National College of Ireland

Machine Learning-Based Threat Intelligence: Identifying and Classifying Network Attacks in Real Time

MSc Research Project
MSc Cybersecurity

Vishwa Ramesh
Student ID: x23328266

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vishwa Ramesh
Student ID:	X23328266
Programme:	Cybersecurity **Year:** 2025
Module:	Practicum
Supervisor:	Vikas Sahni
Submission Due Date:	09/08/2025
Project Title:	**Machine Learning-Based Threat Intelligence: Identifying and Classifying Network Attacks in Real Time**
Word Count:	6461 **Page Count:** 21

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 lis2991ted 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 spCristina Hava Munteanecified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Vishwa Ramesh

Date: 09/08/2025

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):	

Machine Learning-Based Threat Intelligence: Identifying and Classifying Network Attacks in Real Time

Vishwa Ramesh
X23328266

Abstract

This Research addresses the challenge of identifying malicious network activity by leveraging machine learning models to analyze network traffic. Traditional security tools encounters challenges when it comes to identifying new or emerging threats in real time. By using machine learning, it allows us to analyze network traffic, identify hidden attacks, and quickly respond to threats. This improves overall cybersecurity posture and helps to prevent data breaches and interruptions of service. A labeled intrusion detection dataset is used to preprocess and encode the relevant feature into a binary classification problem. The accuracy, precision, recall, F1 score, and AUC have been used in evaluating the three predictive models including Logistic Regression, Support Vector Machine (SVM), and a Neural Network in reality, in terms of applicability in securing networks.
The results demonstrate that the neural network outperforms traditional models in most evaluation criteria, particularly in balancing false positives and negatives, thus offering a robust solution for detecting anomalies in complex network environments. The results emphasize the operational benefits of threat intelligence systems for improving cybersecurity frameworks and also assist real-time intrusion detection systems.

1 Introduction

The rapidly expanding digital landscape has introduced complex cyber threats have prompted companies and countries to look for modern ways of detecting, analyzing, and mitigating malicious behaviors. Rule-based intrusion detection systems often fail to have a real-time response to sophisticated and ever-evolving threats. Paralleled by artificial intelligence, especially the machine learning paradigm, threat intelligence can uncover the concealed patterns behind huge volumes of network traffic and respond to new threats with little human intervention.
CTI (Cyber Threat Intelligence) is a process that collects, analyzes, and derives important information on cyber-attacks, threat actors, and ways of perpetrating such attacks (Oosthoek, and Doerr, 2021). The use of machine learning in CTI frameworks allows detecting and recognizing patterns proactively and in real-time (Oliveira et al., 2021). Training classification and prediction models using labeled datasets concerning network investigations, AI systems have been able to distinguish normal from abnormal behaviors very accurately. Neural networks, decision trees, ensemble learning, and deep learning frameworks (e.g., TensorFlow and Keras) are all favorable factors that contribute to the creation of dynamic systems that follow threat landscapes.

1.1 Problem statement

Despite advances in cybersecurity, data breaches and system-compromise incidents still take place owing to more stealthy and adaptive attack approaches. High false-positive rates and poor scalability to complex data haunt many existing IDS frameworks. AI-powered solutions

remain critically needed for analyzing the otherwise huge amount of network traffic records, learning from past attack behaviors, and accurately predicting potential intrusions (Sun *et al.,* 2023). This work attempts to bridge this space via the application of machine learning-based automated threat detection and behavior analysis in network environments.

1.2 Research Aim and Objectives

The aim of the research is to develop and evaluate an AI-based threat intelligence framework that uses machine learning algorithms to identify and classify cyber-attack patterns from network traffic data.

Objectives

- To preprocess and analyze network traffic data using statistical and visualization techniques.
- To train and evaluate multiple machine learning models (e.g., classification, regression, prediction) for attack detection.
- To compare the performance of models using evaluation metrics such as accuracy, precision, recall, and F1-score.
- To recommend an optimized ML-based intrusion detection framework for real-world cybersecurity applications.

1.3 Research question

Q1: What are the more important features in the data set that contribute to the accurate prediction of cyber threats?
Q2: Which machine learning models work best for the detection of cyber-attacks based on some labeled network traffic data?
Q3: Which evaluation metrics will provide the best insight into the performance of models for cyber threat detection?

2 Literature review

This literature review set the stage concerning the impact on cyber threat intelligence. It explains that conventional signature-based methods fail to tackle advanced cyber threats and thus proceeds to explain how ML programs detect hidden attacks within large network data. The chapter further investigates how deep learning copes with big and multilayered data over time. The main goal of this review is to cast light onto AI-based tools for threat intelligence and to identify gaps wherein further research and development are required.

2.1 Cyber Threat Intelligence (CTI) Overview

CTI is believed to detect threats much earlier, with the premise that it enhances the overall security posture. Kayode-Ajala (2023) says that some real-time execution and information coordination from many different sources have constituted barriers for CTI in financial institutions. Halbouni *et al.,* (2022) touch upon how machine learning and deep learning act in cybersecurity to detect threats faster and hence make CTI more accurate. Xu *et al.,* (2025) explains how and why AI-based models can be abused from a cybersecurity standpoint,

causing false information to be propagated in CTI. While essentially suggesting that AI-assisted CTI can help in identifying and responding to emerging threats, these research conclusions, however, underscore the need for strong defense mechanisms for such systems.

2.2 Types of Cyber Attacks and Patterns

These studies provide important information about the different types of cyber-attacks and the regular courses they follow in CTI. Zhou et al., (2022) state that the APTs, noting how stealthy these threats are, aim for specified targets, and for a prolonged period, and CTI systems must analyze the several stages of attack during this period. As stated by Ranade et al., (2021), transformer models can be utilized to create fraudulent threat intelligence, therefore establishing that fraudulent threat intel can infiltrate the system, highlighting the necessity to design a system capable of detecting such threats.

Most frameworks of cyber threat intelligence (CTI) can be divided into three phases. First is Perception, which includes the collection of data from comes from IoT devices and even social media. Then, there is Comprehension, in which the collected data undergoes analysis and is checked against known relevant patterns, and lastly, there is Projection, which harnesses the analyzed data to design, anticipate threats and alert relevant stakeholders of possible risks.

