

AUTOMATED INTRUSION DETECTION IN NETWORK TRAFFIC USING AI

MAJOR PROJECT

Submitted in the Partial Fulfilment for the Requirement for the Award of the Degree of
Master of Technology
in
Software Engineering

Submitted by:

Chandan Kumar (2k24/SWE/22)

Under the supervision of

Dr. Rahul

Assistant Professor



Department of Software Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May 2026



Department of Software Engineering

Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide, **Dr. Rahul**, for her valuable guidance and wisdom in the development of this project. I would also like to humbly extend my gratitude to **Prof. Ruchika Malhotra** (HOD, Department of Software Engineering) and all the faculty members of the Department of Software Engineering at Delhi Technological University for their constant support, valuable advice, and encouragement throughout the project.

I would also like to thank the University for providing us with the laboratories, infrastructure, testing facilities, and a conducive environment that enabled us to work continuously without interruption. Additionally, I am grateful to our lab assistants, seniors, and peer group for their support, assistance and knowledge shared on a variety of subjects.

Place: New Delhi

Chandan Kumar (2k24/SWE/22)

Date:



Department of Software Engineering

Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE's DECLARATION

I, **Chandan Kumar**, Roll No. **24/SWE/22** student of M.Tech (Software Engineering), hereby certify that the work which is being presented in the thesis entitled "**Automated Intrusion Detection in Network Traffic Using AI**" in partial fulfilment of the requirements for the award of the Degree of Master of Technology, submitted to the Department of Software Engineering, Delhi Technological University, is an authentic record of my own work carried out during the period from June 2025 to May 2026 under the supervision of **Dr. Rahul**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Chandan Kumar

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.


Signature of Supervisor(s)

Signature of External Examiner



Department of Software Engineering

Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "AUTOMATED INTRUSION DETECTION IN NETWORK TRAFFIC USING AI" which is submitted by Chandan Kumar (2k24/SWE/22), Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: New Delhi

Date: 2/06/26

A handwritten signature in blue ink, appearing to be "Dr. Rahul", is written over a circular stamp that matches the Delhi Technological University logo.

Dr. Rahul

Assistant Professor

ABSTRACT

This report examines how Artificial Intelligence (AI) and Machine Learning (ML) can be applied to automate the analysis of network traffic in order to improve the detection of cyber threats. The study is based on the increased need for scalable security solutions, due to growing data volumes and an increasingly complex threat landscape, where traditional manual analysis methods exhibit significant limitations.

The aim is to analyze how AI and ML techniques can be used to identify patterns and anomalies in raw network traffic, and to evaluate the extent to which these techniques can replace or complement manual threat hunting. The study also aims to highlight the opportunities and limitations associated with automated threat detection in operational cybersecurity environments.

LIST OF ABBREVIATIONS

AI – Artificial Intelligence

ML – Machine Learning

IoT – Internet of Things

IT – Information Technology

SIEM – Security Information and Event Management

IDS – Intrusion Detection System

SIDS – Signature-based IDS

IP – Internet Protocol

XDR – Extended Detection and Response

XGBoost – Extreme Gradient Boosting

MLOps – Machine Learning Operations

GUI – Graphical User Interface

AutoML – Automated Machine Learning

NLP – Natural Language Processing

BERT – Bidirectional Encoder Representations from Transformers

Auto-ARIMA – Automated AutoRegressive Integrated Moving Average

CPU – Central Processing Unit

GPU – Graphics Processing Unit

REST API – Representational State Transfer Application Programming Interface

Python SDK – Python Software Development Kit

CLI – Command Line Interface

YAML – YAML Ain't Markup Language

CI/CD – Continuous Integration and Continuous Delivery

SMOTE – Synthetic Minority Over-sampling Technique

BENIGN – Harmless / Non-malicious Traffic

DDoS – Distributed Denial of Service

DOS – Disk Operating System

SQL – Structured Query Language

XSS – Cross-Site Scripting

PCAP – Packet Capture

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

FTP – File Transfer Protocol

SSH – Secure Shell

CSV – Comma-Separated Values

CNN – Convolutional Neural Network

TCP – Transmission Control Protocol

KNN – K-Nearest Neighbours

TP – True Positive

TN – True Negative

FP – False Positive

FN – False Negative

ROC – Receiver Operating Characteristic

AUC – Area Under Curve

OVR – One-vs-Rest

SVC – Support Vector Classification

SGD – Stochastic Gradient Descent

ANN – Artificial Neural Network

SVM – Support Vector Machine

LIST OF FIGURES

Figure 1: Overview of public IDS datasets compared with respect to protocol diversity, traffic types, and metadata (Sharafaldin, Lashkari & Ghorbani, 2018)

Figure 2: ROC curves for different models and AUC variants. A model with a high AUC (closer to 1.0) has good separation ability between classes, while a model close to 0.5 performs like chance

