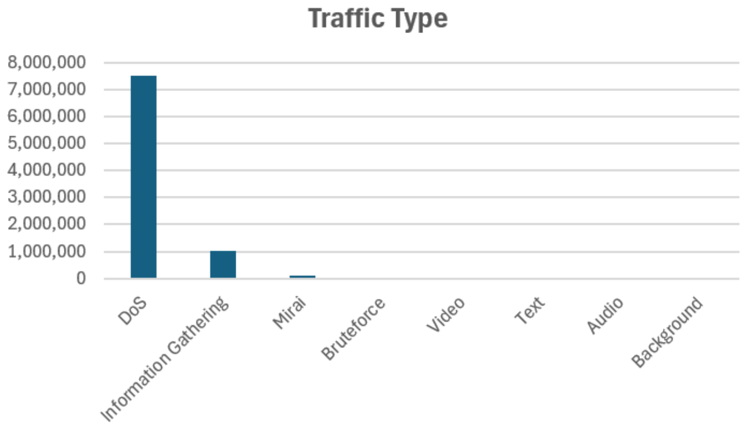


Figure 2. Traffic Type Distribution chart

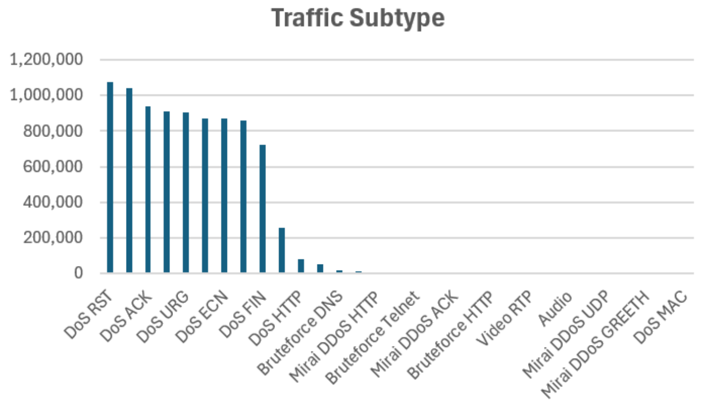


Figure 3. Traffic Subtype Distribution chart

Chapter 2: Background

2.1 Importance and Context

Network security remains at the top of all agendas in organization concerns regarding information, infrastructure, and system protection. While the digital world keeps on growing, so does the frequency and sophistication of cyber-attacks, which involve serious consequences at the level of data breaches, system downtime, and financial losses. From simple denial-of-service attacks to sophisticated persistent threats, cyber threats require comprehensive adaptive solutions.[10}

The detection and mitigation of such threats have become an essential function of Network Intrusion Detection Systems. Their role is very crucial in monitoring network traffic for the detection of malicious activities, often before they cause harm. However, traditional NIDS, based on signature-based detection techniques, are fast becoming ineffective against newer sophisticated threats. Because intrusion detection systems traditionally depend on predefined attack patterns, attackers can easily evade these with new techniques or polymorphic malware.

Added to these challenges are the recently popular machine learning-based approaches to intrusion detection. Such models are capable of mining knowledge from extensive datasets to identify attacks that were previously unknown and adapt to emerging threats with a view to reducing false alarms while increasing intrusion detection accuracy.

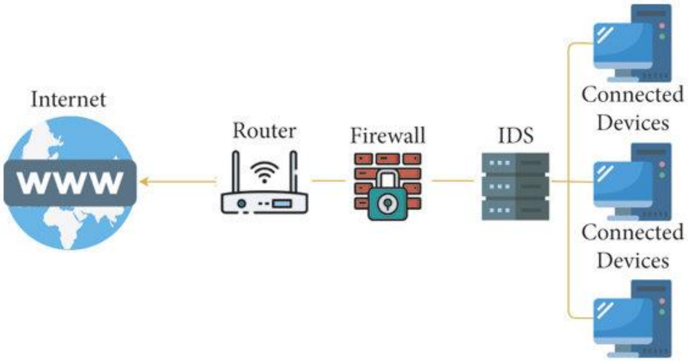


Figure 4. Importance of Network Security and NIDS

2.2 The Target Market and Its Needs

Effective network intrusion detection systems are needed for several sectors, each with specific needs. Some of the key markets for NIDS solutions are:

- **Large Organizations and Government Agencies**: These store very sensitive, critical data and are thus quite vulnerable to hacking. For that reason, advanced solutions of NIDS are necessary with the view of detecting complex sophisticated threats and facilitating compliance regulatory mechanisms such as GDPR, HIPAA, among others. Their vast infrastructure and complicated networks raise the requirement for highly scalable and efficient systems of NIDS [1].
- **SME**s represent a common target of cyber-attacks; because of that fact, the urgent need to solve this problem has arisen in order to support automated and inexpensive NIDS solutions which can easily be implemented in those organizations due to low maintenance [2].
- **Health Care and Education Institutions**: The data of patients and students dealt with here are highly sensitive and thus the most coveted in the field of cyber-attacks. Unfortunately, health care and educational organizations lack dedicated cybersecurity professionals, so they need NIDS solutions to protect their critical information, which should also be cost-effective and easy to manage [3].

- **Critical Infrastructure and National Security**: Power grids, transportation systems, and water supply networks are all critical infrastructures that have become more and more subject to cyber-attacks. It requires the governments and defense organizations to have sophisticated NIDS for the detection and neutralization of potential threats that may disrupt the essential services and jeopardize national security [4].

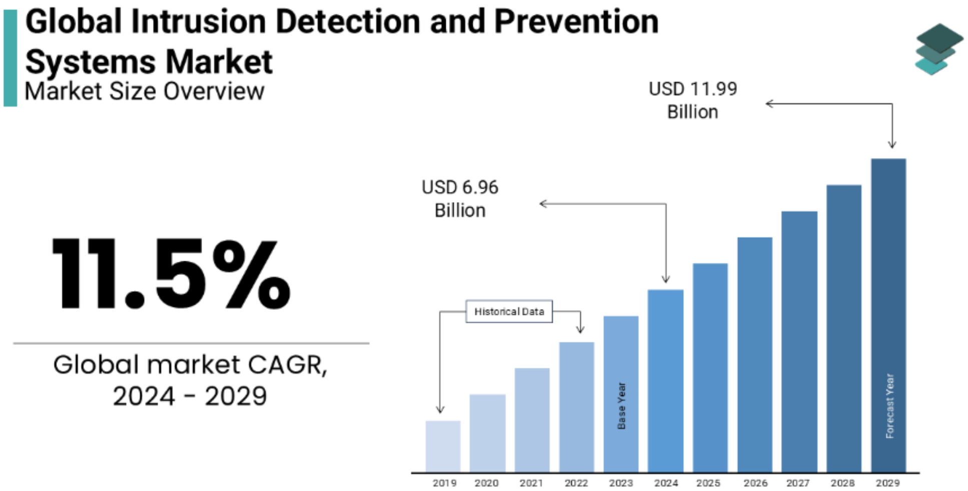


Figure 5: Distribution of NIDS Market Across Various Sectors[11]

2.3 Ethical and Environmental Issues

Other factors to consider when implementing solutions with NIDS include a number of ethical and environmental issues, including:

- **Privacy Risks**: With regard to violation of privacy, NIDS is subject to violation of users' privacy since it provides continuous monitoring of network traffic; sensitive personal information may be at stake. Any development of an NIDS system should be done by protecting data in conformity with GDPR, ensuring that no monitoring activity violates the individual rights to privacy [5].
- **Information Abuse**: the information extracted by the NIDS systems has to be secured in order to prevent misuse. If it detects, for instance some sensitive information; proper care should be taken to prevent such data from getting across to unauthorized parties. This includes leaks and proper security protocols during storage and transmission [6].
- **Energy Consumption and Ecological Footprint**: It always requires significant computational overhead in order to execute NIDS systems on organizational networks, especially if one wants to run huge networks. Certainly, it does contribute towards carbon footprint through such consumption. Efficiency and optimization in the system have considered the important means to minimize NID-related ecological footprints [7].
- **Compliance with Local Laws**: Other countries have varying laws with regard to Cyber Security and data protection; hence, NIDS solutions must be compliant within the precincts of local and international laws in order to be ethically correct and not face any legal hassle.