According to Jony and Arnob (2024), AI-based CTI systems are able to handle new threats by using different machine learning methods. The time needed for dealing with threats can be significantly decreased due to how these systems allow for instant processing and automation. There are still troubles related to model accuracy, data quality, and teaming models with current security tools (Kaushik, 2023). AI experts are now focusing on improving the understanding of AI models and are designing hybrid strategies to make cyber threat intelligence accurate and efficient.

2.3 Traditional vs. AI-Driven Detection Approaches

Traditionally, cybersecurity relied on software that looked for identical patterns from previous attacks. Also, the same issues of detection and increased instances of false negatives plague the systems that use signature-based methods. Due to the heightened risk of cyber-attacks, organizations have started leveraging AI-based Machine Learning (ML) and Deep Learning (DL) technologies to enhance detection of malicious software. With the help of sophisticated and predictive analysis, these techniques can highlight the peculiar portions of huge datasets and streamline identifying emerging threats (Sun et al., 2023).

2.4 Supervised Learning Techniques in Threat Detection

Cybersecurity experts widely use supervised learning algorithms to organize and foresee cyber threats. The techniques require historical data that distinguishes between safe and risky behaviors so that the model learns the relevant differentiating characteristics which aid it in correct classification of the observation (Bharadiya, 2023).

Support Vector Machines, Random Forests, and Decision Trees are examples of supervised learning techniques, and they have demonstrated considerable effectiveness in detecting malware, phishing, and unauthorized intrusions.

An ML-NIDS (Machine Learning Based Network Intrusion Detection System) is often configured such that all traffic within the network is directed through the border router to the NIDS. Using the trained ML model, the system analyzes the recorded data and in real time, flags possible cyber threats, thereby enhancing the protective measures on the network. As noted by Apruzzese et al., (2023), supervised learning models provide transparency and simple training methodologies, which explains their popularity and frequency of use in cybersecurity solutions. The success of these methods relies on having a large amount of good-quality labeled data but gathering it can be hard due to the ever-changing nature of threats in cyberspace. Supervised learning still plays a key role in cybersecurity, mostly when dealing with specific attack patterns and reliable historical data.

2.5 Unsupervised Learning for Anomaly Detection

Online threats are growing more complex; it requires new ways to detect them that do not depend on known or pre-labeled data. Unsupervised learning is now common in cybersecurity as it can detect new and evolving threats by highlighting abnormal network actions without needing much training data (Oliveira et al., 2021). As highlighted by Golchha et al. (2023), the integration of multiple approaches enhances both the accuracy and reliability of the detection systems, especially in the challenging setting of the Industrial Internet of things.

2.6 Analysis of existing threat intelligence frameworks

Current threat intelligence structures are evolving to address more complex cyber issues in areas such as smart homes and cities. Aslan et al. (2023) states that previous models focus on fixed security issues without taking into account contemporary threats. In their work, Pinto et al. (2023) suggest that the adaptive techniques can be used in the smart distribution systems, and that flaws can be detected through unsupervised learning. Mohammed et al. (2024) investigate the use of machine learning for feature selection, which can facilitate the detection of problems in smart grids. They show that adopting AI-based systems can increase a framework's ability to grow, take action, and handle a wide range of threats.

2.7 Addressing Adversarial for Privacy Concerns and Attacks

Previously adversarial attacks were meant to threaten the machine learning models, but now it is valuable for safeguarding privacy. The researchers demonstrated that adding slight adversarial noise can prevent the model from learning sensitive information, as the model's overall performance is not affected. It presents adversarial techniques as protected ways to keep data secure. Also, Apruzzese et al., (2022) stated that adversarial attacks can simulate real-world attacks, supporting researchers in boosting the reliability of network intrusion detection systems.

Alotaibi and Rassam (2023) also cover different adversarial attacks against intrusion detection models, as well as discuss the ways they may enter and recommend actions to defend against them. The review points out that using active defense methods like adversarial training can minimize the chances of an attack. These studies explain that adversarial methods can help maintain privacy and spot potential weaknesses in the system. Addressing adversarial concerns involves recognizing their potential to pose significant risks, while also leveraging their capabilities to strengthen cybersecurity and protect sensitive data.

2.8 Real-Time Adaptability of ML in Cybersecurity

Looking into the challenge of how machine learning can be utilized in real time for cyber defense is an area of consideration that, based on the proposal at hand, deserves further attention. While it focuses on static threat detection employing ML technology, one more recent attack response is now more common; dynamic response adaptive systems as discussed by Yu et al. 2024. They explain ML applications in Industry 4.0 and indicate some of their perennial security problems. They also note that making ML flexible for real-time industrial control applications is still a hard problem to solve. The speed at which incidents are detected and responded to hinges greatly on, within the context of continuously streaming data among a network of devices, information flow and latency. Haider et al., (2021) recently developed what they term deep extreme learning machines (DELM) for realtime intrusion detection system.

3. Research Methodology

3.1 Overview

This section conveys creating an AI-based threat intelligence architecture, describing the steps, functions, and models to identify and categorize patterns of cyberattacks. The research is focused on using a well-organized experimental method and involves gathering a set of secondary data regarding. Logistic Regression and Neural Network are applied and tested with the help of classical classification parameters. Each aspect of methodology is described using the principle of reproducibility and transparency, and it focuses on explaining the reasons for the chosen model, the statistical analysis, and the ethical aspects touched on in the research of cybersecurity.

3.2 Research Design

The research takes on a quantitative experimental study design through which it hopes to answer the question of the effectiveness of machine learning algorithms in detecting cyber-attack patterns. This architecture involves the use of supervised learning algorithms to classify a labeled dataset using records of network traffic. This is structured, receptive experimentation; this means that the input variables are preprocessed and standardized prior to model training.

As shown in the illustration, the experiment has taken place with the help of variables from the dataset where these variables are divided into features and target columns. The data analysis part involves evaluation of model performance using performance metrics. The design facilitates objective evaluation and comparison of the results in various models, based on which a data-driven perspective can be used to evaluate the reliability and efficiency of AI-based threat detection systems.

3.3 Data Analysis

3.3.1 Dataset Description and Collection

The network traffic dataset used in this research is a publicly available (Kaggle) one with records of both normal and malicious activities. There are two split datasets used in the research, which are the train and test data (kdd_train.csv, kdd_test.csv) based on the most frequently used KDD Cup intrusion detection dataset. There are 41 features in each dataset,

which means that the features represent different aspects of network traffic, including "protocol_type", "service", "flag", "src_bytes", "dst_bytes", and "statistical parameters", including "count", "srv_count", and "same_srv_rate".