Figure 3: Confusion matrix for a well-trained model. Most observations lie along the diagonal, indicating good accuracy

Figure 4: Confusion matrix for a weak model. Many misclassifications occur, which is visible in the off-diagonal distribution

Figure 5: Effect of oversampling strategy

Figure 6: ROC curves and Precision-Recall curves

Figure 7: Multi-class confusion matrix

Figure 8: Model Comparison

Figure 9: Features

Figure 10: Confusion Matrices

Figure 11: Class Distribution

CONTENTS

ACKNOWLEDGEMENT	2
CANDIDATE’S DECLARATION	3
CERTIFICATE.....	4
ABSTRACT	5
LIST OF ABBREVIATIONS	6
LIST OF FIGURES	9
CHAPTER-1	12
INTRODUCTION	12
1.1 BACKGROUND	13
1.1.1 Pattern Recognition And Anomaly Detection.....	14
1.2 PROBLEM STATEMENT AND RESEARCH QUESTIONS.....	15
1.3 PURPOSE	16
CHAPTER-2	17
LITERATURE REVIEW AND THEORETICAL BACKGROUND	17
2.1 RAW DATA IN NETWORK SECURITY	17
2.1.1 What Is Raw Data And Why Is It Important?.....	17
2.1.2 Log Files And Network Traffic As Security Sources	17
2.1.3 The Need For Analysis Of Unfiltered Security Systems	18
2.2 THREAT HUNTING AND MANUAL ANALYSIS LIMITATIONS	18
2.3 PROACTIVE CYBERSECURITY THROUGH AI	20
2.4 MACHINE LEARNING - FUNDAMENTALS AND METHODS	20
2.4.1 Supervised Learning	21
2.4.2 Unsupervised Learning.....	21
2.4.3 Semi-Supervised Learning	21
2.5 ANOMALY DETECTION AND PATTERN RECOGNITION	21
2.6 AZURE MACHINE LEARNING AND AUTOML	22
2.6.1 Technical Overview: Azure Machine Learning.....	22
2.6.2 AutoML: Automated Model Development In Azure	23
2.7 CICIDS2017 – A STANDARDIZED DATASET	24
2.8 PERFORMANCE METRICS FOR MODEL EVALUATION	26
2.8.1 Classification Outcomes: TP, TN, FP, And FN	27
2.8.2 Accuracy	27

2.8.3 Precision	28
2.8.4 Recall	28
2.8.5 F1 score	29
2.8.6 AUC – Area Under the ROC Curve	29
2.8.7 Confusion Matrix	30
2.8.8 Macro, Micro And Weighted Calculations	32
2.8.9 Choosing A Primary Metric In Azure AutoMLml	32
2.9 CLASSIFICATION	33
2.9.1 Multiclass Classification	33
2.9.2 Classification Algorithms In Auto ML	34
2.9.2.1 Caruana Ensemble Selection Algorithm	35
2.10 DATA SCALING AND PREPROCESSING IN MACHINE LEARNING	36
2.11 PREVIOUS RESEARCH ON AI IN NETWORK SECURITY	36
2.11.1 Use Of CICIDS2017 In Previous Studies	36
2.11.2 Models That Have Shown High Accuracy	37
2.12 PROBLEMS RELATED TO USING ARTIFICIAL INTELLIGENCE IN THREAT DETECTION	38
2.12.1 Bias In Training Data	38
2.12.2 Lack Of Transparency And Interpretability	39
2.12.3 Generalizability To Real-World Environments	39
2.13 SUSTAINABLE DIGITALIZATION AND CLOUD INFRASTRUCTURE	40
CHAPTER-3	41
METHODOLOGY	41
3.1 DATASET EXPLANATION	43
3.2 PREPROCESSING	44
3.3 ALGORITHM WORKFLOW	46
CHAPTER-4	52
RESULTS	52
CHAPTER-5	61
CONCLUSION	61
CHAPTER-6	63
DISCUSSION	63
CHAPTER-7	66
FUTURE SCOPE	66
REFERENCES	68

CHAPTER-1

INTRODUCTION

The global amount of data is growing exponentially as a result of digitalization, increased use of the Internet of Things (IoT), cloud services and advances in artificial intelligence (Bartley 2024). Forecasts show that the data volume could exceed 181 zettabytes by 2025, which places increased demands on storage, processing and data security (Bartley 2024). In parallel, the vulnerability of IT infrastructures is increasing as organizations' dependence on network-based communication intensifies.