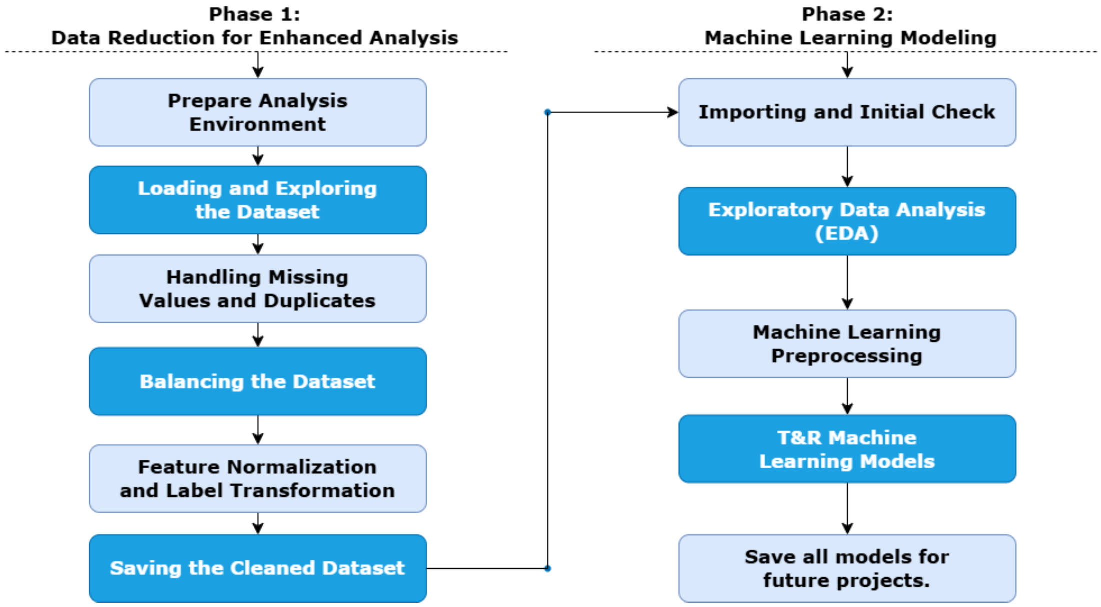
2.4 Summarization

The need for network security is continuously growing because of the increasing number and difficulty of cyber-attacks. Traditional NIDS relies on a signature-based detection approach, which has no ability to detect modern, sophisticated threats. Machine learning-based NIDS will be able to solve this issue by allowing the detection of attacks in real time. However, the successful accomplishment of NIDS depends on issues related to the quality of data, balance between classes, and capabilities for adaptation in front of evolving threat landscapes.

The deployment of NIDS solutions should go along with necessary ethical considerations concerning privacy, data protection, and environmental impact. In addition, the need to understand the demands from sectors like large enterprises, SMEs, and critical infrastructure would guarantee a wide acceptance and overall success of the NIDS solutions.

Chapter 3: Design

The improved version of Network Intrusion Detection will be focused on, using the TII-SSRC-23 dataset. This dataset captures benign and malicious network traffic. The proposed design will focus on two important phases, as represented below.


Phase 1:
Data Reduction for Enhanced Analysis
Prepare Analysis Environment
Loading and Exploring the Dataset
Handling Missing Values and Duplicates
Balancing the Dataset
Feature Normalization and Label Transformation
Saving the Cleaned Dataset
Phase 2:
Machine Learning Modeling
Importing and Initial Check
Exploratory Data Analysis (EDA)
Machine Learning Preprocessing
T&R Machine Learning Models
Save all models for future projects.

Phase 1: Data Reduction for Analysis Improvement

This is the preparation and refining step of the data in order to render it analysis-ready.

1. **Prepare Analysis Environment:** Set up all tools, libraries, and frameworks required for data analysis and Ensure data security and compliance measures are put in place.

2. **Loading and Exploring the Dataset:** Importing data into an appropriate tool for data analysis. And Perform the initial exploratory analysis to get a sense of the structure, content, and possible problems in the dataset.