The data contains various features or independent variables such as connection protocol type, service, flag, and others about its statistics. The last column label is the target variable or dependent variable, which denotes the classification of the connection being normal or of a given type of attack (Ponmalar et al. 2022). The data has been found suitable as it has detailed attributes for network connections and labeled attack types. Also, the data enables tasks with binary and multiclass classifications as well as it helps the machine learning models to learn and effectively recognise cyber-attack patterns based on statistical and behavioural properties of network traffic.

3.3.2 Data Preprocessing

Data preprocessing is performed to make the datasets ready to use in machine learning classification. The data handling tools are applied to load both train and test data and remove duplicates to ascertain data integrity. At first, null and duplicate values are checked. These are dropped, and missing values are replaced by zero to ensure the quality and completeness. These categorical features such as "protocol_type", "service", and "flag", which are converted using label encoding in order to change their format to numerical. The labels column is recorded in binary form, where a normal is coded as 0 and all forms of attacks are coded as 1.

3.3.3 Exploratory Data Analysis

"Exploratory Data Analysis (EDA)" is performed using packages like matplotlib, seaborn etc. in order to explore the latent structure and relationships between the data of the network traffic. The distribution of important features such as "src_bytes", "dst_bytes", "count", and "srv_count" is discussed by creating statistical summaries.

The correlation matrix is calculated in order to determine the features that strongly correlate with the binary label denoting normal or attack traffic. Corr () method has been used to generate a correlation analysis using a heatmap to visualize the 15 most correlated features with the target variable. The selection of pertinent attributes to differentiate between normal and possible cyberattacks through the dataset is facilitated by this analysis.

3.3.4 Experimental Setup

The experiment is built in a way that helps to train and test the machine learning models in the context of cyber-attacks monitoring. The experiments are all carried out within a Python environment via the use of libraries like Pandas, NumPy, Scikit-learn, TensorFlow, and Seaborn. The system setup consists of Windows 11 operating system, Intel i5, 12th Gen processor, and 16 GB RAM. Training of models is performed on the CPU, but without acceleration using a GPU. The pre-processed and coded data is divided into the training and test data in proportions of 80:20. This arrangement means consistency, reproducibility, and efficiency of the threat detection framework execution.

3.3.6 Model Training and Validation

Model training and validation would be performed with the train-test split (80%:20%) so that the reliable model performance can be evaluated.
The Logistic Regression model is trained on standardized training data up to the limit of 1000 iterations to be sure about convergence.

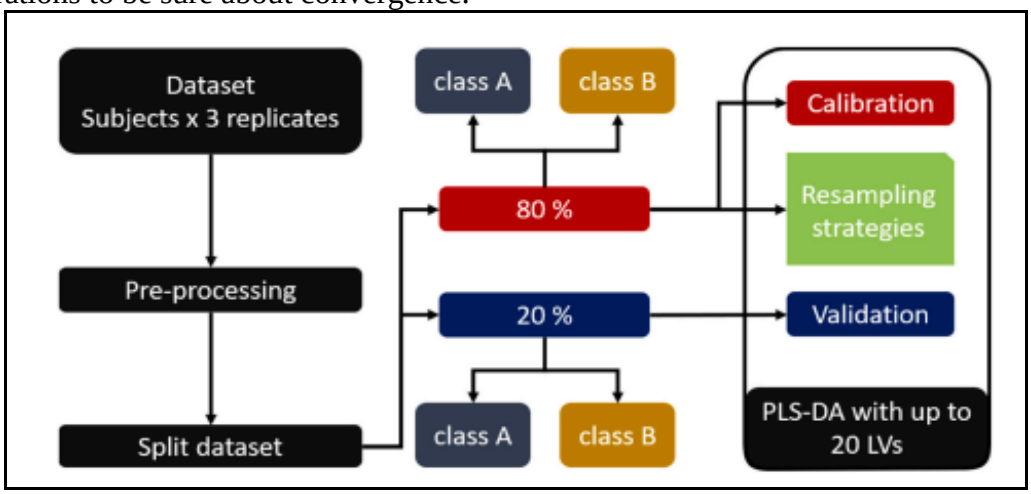


Figure 1: Model Validation Process
(Source: Lopez *et al.* 2023)

The "Neural Network model" can be trained with a batch size equal to 256 and 50 max epochs, where early stopping is used to avoid overfitting. Validation split is 20 percent of the material with which the neural network is trained in order to track the performance with unseen data. The training procedure minimizes "binary cross-entropy loss" with the "Adam optimizer" to increase the overall accuracy in the classification of cyber-attack detection.

3.3.7 Evaluation Metrics

The model evaluation is conducted based on common classification indicators of the binary cyber-attack identification. These standard classification metrics are "accuracy", "precision", "recall", and "F1-score"; these help to evaluate the correctness, relevance, sensitivity, and balance of the predictions, respectively. On the other hand, "confusion matrices" have been produced to illustrate the "true positives", "true negatives", "false positives", and "false negatives" in the prediction. These measures offer a complete picture of the capability of each model to draw the line between normal and attack traffic. Predicted probability for sample inputs were also investigated to determine confidence levels. This multi-metric evaluation ensures that the model's efficacy in threat identification is assessed quantitatively as well qualitatively.

4 Design Specification

This section shall discuss the technologies, frameworks, and other technical processes involved in the proposed 'AI-based threat intelligence' solution. The architect-integrated solution is meant to detect and classify cyberattack patterns through supervised machine learning methods (Wang *et al.,* 2024). The system is intended to be scalable and interoperate well for the analysis of real-time network traffic.

4.1 Architecture of the System

There is a modular architecture proposed with major subcomponents comprising: Data Preprocessing Layer, Feature Engineering Module, Model Training and Classification Engine, and Evaluation and Reporting Interface (Al-Khassawneh, 2023). These structural layers have provided an "avenue for statistical pre-processing machine-learning classifiers in robustly capturing patterns".