Networks generate large amounts of raw data in the form of logs from firewalls, servers, applications and other network components. This type of network traffic is a concrete example of so-called big data, i.e. large, complex and rapidly generated data sets that are difficult to handle with traditional methods. This further highlights the requirement for scalable and automated methods for processing data, particularly in cybersecurity. Previously, analysis of such data was done through security tools like SIEM (Security Information and Event Management), although a very small percentage of this data has ever been analyzed in real-time (Joshi, Ghuge, Anjarlekar and Chavan, 2025). In essence, this means that a lot of data that could help in predicting future threats to cybersecurity remains unused.

This could be because of the fact that the volume and complexity of information have become overwhelming for human beings to process and react to in a timely manner. According to Mahboubi et al. (2024), the major challenge associated with threat hunting includes the challenge of lack of enough time and resources to detect threats in huge amounts of data generated by security systems. The larger the amount of data, the higher the likelihood that the data will not be detected manually.

According to Joshi et al. (2025), manual log analysis is a lengthy and expensive activity that inevitably leads to mistakes in the procedure. With the fast-changing nature of cyberattacks and their automation, traditional techniques of threat detection appear to be unable to provide a proactive approach to cybersecurity defense. Therefore, it seems necessary to apply

advanced instruments able to analyze large amounts of information quickly and effectively, according to Joshi et al. (2025).

It is the rising complexity of attacks that drives experts to create new methods of pattern recognition and anomaly detection. As stated by Ortiz-Garcés, Govea, Sánchez-Viteri, and Villegas-Ch. (2024), pattern recognition and anomaly detection become crucial because these deviations indicate the existence of attacks. It is important to note that these deviations are key in recognizing attacks since signature-based detection cannot identify them all.

Machine learning and artificial intelligence provide new ways of automating the process of recognizing anomalies and attack patterns in network traffic (Han, Guo, Wang, and Zhang, 2023). Through the analysis of raw data in logs and discovering of hidden relations and anomalies, security work becomes more effective and efficient. Automation of the process is likely to decrease the workload of security specialists while making threat detection and management possible.

Against this background, the study aims to investigate how machine learning and artificial intelligence techniques can be applied to analyze raw data from network logs in order to identify patterns and anomalies related to cyberattacks. The ability to address the imbalances in the training data is one of the key factors to be considered when assessing the performance of the proposed models, since it could influence their capability to detect attacks. In the current research, the attention will be paid to analyzing structured network traffic data from the CICIDS2017 database.

1.1 BACKGROUND

The growth of data volume in IT infrastructure has given rise to both new opportunities for business growth and security challenges (Zuech, Khoshgoftaar & Wald, 2015). Corporations have an ever-increasing amount of raw data coming from corporate networks in the form of firewall, server, application, and other network component logs. Often, such logs may contain information about the behavior and security state of the system (Joshi et al., 2025). Conventionally, processing of this kind of data occurs in Security Information and Event Management (SIEM) systems, which specialize in analyzing and correlating events in real time. In general, SIEM technologies are rule-based, meaning that all the unusual or non-standard traffic goes through filtering (Joshi et al., 2025). It means that any security-relevant

data may be ignored by SIEM solutions. While SIEM technology provides certain automation and helps analyze data, its efficiency still relies on the use of predefined rules and human interaction. With the increase of data volume, it becomes unreasonable to employ humans as the primary resource for data processing, as argued by Mahboubi et al. (2024).

Intrusion Detection Systems (IDS) have been developed as an important complement to SIEM systems to enable the detection of both known and unknown attack patterns in network traffic. However, traditional signature-based IDS (SIDS) are limited to detecting attacks that are already documented in signature databases, and thus lack the capacity to identify new, as yet unknown threats such as zero-day attacks (Khraisat, Gondal, Vamplew, & Kamruzzaman, 2019).

1.1.1 Pattern Recognition And Anomaly Detection

According to Ortiz-Garcés et al. (2024), one of the major tools used in cybersecurity is pattern recognition. Pattern recognition is helpful in identifying anomalies that do not follow the patterns of usual behaviors in network traffic. After defining the normal patterns of activities within a network, any anomaly could be detected, which may possibly pose a threat to the network.

Traditional approaches to attacks based on signatures suffer from the limitation that they can detect only those attacks that are known before since they depend on threat patterns observed previously. The process of pattern recognition and anomaly detection helps in identifying any new type of attack through dynamic analysis of traffic flows and activities. According to Ortiz-Garcés et al. (2024), dynamic analysis is required because cyberattacks constantly evolve using innovative techniques in order to bypass conventional techniques used for detecting these cyberattacks.

It has been observed that the rise in security threats has resulted in the deployment of machine learning and artificial intelligence in order to improve traffic analysis and intrusion detection (Han et al., 2023). Intrusion detection systems based on networks employ machine learning algorithms in their functioning by training them with past data in order to differentiate between normal and abnormal network traffic. One other method of detecting intrusions is through anomaly detection.