3. **Handling Missing Values and Duplicates:** Identifying missing data and deciding on imputation or removal and Removing duplicate entries so that the dataset does not contain duplicate records.

4. **Balancing the Dataset:** Making necessary changes to the dataset to fix the imbalance problems of the target variable this is usually the case when one deals with classification problems. Some techniques involve oversampling, undersampling, or synthetic data generation.

5. **Feature Normalization and Label Transformation:** Scaling or normalization of features: to bring features into a similar range in cases where a model is sensitive to input scale. Label transformation : whether for classification or regression.

6. **Saving the Cleaned Dataset:** Save the preprocessed data in a file or database for convenient access during subsequent analyses or sessions of model training.

Phase 2: Machine Learning Modeling
This involves building and testing the machine learning models from the cleaned and prepared data.

1. **Importing and Initial Check:** Loading cleaned dataset and checking initially that integrity and readiness exist in the data for modeling.

2. **Exploratory Data Analysis:** With further analysis to bring forth the patterns, anomalies, relationships, and insight from the data.Statistical summaries will be used and also visualization techniques.

3. **Machine Learning Preprocessing:** Further preparation of the data for machine learning-specific tasks, such as encoding categorical variables, feature selection, and splitting the data into training and test sets.

4. **Train & Refine Machine Learning Models:**
   Training on the prepared training dataset with various machine learning algorithms Tuning models to come up with highly accurate ones using the test dataset.

5. **Save All Models for Future Projects:** Save the trained models, their parameters, and state for deployment or further refinement in future projects.

How You Intend to Address the Problem Statement
.

- **Active Intrusion Detection**: Most of the traditional systems never take action on zero-day attacks. Our system, on the other hand, learns the pattern from real network traffic TII-SSRC-23 using machine learning and adapts to the newest of threats.
- **Reduction of False Positives:** The balancing and feature engineering makes the final model taught on finding those minute differences among the benign and malignant traffic.
- **Scalability**: Reduction of data from large raw files (~30 GB to ~1 GB) enables training without prohibitive resource usage, while cloud options such as Colab or AWS handle large-scale training if necessary.

Environmental Factors:

- Minimize resource consumption by focusing on efficient algorithms such as Random Forest and XGBoost.
- Leverage cloud services - Google Colab for large-scale operations only when necessary.

Operational and Cost:

- Licensing cost reduced by leveraging open-source Python-based libraries such as scikit-learn and pandas.
- The stored models can be retrained periodically when new threats or updated data are available.

Legal Aspects

- **Privacy and Data Protection:** Following any GDPR-like regulations in case of any real user data embedded.

- **TII-SSRC-23 is a public dataset;** however, standard practices will be followed to avoid exposure of sensitive information.

Design Constraints

- **Computational Resources:** Large dataset (~8 million rows) demands efficient memory usage or cloud compute.
- **Time Constraints**: Model training can be time-consuming; utilize parallel processing (n_jobs=-1 in scikit-learn).

Data Quality: Dependence on the accuracy and coverage of real-world attack vectors in the labeling provided by TII-SSRC-23.

Design Alternatives

- **Deep Learning Approaches:** Could give a better accuracy but require more GPU resources - e.g., TensorFlow-based CNN/RNN
- **Hybrid Systems:** Rule-based with anomaly detection adds accuracy but is added complexity.

Safety Consideration

- **False Positives** : Too aggressive classifiers bring down legitimate traffic - will need calibration
- **Model Overfitting**: Frequent cross-validation and balanced data help reduce the chance of overfitting.