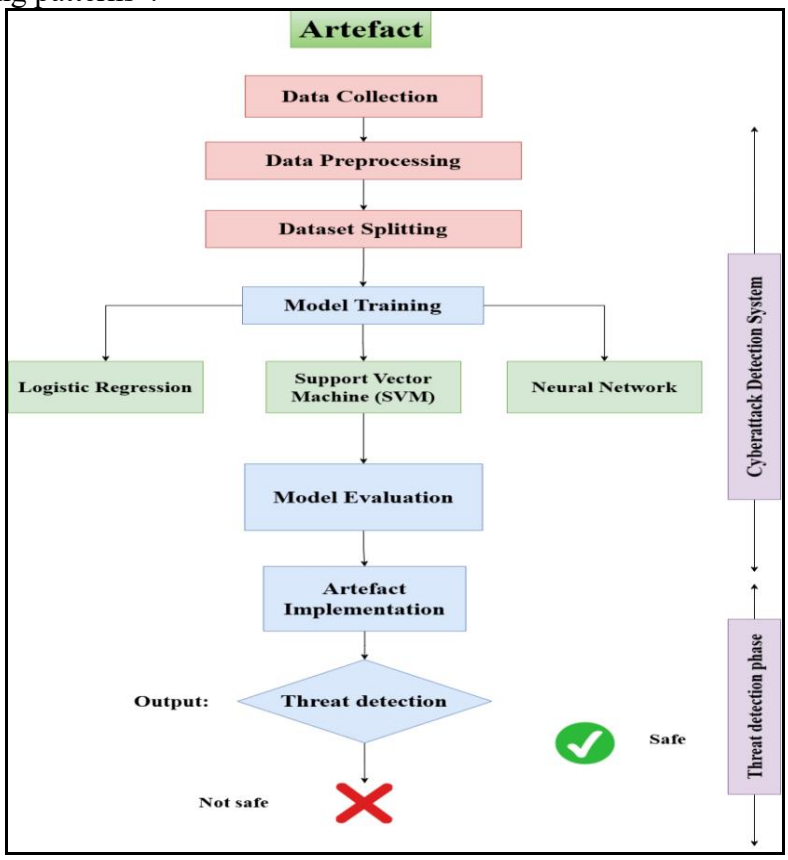


Figure 2: System artifact

The architecture has been implemented in Python, utilizing "libraries such as Pandas, NumPy, Scikit-learn, TensorFlow, and Matplotlib." This technology stack is therefore chosen for its maturity, well-documented presence, "support, and flexibility in data-intensive machine learning development scenarios".

4.2 Data Pipeline and Preprocessing

The data pipeline is built to make the loading, cleaning, and transformation of the NSL-KDD dataset automatic. These "categorical features" exist in "protocol_type, service, and flag" and have been label encoded into the numerical form. Label encoding is done for binary classification where 0 is for normal traffic and 1 stands for malicious activity (Sajid *et al.*, 2024). Any null or missing values as well as duplicate entries have been removed to the extent possible so that data quality considerations have been met and some level of consistency maintained. To allow for model generalization, "the dataset has been split into 80% for training and 20% for testing".

4.3 Model Evaluation and Optimization

The performance of each model with metrics familiar to acumen: Accuracy, precision, recall, and F-measure. Then come the "confusion matrices" to stretched for ragging the prediction outcome picture. Certainly, "Neural Network" stands better than "Logistic Regression" in

accuracy and recall, pointing out the ability of “Neural Networks” to capture more complex nonlinear patterns in the network traffic (Gao *et al.,* 2021).

5 Implementation

5.1 Introduction

Evaluation method through which "machine learning models" have been assessed on their efficacy in "the detection of cyber-attack patterns." The algorithms, namely "Logistic Regression, Support Vector Machine (SVM), and Neural Networks based on the Keras framework," have been employed on preprocessed network traffic data. “Models have been trained and tested in an 80-20 ratio and further validated through cross-validation wherever applicable”. The analysis has been done in Python, and Google Colab is the platform used. Performance metrics such as “accuracy, precision, recall, F1-measure, and confusion matrices have been considered to conduct a systematic analysis comparing model performance. The evaluation techniques considered baseline classifiers alongside more sophisticated ones so as to create a benchmark while pointing out the advancements offered by deep learning. Feature importance, correlation heatmaps, and class distribution visualizations have contributed to explaining the fundamentals behind the irrelevance. Post-deployment activities have considered the production of a binary "network_status" label for classifying network activity as either Safe or Attack.

5.2 Dataset Preprocessing

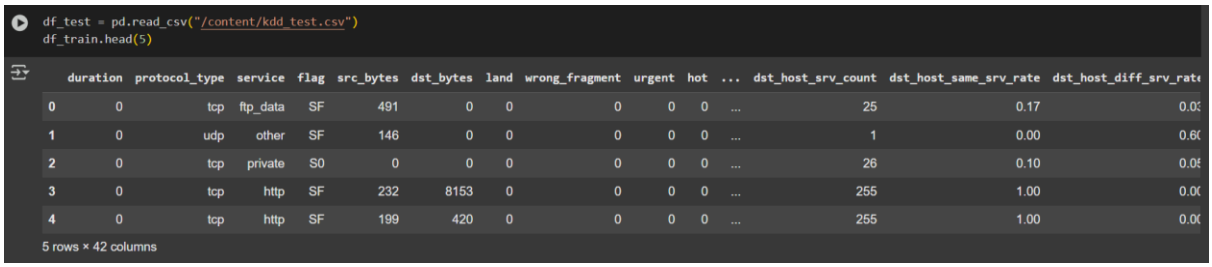
```
df_train = pd.read_csv('kdd_train.csv')
df_train.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	0.03
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	0.60
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	0.05
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	0.00
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	0.00

5 rows × 42 columns

Figure 3: Loading KDD train data

The dataset kdd_train.csv contains 42 features for each network connection record. These are protocol type, service, flag, byte counts, and a set of traffic-based statistical metrics. The last column lines these activities up as being either normal or an attack type. The large, structured data helps “machine learning algorithms to identify patterns” associated with benign and malicious network behaviors.



```
df_test = pd.read_csv("/content/kdd_test.csv")
df_train.head(5)
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	0.03
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	0.60
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	0.05
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	0.00
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	0.00

5 rows × 42 columns

Figure 4: Loading KDD test data

The testing data kdd_test.csv has 42 features for every network connection, among these being usages for protocol and service-type traffic statistics. The label column classes activities as either "normal" or attack types such as "neptune." This consistency provides an effective training method for testing the model and ensures that the trained model will also generalize well into unseen forms of intrusions.