Standard datasets are used for the purpose of developing and testing intrusion detectors. One such dataset is CICIDS2017 (Sharafaldin, Lashkari & Ghorbani, 2018), which consists of simulated network traffic data including both normal traffic and 14 different cyberattacks. The traffic has been converted to bidirectional flows using the CICFlowMeter tool, which generates 79 statistical functions per flow. This provides a comprehensive basis for training and validating intrusion detection models.

CICIDS2017 thus constitutes a stable and representative basis for developing and evaluating machine learning models in network security, which is of great importance in dealing with today's increasing cyber threats.

1.2 PROBLEM STATEMENT AND RESEARCH QUESTIONS

Despite the exponential increase in data volumes in corporate networks, driven by technologies such as AI, IoT and cloud storage, a large part of the collected network traffic remains insufficiently analysed. In many organizations, network traffic is stored in platforms such as Azure Data Lake and is still analyzed manually by threat hunting and cybersecurity specialists.

The manual analysis of large amounts of network data is both resource-intensive, time-intensive and difficult to scale. It is imperative that firms have the ability to invest in skilled security experts to examine traffic flows and logs all the time in order to detect any intrusion attempts or other anomalies. As a result, it becomes costly for firms to operate and difficult to scale their operations as the amount of data increases. Furthermore, the use of manual techniques increases the likelihood of attacks being detected too late or even not at all due to human error. In some situations, firms are not able to allocate the necessary resources to be able to monitor traffic flows continually. This is why the use of AI in threat hunting becomes crucial because, according to Mahboubi et al. (2024), automated systems can handle much more data than humans.

Against this background, a number of key research questions arise:

- To what extent can automated machine learning models contribute to the detection of threats in raw network traffic?
- How is the accuracy of an AI-based model affected by class imbalance in the CICIDS2017 dataset when detecting different types of attacks?

1.3 PURPOSE

The aim of this research paper is to analyze how machine learning and artificial intelligence can be used for the automation of the analysis process of raw data network traffic for the purpose of detecting anomalies and patterns of cyber attacks. It will also be discussed how machine learning and artificial intelligence can help in the analysis process, which is traditionally done manually, for detecting patterns in network security.

CHAPTER-2

LITERATURE REVIEW AND THEORETICAL BACKGROUND

2.1 RAW DATA IN NETWORK SECURITY

2.1.1 What Is Raw Data And Why Is It Important?

Raw data is defined as information collected directly from instruments, sensors, or other sources before any extensive processing has taken place. According to Borgman (2015), this data often forms the basis of scientific analysis and can include, for example, network logs, traffic patterns, and other digital signals. Although level 0 data in technical systems such as NASA's EOS DIS is considered raw, it has often already undergone some cleaning, which highlights that the concept of "raw data" is relative and context-dependent (Borgman, 2015, pp. 47–50).

Raw data can be categorized as either quantitative, such as numerical measures such as traffic frequency or packet size, or qualitative, where contextual interpretation is required. In network security, raw data consists primarily of quantitative variables, making it particularly suitable for machine learning and pattern recognition. Borgman (2015) points out that researchers often prefer access to as little pre-processed data as possible, as processing can mean that subtle anomalies or patterns disappear.

Therefore, especially in security-critical applications such as intrusion detection, it is often necessary to analyze information in its most original form. Preserving different levels of processing of data is seen as an important aspect of research infrastructure, as it allows different types of analysis to be carried out while maintaining transparency and reproducibility (Borgman, 2015, pp. 48–52).

2.1.2 Log Files And Network Traffic As Security Sources

In current IT systems, a huge amount of network raw data is produced daily, namely system logs, traffic flows, authentication requests, firewall actions, and applications logs. All

together, these sources of information give us a comprehensive view of what happens within the network. Network raw traffic can contain not only regular but also suspicious actions and traffic flows that can be useful in cybersecurity. According to Avila, Khoury, Khoury, and Petrillo (2021), analyzing security logs is important for detecting leaks and attacks because they may contain anomalies and leaks of information.

2.1.3 The Need For Analysis Of Unfiltered Security Systems

Although modern IT systems produce a huge amount of network data, their potential is not fully used. According to Avila et al. (2021), security systems such as SIEM are mainly interested in finding out events that correspond to some previously known rules or signatures.

This creates a blind spot for unknown, unusual, or slowly developing attacks, which therefore risk remaining undetected.

To complement these systems, manual analysis in the form of so-called threat hunting is sometimes used. However, this method is resource-intensive and requires both time, expertise and continuous staffing, making it difficult to apply on a larger scale. According to Avila et al., few organizations can effectively analyze all of the existing traffic systematically because there is not enough capacity. Therefore, new approaches should be sought to solve this problem.

The authors point out that it is necessary to develop automated ways to conduct log analysis, especially those that are related to machine learning and artificial intelligence. These tools can be used not only to detect anomalies in the network traffic but also to constantly improve their performance and differentiate between normal and deviant behavior by applying dynamic algorithms instead of static signatures.