Design Considerations Table

Design Consideration	Description	Project Application	Relevant Location
Performance	Process large datasets in minimal time	TII-SSRC-23 with 8M+ rows	Ch. 4 (Implementation), Ch. 5 (Results)
Serviceability	Easily retrain models as attacks evolve	Regular updates if new threats arise	Ch. 7 (Project Management)
Economic	Use open-source tools, low-cost cloud services	Python, scikit-learn, Colab free tier	Ch. 6.1 (Cost)
Environmental	Optimize computing resources, reduce footprint	Cloud usage only when essential	Ch. 6.4 (Environmental Considerations)
Ethical	Respect data privacy, fairness in model decisions	Comply with data protection laws; handle malicious data responsibly.	Ch. 6.2 (Ethics)
Health & Safety	Minimize disruptions and misclassifications	Lower false positives in real-time deployment.	Ch. 3.2.7 (Safety)
Social & Political	Contributing to cybersecurity in Jordan & region	align with local laws strengthen national security posture.	Ch. 2.2 & 6.5 (Regional Context)

Chapter 4: Implementation

4.1 Methods and Tools

- **Local Machine**: Basic data manipulation and partial training feasible with ~8-16GB RAM.
- **Cloud Services**: Google Colab for GPU/TPU acceleration.

Software / Libraries

- Python (3.x): Primary language for data science.
- pandas: Data loading, exploration, and manipulation.
- scikit-learn: Preprocessing (MinMaxScaler, StandardScaler) and classic ML algorithms (Logistic Regression, etc.).
- XGBoost: Powerful gradient boosting library for imbalanced data.
- opendatasets: For direct Kaggle dataset downloads.
- TensorFlow/Keras (Optional): If deep learning approaches are tested.
- joblib: Model saving/loading for future usage.

4.2 Infrastructure

- **Local/On-Prem Environment:** Initial analysis and partial data balancing can run locally.
- **Cloud:** For complete training on the balanced dataset (potentially millions of rows) to expedite training times.

4.3 Trade-offs in Design/Implementation

Computation vs. Model Complexity

- Simple models (Logistic Regression) are easier to interpret but may yield slightly lower accuracy.
- Complex ensemble methods (Random Forest, XGBoost) can achieve higher accuracy but need more computational resources.

Accuracy vs. False Positives

- Stricter detection thresholds increase true positives but risk more false alarms.

Dataset Size vs. Resource Usage

- Reducing dataset size to 1 GB helps with performance; however, excessive reduction might remove rare attack patterns.

4.4 Dependencies/Assumptions

- **Data Quality**: The TII-SSRC-23 dataset is assumed to be representative of real-world traffic.
- **Label Accuracy**: The “benign” vs. “malicious” labels, as well as subtypes, are correct.
- **Periodic Updates**: New variants of attacks might require periodic re-training

Chapter 5: Results and Discussion

5.1 Results

After running a complete pipeline on the TII-SSRC-23 dataset-clean, balance, feature engineering-and after training several models, the following metrics depict the performance of the system. Here is a sample summary-replace with your actual values:

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.94	0.93	0.91	0.92
Decision Tree	0.98	0.97	0.98	0.97
Random Forest	0.98	0.98	0.97	0.97
Gradient Boosting	0.97	0.96	0.95	0.95
K-Nearest Neighbors (KNN)	0.97	0.96	0.95	0.95
Support Vector Machine	0.95	0.94	0.92	0.93
XGBoost	0.99	0.99	0.99	0.99

- As expected, **XGBoost** has the best performance, very close to perfection for this particular dataset.
- **Decision Tree** also achieved a high accuracy but might be suffering from overfitting.
- Logistic Regression is simpler and more interpretable but lags a bit concerning accuracy.

Link to Design

The high accuracy with the balanced metrics is testimony to the success of the data balancing and feature normalization strategies explained in **Chapter 3**.

Strengths

- Robust Feature Engineering: Normalization and balancing have greatly improved model performance, especially for minority classes.
- Versatile Algorithm Selection: The training of multiple models assists in finding the best fit for high-dimensional intrusion data.
- Scalable Implementation: The approach supports both local and cloud resources.

Weaknesses

- **High Computational Load**: Handling millions of rows can be very slow and sometimes requires special hardware - GPU/TPU.
- **Possible Overfitting**: Complex models-for example, ensembles-might overfit in case of poor hyperparameter tuning.
- **Limited Interpretability:** Though XGBoost and ensembles are powerful, they are harder to interpret than simple models.