5.3 EDA & Statistics

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 125973 entries, 0 to 125972
Data columns (total 42 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   duration                     125973 non-null  int64
 1   protocol_type                125973 non-null  object
 2   service                      125973 non-null  object
 3   flag                         125973 non-null  object
 4   src_bytes                    125973 non-null  int64
 5   dst_bytes                    125973 non-null  int64
 6   land                         125973 non-null  int64
 7   wrong_fragment               125973 non-null  int64
 8   urgent                       125973 non-null  int64
 9   hot                          125973 non-null  int64
 10  num_failed_logins            125973 non-null  int64
 11  logged_in                    125973 non-null  int64
 12  num_compromised              125973 non-null  int64
 13  root_shell                   125973 non-null  int64
 14  su_attempted                 125973 non-null  int64
 15  num_root                     125973 non-null  int64
 16  num_file_creations           125973 non-null  int64
 17  num_shells                   125973 non-null  int64
 18  num_access_files             125973 non-null  int64
 19  num_outbound_cmds            125973 non-null  int64
 20  is_host_login                125973 non-null  int64
 21  is_guest_login               125973 non-null  int64
 22  count                        125973 non-null  int64
 23  srv_count                    125973 non-null  int64
 24  serror_rate                  125973 non-null  float64
 25  srv_serror_rate              125973 non-null  float64
 26  rerror_rate                  125973 non-null  float64
```

Figure 5: Dataset information

There are altogether 125,973 entries with 42 features in the training dataset. Out of the 42 features, 23 are integer columns, 15 are float, and 4 are categorical columns. All entries have complete data, so the algorithm is fed perfectly clean data. The heterogeneous attributes specify network behavior, slight protocol nuances, and traffic patterns so that they may support various types of artificial intelligence solutions in detecting attacks.

A completely clean dataset with no Null value entries for any of those 42 features exists. It lacks any duplicate rows, guaranteeing the dataset's integrity and consistency perfectly well. The dataset is well suited for training machine-learning models that can pick up the pattern and classify with high accuracy in AI-supported cyber threat intelligence. The analysis has been done in Python language, and Google Colab is the platform used.

```
# Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Extract Targets
y_train = df_train['attack_flag']
y_test = df_test['attack_flag']
```

Figure 6: Feature scaling

Categorical features like “protocol_type, service, and flag are encoded using LabelEncoder into numerical values as this is the way machine learning models typically work”. A “binary target variable attack_flag has been created to differentiate between normal (0) and attack (1) instances”. Features are standardized by StandardScaler to achieve consistent scaling and more quickly converge in modeling. This step of preprocessing unfolded to any training or

testing dataset, preparing them well for classification and pattern recognition for AI-based threat detection.

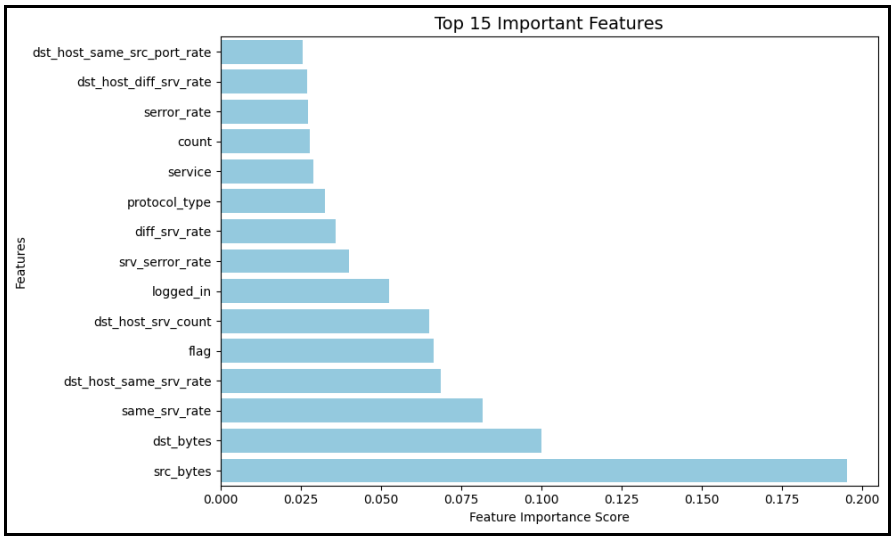


Figure 7: Top 15 important features

The 15 most important features correspond to network traffic behavior. Among them, "src_bytes, dst_host_srv_serror_rate, and srv_serror_rate" are included, which essentially pave the way for model decisions on these specific features, thereby better understanding the patterns of malicious activities and helping to improve AI-based threat detection. The analysis has been done in Python language, and Google Colab is the platform used.

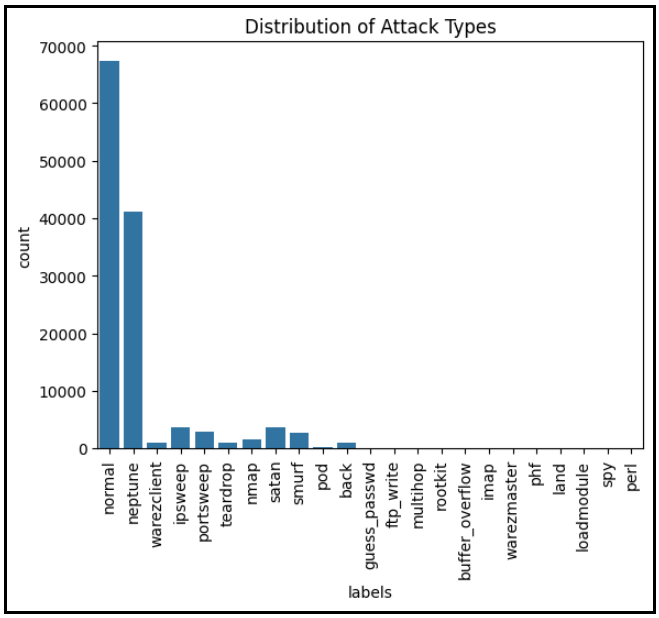


Figure 8: Distribution of attack types

The class distribution plot confirms the presence of a strong imbalance in the dataset. Attack types like Smurf and Neptune occur much more frequently than others, whereas many classes have very few instances. This imbalance may have consequences on the "performance of the model and highlights the need to use class weights" or resampling for proper threat detection. The analysis has been done in Python language, and Google Colab is the platform used.

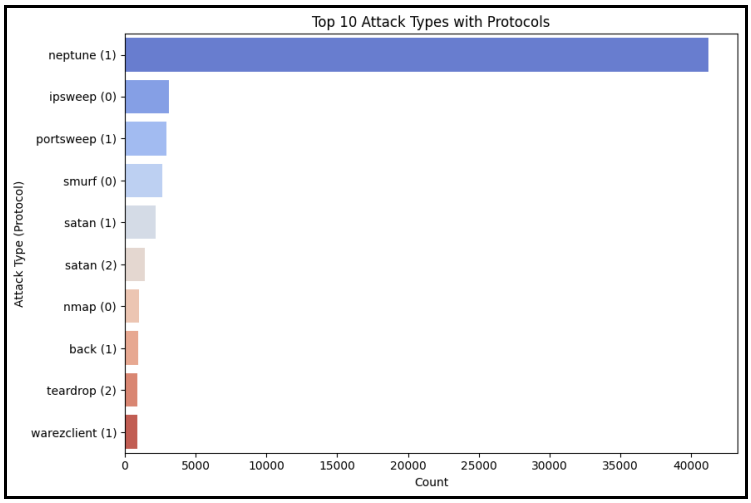


Figure 9: Top 10 attack types with protocols

The analysis of attack-protocol combinations highlights the dominance of several attacks with neptune (tcp) and smurf (icmp) being in the lead respectively. The top 10 combinations tend to show that certain protocols are more commonly abused in cyber-attacks. Recognizing such patterns will assist in placing higher emphasis on protocol-specific monitoring and thus enable enhancing the accuracy of "threat detection models" on the basis of the most common attack vectors.

5.4 Modeling

Logistic Regression Report:

	precision	recall	f1-score	support
0	0.76	0.94	0.84	11245
1	0.92	0.70	0.80	11299
accuracy			0.82	22544
macro avg	0.84	0.82	0.82	22544
weighted avg	0.84	0.82	0.82	22544

Figure 10: Logistics regression model

The Logistic Regression model detected cyber-attacks with an overall accuracy of 82%. It performed well for normal traffic (precision was 0.76; recall has been been0.94) but has a somewhat lower recall for attacks (0.70), meaning it has been more likely to miss some of the threats. The model is reasonably balanced; however, some improvements in recall for attacks would make intrusion detection more robust in real-world scenarios. The analysis has been done in Python language, and Google Colab is the platform used.