2.2 THREAT HUNTING AND MANUAL ANALYSIS LIMITATIONS

Threat hunting is the process of proactively searching for signs of any threat using analysis of the data from network and computer systems, where the objective is to find patterns or anomalies that could be indicative of an attack.

In contrast to typical security systems that respond to alerts generated by them, threat hunting relies on hypothesis-driven approach, whereby security experts manually analyze indicators that cannot be detected by signature-based systems. According to Belal and Sundaram (2022), the aim is to detect such cyberattacks that can evade automatic protection through the simulation of normal network behavior.

Even though threat hunting allows flexible application and discovery of new threats, it requires access to expert knowledge and resources. Threat hunting, in particular, is mostly performed in a targeted manner and not continuously. It is clear that they currently do not allocate resources for manual analysis of the extensive data generated by XDR platforms and other log sources. At the same time, they see clear potential for improvement in using machine learning to strengthen the ability to detect threats.

As Belal and Sundaram (2022) note, traditional detection systems were not meant to be used for processing large amounts of data. The problem is that current networks generate huge amounts of information that cannot be effectively processed manually, thus making it very likely that some possible dangers will remain unnoticed.

Furthermore, this challenge is confirmed by security experts, who state that only selected parts of the raw data are manually reviewed, and that some data is purged or archived after 30–90 days depending on policy. The actual threat identification is mainly done through automated tools such as Microsoft Sentinel and Defender, which means that threats that do not match known patterns risk remaining undetected without additional technical support. In addition to the resource requirements, threat hunting also has significant scalability limitations. It is unreasonable to expect individual analysts to be able to identify patterns in real time across millions of data points without technical support. Mahboubi (2024) highlights that human cognition cannot easily detect small changes in high-dimensional and fast-changing data, thus making the threat remain unnoticed in manual approaches. Furthermore, there is an increased possibility of errors where the human being interprets the data independently. To solve these problems, Belal and Sundaram (2022) stress the need for replacement or augmentation of manual approaches through intelligent approaches based on machine and deep learning. Intelligent approaches make it possible to detect anomalies and raise alarms accurately.

2.3 PROACTIVE CYBERSECURITY THROUGH AI

As a result of the increased sophistication of cyber attacks, the cyber security strategy has transformed from a reactive method to a predictive method. The emergence of new technologies has helped this transformation take place. For example, due to innovations in artificial intelligence and machine learning techniques, it is now possible to process large volumes of data to detect cyber attacks (Belal & Sundaram, 2022).

The idea of automation plays an important role in this shift. As noted by Shaukat et al. (2020), ML algorithms can be used to analyze any anomalies in the network traffic that cannot be caught by using rules-based techniques. The advantage of such algorithms is that they have the ability to constantly learn about new threats that might arise.

Tang, Alazab and Luo (2017) further show that proactive analysis supported by AI not only enables the detection of ongoing attacks, but can also help to predict future threats by identifying patterns in historical and contextual data. Predictive systems thus provide organizations with the conditions to act preventively, which reduces response times and increases the accuracy of detecting advanced attacks. The proactive approach thereby reshapes the fundamental logic of cybersecurity. With the use of AI-based techniques, organizations can become proactive, rather than reactive, which is especially relevant given the fast-changing nature of threats and resource constraints.

From one perspective, this example demonstrates what can be achieved through the application of artificial intelligence and machine learning from a technical standpoint; from another perspective, this is a theoretical foundation of cybersecurity.

2.4 MACHINE LEARNING - FUNDAMENTALS AND METHODS

Machine learning refers to an academic discipline that involves more than one discipline, and the objective of this discipline is the creation of computer programs that can identify patterns and draw conclusions from such patterns irrespective of whether there are any instructions from the programmer beforehand. As stated by Li et al. (2024), machine learning requires the use of statistical and probabilistic techniques to identify patterns and predict them. However, it was not until the 1980s that Herbert A. Simon, the Nobel Laureate and an

excellent researcher in artificial intelligence and decision-making theory, defined learning as the process of improving one's performance on the basis of experience.

2.4.1 Supervised Learning

Supervised learning can be seen as the earliest kind of machine learning, wherein the algorithm is trained by using an input-output (or labeled) dataset. Some of the most prevalent forms of supervised learning include regression and classification (both for continuous and discrete values respectively), making supervised learning extremely useful for intrusion detection, diagnostics, and finance (Li, Lin, and Zeng, 2024).

2.4.2 Unsupervised Learning

Unsupervised learning comes into play in cases where the data does not have any preexisting labels. In this case, the algorithm tries to find some form of pattern in the data by using clustering or dimensionality reduction techniques, among others. The main application areas of unsupervised learning are those where there is no clear understanding of what is normal behavior and what is abnormal (Li, Lin, and Zeng, 2024).