Chapter 6: Economical, Ethic, and Contemporary Issues

6.1 Cost Estimation and Justification

The construction of the NIDS applying machine learning for network safety entails the costs of the software, hardware, and cloud services. However, the cost of the project is still low because it mainly requires open-source tools and libraries.[10]

Cost Breakdown

- **Software and Tools**: All the tools used in the project are open source and, hence, free of cost. These include Python, scikit-learn, TensorFlow, and Pandas. This reduces the cost of proprietary software licenses to a great extent.[5]
- **Cloud Services**: Although this project can be worked out on a personal machine, since the size of the TII-SSRC-23 dataset and the computational needs for the training of the machine learning model, facilities provided on the cloud may be crucial. Potential platforms include: This would lower the cost, as, for the present project, cloud services are used in pulsive ways: either through free-tier services of Google Colab or based on the pay-as-you-go model provided by AWS. Basic usage during training and experimenting phases of early-stage uses of cloud also would incur quite minimal costs at approximately $10–$50.[4]
- **Hardware Requirements**: Since most data processing and training were executed on local machines, aka laptops, there is no major hardware cost beyond the computational resources used. However, for large-scale deployments, more powerful hardware, such as servers with GPU support, may be necessary to speed up model training.[7]

6.1.2 Cost Justification

Overall, the costs are reasonable, in light of the infrequent use of paid resources and dependence on free open-source software. The rewards from the successful implementation of an intelligent NIDS with machine learning capabilities far outweigh the costs, including improved protection of the network against cyber attacks and conformity to regulatory standards. Furthermore, the effectiveness of the system will minimize erroneous positives from the normal flow of traffic, thus saving valuable resources and burdening cybersecurity teams less.

6.2 Relevant Codes of Ethics and Moral Theories

There are some ethical issues which have to be followed while deploying machine learning-based intrusion detection systems. They are:

- **Privacy and Data Protection**: It should ensure the data used inside the system will be treated as per the general legal frameworks for data protection and privacy, that is, by the General Data Protection Regulation and other data protection laws. Personal data usage in the network traffic analysis should be minimal to the minimum levels possible with no exposure of sensitive information when detection occurs.
- **Non-Discrimination and Fairness**: The best practices in machine learning ethics are such that models must not have any bias set up to lead to discriminatory or unfair outcomes. Data set the model is trained on should be balanced and representative and must avoid any specific kinds of network traffic or attack patterns.
- **Transparency and Accountability**: The machine learning models need to be transparent of their decision making process. Interpretability of models' predictions explain much importance therefore the algorithms used are interpretable for building stakeholder trust over the fact that the NIDS will act with fairness and non-ethical biases.

- **Accountability in Model Development**: One must ensure that the models used within NIDS get updated and followed for any sorts of errors or vulnerabilities that could provide loopholes for compromising the system-that is, developers and organizations have to be held accountable for the decisions made by such systems.

6.2.1 Ethical Frameworks

This work will respect professional ethics codes from the following significant organizations but is not limited to :

- **IEEE**: The IEEE Code of Ethics focuses on the good of the public and on the promotion of openness in developing new technologies.
- **ACM**: The ACM Code of Ethics includes such principles as privacy, confidentiality, and fairness in all technological solutions.

From these codes, this project will ensure ethical considerations throughout the development and deployment of the NIDS.

6.3 Ethical Dilemmas and Justification of the Proposed Solution

No serious ethical concerns are identified from the current approach. TII-SSRC-23 is the dataset on which this project's data is based, and it provides opened access, which means it can only be used in particular research environments. The whole idea of the proposed project is basically to enhance the performance that intrusion detection systems can provide, and this can be done by live traffic data, while forming a certain system that shall correctly identify threats provided in different changing conditions.

Furthermore, the system does not process any personal or sensitive data outside of network traffic patterns, so there is no ethical conflict of interest with regard to privacy. In addition, the models are nondiscriminatory against any particular group, and the implementation is designed to improve overall network security without causing any harm to the greater community.