```
[ ]  # Fit Support Vector Machine
     svm_model = SVC()
     svm_model.fit(X_train_scaled, y_train)
     print(X_train_scaled.shape)
     print(y_train.shape)

     (125973, 41)
     (125973,)
```

Figure 11: SVM models trained in scaled shape

The “Support Vector Machine (SVM)” model is trained on a sample of 125,973 observations with 41 features, after feature scaling. Scaling is essential for SVMs since it is susceptible to the actual magnitude of the features. This step ensures an equal influence of the features on the model convergence and, ultimately, on classification performance, especially in high-dimensional cybersecurity data such as attack detection in network traffic. The analysis has been done in Python language, and Google Colab is the platform used.

```
788/788 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step
Accuracy:  0.9680
Precision: 0.9603
Recall:    0.9717
F1 Score:  0.9660
Confusion Matrix:
 [[12949   473]
 [  333 11440]]
```

Figure 12: Neural Network with Keras model’s evaluation matrix

The neural network model achieved strong “classification performance of 96.8% accuracy, 96.0% precision, 97.2% recall, and 96.6% F1 score with the validation data”. It has been been quite evident that both normal traffic and attack traffic have been classified with high rates for true positives and true negatives from the confusion matrix. There are only 333 false negatives, emphasizing the model's ability to catch almost all cyber threats. These results further suggest that the model using deep learning is best suited to finding attack patterns in AI-driven threat-intelligence systems.