2.4.3 Semi-Supervised Learning

Semi-supervised learning involves both labeled and unlabeled data points during training, and it proves useful when labeling becomes expensive. As stated by Zhou et al. (2023), semi-supervised learning methods have proven to be useful in improving the performance of models, particularly in anomaly detection where there are very few attacks.

2.5 ANOMALY DETECTION AND PATTERN RECOGNITION

Pattern recognition is an essential part of anomaly detection, which is crucial in achieving cybersecurity. This is because most security threats, including data intrusion and exfiltration, exhibit statistical patterns different from those of normal network traffic. Pattern recognition helps in detecting such anomalies (Zamanzadeh Darban et al., 2024).

Moreover, the authors claim that, unlike signature-based approaches, which depend on the existing knowledge about the patterns of attacks, anomaly detection allows for the detection of previously unknown attacks. It is because by building models based on the data from

normal traffic, it is possible to spot potential abnormalities before they become actual problems.

Also, the authors state that the decision regarding the type of anomaly detection method to use largely depends on the availability of labeled data. In cases when there is no labeled data available, which is common in the field of network security, unsupervised techniques like clustering or autoencoders are used, while supervised methods are employed in cases when there is labeled data available for both normal and abnormal traffic. Besides, traditional statistical methods are also considered, which demonstrates the versatility of the approach.

2.6 AZURE MACHINE LEARNING AND AUTOML

Various prior works have emphasized the necessity of deploying machine learning in an automated manner due to the need for processing large volumes of complex and unstructured datasets (Tang, Alazab & Luo, 2017; Belal & Sundaram, 2022). In the case of cybersecurity, the importance of cloud computing platforms, such as Azure Machine Learning (Azure ML), has grown because they allow developers to create ML models while providing various types of automation, including AutoML. Thus, combining such tools allows applying machine learning in practice in the face of specific requirements related to data storage capacity and analysis speed.

It is worth noting that previous works have proven that Azure ML can be effectively used for classification purposes, time-series analysis, and real-time monitoring for safety-critical applications (Fakhraee, Balakreshnan & Masanz, 2023). Thanks to MLOps, which is readily available in Azure ML, it becomes possible to control the entire lifecycle of the model through tracking the data pre-processing, training, deployment, monitoring, and re-training phases.

2.6.1 Technical Overview: Azure Machine Learning

Azure Machine Learning is a cloud-based framework offered by Microsoft which allows developers to create, train, deploy and manage their machine learning models. The tool is able to use both the workflow based on programming code as well as graphical interface (GUI), as well as provide support for various existing open-source platforms such as PyTorch, TensorFlow, and scikit-learn (Microsoft, 2023).

One of the major strengths of Azure ML is the tight connection between various services provided by Azure, thus allowing for the creation of consistent and scalable infrastructure for machine learning. Some examples of those include: Azure ML studio for management of experiments, models and pipeline; Azure Key Vault for secure authentication keys management; Application Insights for logging and performance tracking; Azure Container Registry for Docker images management; and Azure Storage for structured management of data required for model training and validation (Microsoft, 2023). Therefore, thanks to all these capabilities, Azure ML offers a comprehensive environment for machine learning model lifecycle management.

2.6.2 AutoML: Automated Model Development In Azure

The concept of Automated Machine Learning (AutoML) in Azure Machine Learning entails an automation of various stages involved in the development of ML models, including selecting the right algorithm, carrying out the feature engineering process, hyperparameter tuning, and data splitting into training, testing, and validating datasets. This is achieved by creating multiple pipelines simultaneously in order to evaluate several configurations, which are then compared based on their ability to produce the best performance according to the predefined criterion. The selected model that shows superior performance becomes the one to be applied as a standard for use (Microsoft, 2025).

Many scholars prove the effectiveness of using AutoML for building effective models requiring minimal human involvement. This method has been found quite effective in different sectors, including cybersecurity, where it is utilized to develop ML models forecasting incidents, network classification, and intrusions (Google Developers, 2024). The AutoML technique may also be applied in classification, regression, forecasting, computer vision, and NLP, among others. Some special types of models include BERT in NLP and Auto-ARIMA in time series analysis.

In addition to the AutoML service, Azure ML supports other development environments tailored to specific needs of different user groups. Azure ML Designer is a no-code environment that allows building ML models without programming skills. The platform provides a graphical environment that allows building a workflow from data import through the model development stage to its training and deployment as an endpoint, with capabilities for advanced data preprocessing and training on CPU/GPU cluster environments. The Designer makes it possible for users to have a straightforward way of creating models by not

writing any code and then publishing them to REST API endpoints (Fakhraee, Balakreshnan & Masanz, 2023).