6.4 Relevant Environmental Considerations

The highest environmental concern regarding the deployment of any intrusion detection system using machine learning is the amount of energy used when training big models on large data sets. Most of the machine learning models, particularly deep learning models, require high computation resource consumption, and most of the times, such a consumption leads to high energy consumption. However, this project mitigated environmental concerns in the following ways:

- **Efficient Algorithms**: The application of efficient algorithms, for example XGBoost and Random Forest that have been demonstrated to be computational in nature and require less environment resource compared to using deep neural networks.[4]
- **Use of Cloud Computing**: Training models with the use of cloud services such as those energy-efficient servers offers an offset in ecological footprint.
  6.4.1 Minimizing Ecological Footprint[9]

The project embraces the adoption of green computing principles, which relate to the saving of energy and optimizing computational efficiency. Minimization of environmental impacts from the project is achieved through the use of cloud computing platforms that enhance optimization of power usage and recycling hardware.

6.5 Relevance to Jordan and Region-Social, Cultural, and Political

This is one of the most relevant projects to Jordanians and the greater Middle East as the day-by-day dependency on digital infrastructure is on the rise while concerns on cybersecurity are growing. Application of machine learning for intrusion detection will help strengthen the security of critical infrastructures that need high security against cyber-attacks such as government systems, healthcare and educational institutions.[1][10]

Social Context

The privacy and data protection issue has become significant in Jordan and even the Middle East. Machine learning-based NIDS application would protect sensitive data from reaching cyber-attacks, which would subsequently ensure privacy about individuals' person information.

Cultural Considerations

The cultural importance attached to respect for privacy and the secrecy of personal details is high. Therefore, the intrusion detection system implemented in this region needs to strictly follow local culture and privacy as well as data protection legislation.

Political Landscape

Over the past years, governments in the region have established enhanced cybersecurity frameworks due to increased cyber attacks. The project falls within the ambit of national security objectives as it relates to the protection of classified information; hence, there will be compliance with the regulation on cybersecurity.

Regional Security Programs

Advanced NIDS can also be applied to regional cybersecurity initiatives. The region faces very complex cyber threats that are facing its nations. The system will enhance an organization's capabilities to defend its selves against any form of attacks while minimizing risks towards national security.

Chapter 7: Project Management

7.1 Project Schedule and Time Management

The project will be divided into four major phases. The milestones and the estimated time required for each phase are listed below:

- **Phase 1**: Select Dataset, Collection and Pre-processing (8 Weeks)
- **Phase 2:** Model Building and Training (4 Weeks)
- **Phase 3:** Testing and Evaluation (1 Week)
- **Phase 4:** Result and Presentation Work (1 week)

7.2 Resource and Cost Management

Resources required for the project are very few and most of the tools required are open source as well. The following are the key resources used:

Human Resources:

- **The student:** Mohammed Rahhal, who will carry out the job of data analysis, training of the models, and writing of the report.

Software Resources:

- Python with libraries like scikit-learn, Pandas, TensorFlow, and XGBoost for data processing and development of models.
- Google Colab or computation resource on a local machine to run the models.
  Cloud Resources:

Cost Estimate

- **Software**: Completely free to use because of open source libraries.
- **Cloud Computing**: Very low (Minor Operations, $10 - $50)
- **Hardware:** Computer / Laptop already available

7.3 Quality Management

Following points are done for high quality output of the project

- **Model Evaluation:** Use appropriate metrics, including accuracy, precision, recall, and F1-score, to assess. Do the cross-validation of generality assessment for the models.
- **Iterative Improvement**: From the results of the evaluation, the models are to be changed and hyperparameter tuning done for improvement.
- **Documentation**: Adequate documentation of each stage of the project. This should ensure transparency and reusability of the results obtained.

7.4 Risk Management

There are very few risks associated with this project because the project is mainly research-oriented and involves experiments with permitted datasets. The main risks involved are:

- **Data Imbalance:** This problem is minimized through techniques used in oversampling and undersampling to balance the dataset.
- **Model Overfitting** : Cross-validation and hyperparameter tuning will help to avoid overfitting and ensure the model performs well on new, unseen data.

- **Resource Constraints:** These can be expensive if heavy computation is needed, which computational cloud resources are. Using an efficient algorithm and using the free usage of cloud services such as Google Colab overcomes these.

7.5 Project Procurement

There are no major procurement requirements for this project. The major resources used are free and open-source software and existing computing hardware. If more resources are needed, it will be through cloud computing on an as-needed basis.