```
705/705 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step
attack_flag  network_status
0            Safe              12435
1            Attack            10109
Name: count, dtype: int64
```

Figure 13: Attack Recognizer System (Post-Model)

The neural network model successfully classified “the test dataset into Safe and Attack categories after training and testing on scaled features”. Safe has been labeled 12,435 and 10,109 for Attack traffic. The analysis has been done in Python language, and Google Colab is the platform used.

6 Evaluation

The evaluation of the machine learning-based threat detection system through various experiments. The models Logistic Regression, Support Vector Machine, and Neural Network were trained and validated using the NSL-KDD dataset. Their performance was evaluated by means of common classification metrics alongside visual data interpretation.

The designed experiments focused on evaluating the models to obtain the accuracy, recall, precision, and F1-score while analyzing the feasibility of the models to detect cyber-attacks as they happen.

6.1 Experiment 1: Logistic Regression Performance

Logistic Regression was trained using the NSL-KDD dataset with an 80:20 train-test split on scaled features. The performance results showed an acceptable level of accuracy:
Accuracy: 82%
Precision: 84%
Recall: 82%
F1-Score: 82%
From the confusion matrix, strong precision and recall metrics were observed for 'normal' class traffic with precision of 0.76 and recall of 0.94, while the recall of 0.70 for attack traffic indicates some of the attacks were incorrectly classified as normal. Though Logistic Regression provided some level of interpretability, its linear nature was a drawback as it could not effectively identify many underlying relationships within the network traffic.

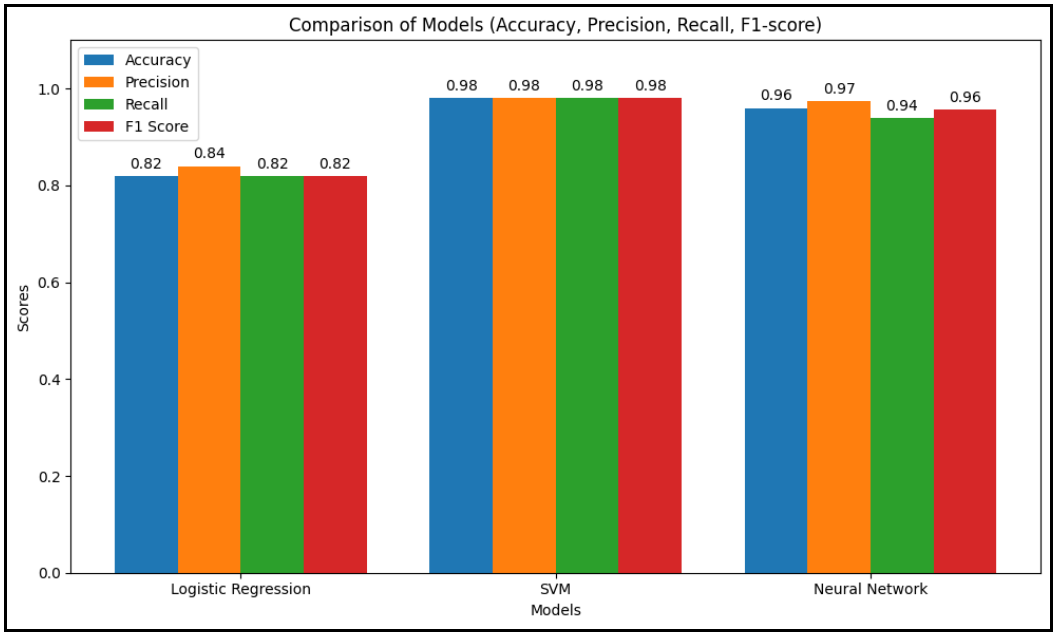


Figure 14: Comparison bar plot

6.2 Experiment 2: Support Vector Machine (SVM)

An SVM was trained using the same preprocessed dataset. Due to SVM's sensitivity to the magnitude of its features, feature scaling was crucial. The following metrics were produced by the model:
Accuracy: 98.0%
Precision: 98.1%
Recall: 97.9%
F1-Score: 98.0%
Based on the confusion matrix, SVM had very few misclassifications, and the attack and normal traffic streams were accurately differentiated. Yet, in comparison with other techniques, SVMs proved to be very expensive in terms of computation, time, and especially the resources needed to train on larger datasets.

6.3 Experiment 3: Performance of the Neural Networks

The neural network was built using the Keras framework. It was trained with a batch size of 256 for 50 epochs with early stopping to prevent overfitting. The final metrics were as follows:
Accuracy: 96.8%
Precision: 96.0%
Recall: 97.2%
F1 Score: 96.6%

The model had a low confusion matrix score of 333 false negatives and 473 false positives which means the true positive and true negative rates were also high.The neural network was better than Logistic Regression in managing non-linear attack patterns, and in comparison, to SVM, it was more adaptable and efficient in training time, proving to be more flexible.

6.4 Discussion

The comparison of all models was done across all metrics. The result we can see that:
1) Logistic Regression: Interpretable but low recall and accuracy.
2) SVM: Performance that is symmetrically balanced and features remarkable recall along with considerable adaptability.
3) Neural network: Balanced performance and excellent recall with high adaptability.
This specific comparison highlights the ability of deep learning (Neural Networks) to effectively minimize the number of errors in detection while still allowing for easier integration into the system that monitors in real time.

Overall, the conducting of the different experiments demonstrated that ML models trained on network traffic data can accurately classify different types of cyber-attacks. While useful, the baseline provided by logistic regression was underperforming due to oversimplification. SVM was nearly flawless but proved to be too resource heavy. Neural network shows the best balance of performance and scalability and considered as the best option for real-time threat detection.
The class imbalance problem added with the balancing act of tending to the high number of false positives and false negatives created quite the challenge. While the models were doing quite fine with class imbalance, there is open room for future developments that involve resampling methods, resembling methods or performance boosting techniques. There is also the underlying need to continuously retrain and validate models in the face of changing patterns of threats to maintain accuracy.
The assessment confirms the ability of AI-based threat intelligence systems to automate detection of cyber-attacks and affirms their importance in improving an organization’s cyber security posture.

7 Conclusion and Future Work

Using neural networks, Support Vector Machines, and Logistic Regression models, it has been successfully demonstrated that machine learning approaches can distinguish between malicious network activity. By differentiating the models, the neural network gave the best results since it was able to abstract many complicated non-linear relationships within a dataset.

The entire flow of preprocessing such as scaling and feature selection helped ensure model stability. The results of the evaluation presented an equal detection of cases of attack and security, proving the practicality of this model in real-time cybersecurity operations. The findings have stressed the importance of keeping data always updated while also ensuring that models are retrained and validated to keep pace with evolving cyber threats. Embedding these AI models into security points basically provides a scale, automated solution in augmenting an organization's power to prepare and respond. In this respect, machine learning is an empowering technique in developing cyber defense mechanisms against the rise of digital threats.
Future research could explore how to put into practice the trained neural network model in streaming environments for proactive threat detection and streaming context for live performance benchmarking evaluation. Additionally, the model must be developed to autonomously and continuously retrain with the latest datasets and cyber threat intelligence to adapt and respond proactively to evolving cyber threat landscapes.

References:

Oosthoek, K. and Doerr, C., 2021, December. Inside the matrix: CTI frameworks as partial abstractions of complex threats. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 2136-2143). IEEE. https://research.tudelft.nl/files/104991156/Inside_the_Matrix_CTI_Frameworks_as_Partial_Abstractions_of_Complex_Threats.pdf