In more complex scenarios, a code-first method can also be applied. The latter means having control over the whole process of model development in terms of code through Python SDK or Azure CLI. With the help of a code-first method, users have full access to their environment, settings, and resources. This method is suitable to apply in cases where it is necessary to work with the applications where CI/CD and flexibility matter.

2.7 CICIDS2017 – A STANDARDIZED DATASET

CICIDS2017 Data Set is an experimental data set, which is used when developing models using machine learning to detect intrusions. The said data set has been described as modern and extremely valuable in this field of study by Zhao et al., 2023. The data set has 2,827,876 flows that are categorized into those classified as being normal (BENIGN) at 80.3% and fourteen different types of attacks that account for the other 19.7%.

Unlike traditional log files, CICIDS2017 is based on network sessions collected in a controlled test environment. Using CICFlowMeter, NetFlow-like attributes were extracted from raw PCAP traffic, resulting in 79 features per data point. Each data record represents a bidirectional flow between two parties and is annotated as benign or belonging to a specific attack category. The above arrangement allows this data set to be ideal for supervised learning algorithms since it already comes with labeled data and realistic network traffic. The B-Profile mechanism was utilized to create realistic network patterns through simulating behavior from twenty-five user profiles (Sharafaldin, Lashkari & Ghorbani, 2018). They interact using protocols like HTTP, HTTPS, FTP, SSH, and email. The collection of data took place over five days ranging from Monday, July 3 to Friday, July 7, 2017, where only benign traffic was created on the first day, while an attack took place on all other days (Zhao, Hu, & Liu, 2023).

The recorded traffic was processed with CICFlowMeter V4.0 to generate CSV files with statistical properties such as packet density, baud rate, inter-arrival times, and TCP flags (Lashkari et al., 2018). Thanks to this structured feature extraction, the dataset is directly accessible for both supervised and unsupervised learning methods.

The dataset meets several criteria considered central to robust intrusion detection: complete network topology, high diversity of traffic and attacks, full protocol coverage, extensive metadata, and clear documentation of flows and labels. In a comparison with eleven other public IDS datasets, CICIDS2017 proved to be one of the few that offers complete protocol and attack diversity, something that is clearly evident in the original research report's compilation (Sharafaldin, Lashkari & Ghorbani, 2018) (see Figure 1).

In addition to structure and labels, CICIDS2017 also contains quantitative results that show how different classification algorithms perform on the dataset. In this original evaluation, for example, Random Forest and KNN showed both high precision ($Pr \approx 0.98$ and 0.96 , respectively) and a good balance between recall and execution time, while algorithms such as Naïve Bayes were faster but somewhat less accurate (Sharafaldin, Lashkari & Ghorbani, 2018). The most important features for each attack type were also analyzed, which facilitates customized model training.

CICIDS2017 thus offers a well-documented and structured basis for training and testing models in network security. The combination of realistic benign traffic, varied attack data and low incidence of incorrect labeling creates good conditions for developing models with high precision and generalization ability (Zhao, Hu and Liu, 2023).

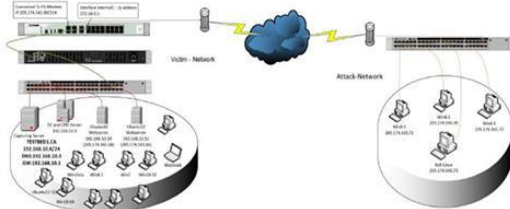
At the same time, there are certain methodological limitations to consider. Since the data is collected in a simulated environment, the traffic patterns may differ from real production networks. In addition, the attacks are relatively short-lived and predefined, which may affect the models' ability to generalize to slower, distributed, or complex attacks. For this reason, results based on CICIDS2017 should be interpreted with caution when applied to operational environments.

List of CICIDS2017 Attacks

[illegible]

Days	Labels
Monday	Benign
Tuesday	BForce.SFTP and SSH
Wednes.	DoS and Heartbleed Attacks slowloris, Slowhttptest, Hulk and GoldenEye
Thurs.	Web and Infiltration Attacks Web BForce, XSS and Sql Inject. Infiltration Droptbox Download and Cool click
Friday	DDoS LOIT, Botnet ARES, PortScans (sS,tF,sX,sN,sP,sV,sU, sO,sA,sW,sR,sL and B)