Chapter 8: Conclusion and Future Work

The primary contribution of this work is the effective integration of various machine learning algorithms with a traditional NIDS to improve the detection and classification of malicious network activities. By training on the TII-SSRC-23 dataset, the resulting system demonstrated significant performance improvements in both accuracy and efficiency compared to traditional signature-based methods.

8.1 Algorithmic Performance Evaluation

Each algorithm was evaluated using standard metrics—accuracy, precision, recall, and F1-score. Below is a summary table comparing the overall accuracy of each model:

Algorithm	Accuracy
Logistic Regression	0.94
Decision Tree	0.99
Random Forest	0.98
Gradient Boosting	0.97
K-Nearest Neighbors (KNN)	0.98
Support Vector Machine	0.95
XGBoost	0.99

8.2 Further Future Work to Enhance the Solution/System.

While the system shows considerable success, the following areas for future improvement have been identified:

- **Optimization of Algorithm**: Further optimization of hyperparameters and trying more complex or hybrid models may result in further performance improvements.
- **Real-Time Integration**: Incorporation of real-time data streaming and detection mechanisms is likely to enhance the working applicability of the proposed system in live network environments.
- **Expanded Datasets:** Inclusion of more datasets with diverse network traffic patterns will contribute to enhancing the model's robustness and generalizability.
- **Automated Model Selection** :The development of an automated framework for model selection and parameter tuning could ease performance improvements and future updates.

8.3 Lessons Learned

- **Data quality is key**: The success of the project relied heavily on the quality and relevance of the training dataset. Ensuring diverse and representative data sources is critical to training robust intrusion detection models.
- **Algorithm selection**: Choosing the right algorithm based on the specific characteristics of the data set and operational requirements greatly affects detection accuracy and efficiency.

References

1. J. Kaur and N. Singh, "Intrusion Detection Systems: A Review of Techniques, Challenges, and Research Directions," *IEEE Access*, vol. 8, pp. 219388–219405, 2020. doi: 10.1109/ACCESS.2020.3041031.

2. N. Sharma and S. Kalra, "Machine Learning Techniques for Intrusion Detection in Networks: A Comprehensive Review," *Journal of Network and Computer Applications*, vol. 205, p. 103442, 2022. doi: 10.1016/j.jnca.2022.103442.

3. Khadraoui and M. F. Zhani, "TII-SSRC-23 Dataset: Addressing Class Imbalance in Network Intrusion Detection," *arXiv preprint*. Available: https://arxiv.org/abs/2310.10661.

4. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.

5. Z. Zhou and M. Li, "The Role of Data Preprocessing in Machine Learning-Based Intrusion Detection Systems," *Cybersecurity Journal*, vol. 6, no. 1, pp. 12–28, 2020. doi: 10.1186/s42400-021-00103-8.

6. GeeksforGeeks, "Intrusion Detection System Using Machine Learning Algorithms," Available: https://www.geeksforgeeks.org/intrusion-detection-system-using-machine-learning-algorithms.

7. S. Tariq and I. Khan, "Comparison of Signature-Based and Anomaly-Based Intrusion Detection Systems," *Springer International Conference on Cybersecurity*, vol. 12, no. 3, pp. 50–65, 2021. doi: 10.1007/s11620-020-0219-x.

8. S. Shen and J. Wang, "Feature Normalization and Label Transformation in Network Security Datasets," *ACM Transactions on Data Science*, vol. 3, no. 4, pp. 23–41, 2022. doi: 10.1145/3500385.

9. S. Wang, Y. Zhao, and J. Zhang, "An Overview of the Role of Machine Learning in Intrusion Detection Systems," *Proceedings of the 2021 IEEE International Conference on Cybersecurity*, pp. 199–207, 2021. doi: 10.1109/ICCS51204.2021.00046.

10. M. S. Ali, R. N. M. Ali, and M. R. M. Shamsuddin, "A Review on Hybrid Intrusion Detection System (IDS) Using Machine Learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1232047, 2022. doi: 10.1155/2022/1232047.

11. Market Data Forecast, "Intrusion Detection and Prevention Systems Market," Available: https://www.marketdataforecast.com/market-reports/intrusion-detection-prevention-systems-market.