Oliveira, N., Praça, I., Maia, E. and Sousa, O. (2021). Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems. *Applied Sciences*, 11(4), p.1674. doi: https://doi.org/10.3390/app11041674.

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y. and Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 25(3), pp.1–1. doi: https://doi.org/10.1109/comst.2023.3273282.

Kayode-Ajala, O., 2023. Applications of Cyber Threat Intelligence (CTI) in financial institutions and challenges in its adoption. Applied Research in Artificial Intelligence and Cloud Computing, 6(8), pp.1-21.

Halbouni, A., Gunawan, T.S., Habaebi, M.H., Halbouni, M., Kartiwi, M. and Ahmad, R. (2022). Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access*, 10, pp.19572–19585. doi: https://doi.org/10.1109/access.2022.3151248.

Ke, H., Xu, J., Wang, Y., Chen, H. and Shen, Z. (2025). Adversarial Machine Learning in Cybersecurity: Attacks and Defenses. *International Journal of Management Science Research*, [online] 8(2), pp.26–33. doi: https://doi.org/10.53469/ijomsr.2025.08(02).04.

Zhou, Y., Tang, Y., Yi, M., Xi, C. and Lu, H. (2022). CTI View: APT Threat Intelligence Analysis System. *Security and Communication Networks*, 2022, pp.1–15. doi: https://doi.org/10.1155/2022/9875199.

Ranade, P., Piplai, A., Mittal, S., Joshi, A. and Finin, T., 2021, July. Generating fake cyber threat intelligence using transformer-based models. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-9). IEEE.

Jony, A.I. and Arnob, A.K.B. (2024). A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset. *Journal of Edge Computing*. [online] doi: https://doi.org/10.55056/jec.648.

Kaushik, Dr.Priyanka. (2023). Unleashing the Power of Multi-Agent Deep Learning: Cyber-Attack Detection in IoT. *International Journal for Global Academic & Scientific Research*, 2(2), pp.23–45. doi: https://doi.org/10.55938/ijgasr.v2i2.46.

Bharadiya, J. (2023). Machine Learning in Cybersecurity: Techniques and Challenges. *European Journal of Technology*, [online] 7(2), pp.1–14. doi: https://doi.org/10.47672/ejt.1486.

Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V. and Di Franco, F., 2023. The role of machine learning in cybersecurity. Digital Threats: Research and Practice, 4(1), pp.1-38.

Apoorv Joshi - Home. (2025). *Author DO Series*. [online] doi: https://doi.org/10.1145/contrib-99660733127.

Aslan, Ö., Aktuğ, S.S., Okay, M.O., Yilmaz, A.A. and Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics*, [online] 12(6), pp.1–42. doi: https://doi.org/10.3390/electronics12061333.

Pinto, S.J., Siano, P. and Parente, M., 2023. Review of cybersecurity analysis in smart distribution systems and future directions for using unsupervised learning methods for cyber detection. Energies, 16(4), p.1651.

Muhammad, A.R., Sukarno, P. and Wardana, A.A. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. *Procedia Computer Science*, [online] 217(1), pp.1406–1415. doi: https://doi.org/10.1016/j.procs.2022.12.339.

Li, Z.X., Li, Y.J., Liu, Y.W., Liu, C. and Zhou, N.X., 2023. K-CTIAA: automatic analysis of cyber threat intelligence based on a knowledge graph. Symmetry, 15(2), p.337.

Alotaibi, A. and Rassam, M.A., 2023. Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet*, *15*(2), p.62.

Yu, J., Shvetsov, A.V. and Saeed Hamood Alsamhi (2024). Leveraging Machine Learning for Cybersecurity Resilience in Industry 4.0: Challenges and Future Directions. IEEE Access, [online] pp.1–1. doi: https://doi.org/10.1109/access.2024.3482987.

Haider, A., Adnan Khan, M., Rehman, A., Rahman, M. and Seok Kim, H. (2021). A Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System. *Computers, Materials & Continua*, 66(2), pp.1785–1798. doi: https://doi.org/10.32604/cmc.2020.013910.

Nick Jain, 2023. What is Quantitative Research Design? Definition, Types, Methods and Best Practices. Available at: https://ideascale.com/blog/quantitative-research-design/[Accessed on 20th June, 2025]

Sridevi Ponmalar P, Kamboj, A., Vikram Shete and R. Harikrishnan (2022). Machine Learning Based Network Traffic Predictive Analysis. 9(2), pp.96–108. doi: https://doi.org/10.18488/76.v9i2.3065.

Jaz Allibhai (2018). *Building A Deep Learning Model using Keras - TDS Archive - Medium*. [online] Medium. Available at: https://medium.com/data-science/building-a-deep-learning-model-using-keras-1548ca149d37

Lopez, E., Jaione Etxebarria-Elezgarai, Jose Manuel Amigo and Seifert, A. (2023). The importance of choosing a proper validation strategy in predictive models. A tutorial with real examples. *Analytica chimica acta*, 1275, pp.341532–341532. doi: https://doi.org/10.1016/j.aca.2023.341532.

Pickering, B., Roth, S. and Webber, C., 2021. Ethical Approaches to Studying Cybercrime: Considerations, Practice and Experience in the United Kingdom. *Researching Cybercrimes: Methodologies, Ethics, and Critical Approaches*, pp.347-369.

Wang, X., Qiao, Y., Xiong, J., Zhao, Z., Zhang, N., Feng, M. and Jiang, C. (2024). Advanced Network Intrusion Detection with TabTransformer. *Journal of Theory and Practice of Engineering Science*, [online] 4(03), pp.191–198. doi: https://doi.org/10.53469/jtpes.2024.04(03).18.

Al-Khassawneh, Y.A. (2023). A Review of Artificial Intelligence in Security and Privacy: Research Advances, Applications, Opportunities, and Challenges. *Indonesian Journal of Science and Technology*, [online] 8(1), pp.79–96. doi: https://doi.org/10.17509/ijost.v8i1.52709.

Sajid, M., Kaleem Razzaq Malik, Almogren, A., Tauqeer Safdar Malik, Ali Haider Khan, Tanveer, J. and Ateeq Ur Rehman (2024). Enhancing intrusion detection: a hybrid machine and deep learning approach. *Journal of Cloud Computing Advances Systems and Applications*, 13(1). doi: https://doi.org/10.1186/s13677-024-00685-x.

Gao, P., Shao, F., Liu, X., Xiao, X., Qin, Z., Xu, F., Mittal, P., Kulkarni, S.R. and Song, D., 2021, April. Enabling efficient cyber threat hunting with cyber threat intelligence. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (pp. 193-204). IEEE.

Bilen, A. and Özer, A.B. (2021). Cyber-attack method and perpetrator prediction using machine learning algorithms. *PeerJ Computer Science*, 7, p.e475. doi: https://doi.org/10.7717/peerj-cs.475.

Sharma, A., Tyagi, A. and Bhardwaj, M. (2022). Analysis of techniques and attacking pattern in cyber security approach. *International journal of health sciences*, pp.13779–13798. doi: https://doi.org/10.53730/ijhs.v6ns2.8625.

Asiri, M., Saxena, N., Gjomemo, R. and Burnap, P., 2023. Understanding indicators of compromise against cyber-attacks in industrial control systems: a security perspective. *ACM transactions on cyber-physical systems*, *7*(2), pp.1-33.

Zhang, J., Pan, L., Han, Q.-L., Chen, C., Wen, S. and Xiang, Y. (2021). Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey. *IEEE/CAA Journal of Automatica Sinica*, 9(3), pp.1–15. doi: https://doi.org/10.1109/jas.2021.1004261.