Testbed Architecture



Top Features Per Attack Category

Label	Feature	Weight	
Benign	B-Packet Len Min	0.0479	SSH-Patator
	Subflow B-Fackets	0.0007	Subflow B-Fackets
	F-Packet Len F-Packets	0.0004	Total Len F-Packets
	F-Packet Len Mean	0.0002	ACK Flag Count
	B-Packet Len Std	0.1585	Win In B-Fackets
DoS GoldenEye	Flow IAT Min	0.0317	FTP-Patator
	Fwd IAT Min	0.0257	SYN Flag Count
	Fwd IAT Mean	0.0214	F-Packets
Heartbleed	B-Packet Len Std	0.3028	Web Attack
	Subflow B-Fackets	0.1367	Win In B-Fackets
	Flow Duration	0.0991	Total Len F-Packets
DoS Hulk	Total Len F-Packets	0.0903	Infiltration
	B-Packet Len Std	0.0728	Active Mean
	B-Packet Len Std	0.1777	Subflow B-Fackets
DoS Slowhitp	Flow Duration	0.0437	Total Len F-Packets
	Flow IAT Std	0.0232	F-Packet Len Mean
	Flow Duration	0.0443	B-Packets
DoS Slowhitp	Active Min	0.0219	Win In B-Fackets
	Active Mean	0.0228	B-Packets
	Flow IAT Std	0.0200	PSH Flag Count
DoS slowloris	Flow Duration	0.0431	B-Packet Len Count
	FIAT Min	0.0378	Avg Packet Size
	FIAT Mean	0.0000	Flow Duration
	FIAT Mean	0.0565	Flow IAT Std

The Performance Examination Results

Algorithm	Pr	Rc	F1	Execution (Sec.)
KNN	0.96	0.96	0.96	1908.23
RF	0.98	0.97	0.97	74.39
ID3	0.98	0.98	0.98	235.02
Adaboost	0.77	0.84	0.77	1126.24
MLP	0.77	0.83	0.76	575.73
Naive-Bayes	0.88	0.04	0.04	14.77
QDA	0.97	0.88	0.92	18.79

Conclusion and Future Works:

• In this paper, we have monitored the state-of-the-art in the IDS dataset generation and evaluation by analyzing the eleven publicly available datasets. Then we generate a new IDS dataset includes seven common updated family of attacks that meet real worlds criteria and is publicly available (<http://www.unb.ca/cic/datasets/IDS2017.html>).

Figure 1: Overview of public IDS datasets compared w.r.t protocol diversity, traffic types, and metadata

2.8 PERFORMANCE METRICS FOR MODEL EVALUATION

When developing and evaluating classification models in cybersecurity, the choice of performance metrics is crucial. Especially in applications such as intrusion detection, where the data is often highly unbalanced, i.e. normal traffic dominates over malicious traffic, traditional metrics such as accuracy can give a misleading picture of the model's real performance. Uddin in his article titled "Addressing Accuracy Paradox Using Enhanced Weighted Performance Metric in Machine Learning" talks about the problem of using only the accuracy measure when evaluating machine learning models especially when there is class imbalance within the dataset. It should be noted that the accuracy measure cannot indicate the true worth of the model because it ignores the balance ratio between the classes. This situation is referred to as the accuracy paradox, whereby a model may attain high accuracy scores through prediction of the majority class. In most cases, this approach leaves out the less represented class in favour of the dominant class. For instance, in an intruder detection system, attacks will remain unnoticed if the majority class dominates.

Most machine learning algorithms assume balanced data sets, meaning they will focus on the majority class, thereby ignoring the minority class, which, in most cases, is intrusions/attacks. This problem is referred to as class imbalance, whereby the machine learning model can be highly accurate yet fail to detect important attacks (Vargas, Aranda, Costa, Pereira, & Barbosa, 2023). Therefore, a set of complementary metrics is used that capture different aspects of the model's strengths and weaknesses. Below, the most commonly used performance metrics in Azure AutoML and their relevance in the context of this study are presented.

2.8.1 Classification Outcomes: TP, TN, FP, And FN

In order to interpret the metrics of performance correctly, one needs to know about the four fundamental outcomes in the context of classification models, which are discussed by Grigorev (2021).

- True Positive (TP): The output that the model classifies as being a true positive (e.g. an attack), is indeed so.
- True Negative (TN): The model predicts a negative outcome (e.g. benign traffic) and this is correct, i.e. normal traffic that is correctly classified.
- False Positive (FP): The model classifies benign traffic as an attack, a false alarm in other words.
- False Negative (FN): A real attack is classified as benign, the threat has been completely missed.

In other words, for the purposes of cybersecurity, the right classification should not have any false positives (FP) or false negatives (FN). The four criteria are the basic ones used when evaluating any classification measure discussed below.

2.8.2 Accuracy

Accuracy describes the rate at which classification is done correctly relative to the total number of cases. This measure ranges from 0 to 1, with a higher accuracy approaching one

(Microsoft, 2025). To give an example, if there are 100 predictions made by a classifier and 90 of them are correct (correspond to the label of the data point), then the accuracy equals 0.90 or 90%.

Although the metric is easy to interpret, it can give a false impression of the model's performance in scenarios with unbalanced data. In intrusion detection, where normal traffic often constitutes a clear majority, a model can achieve high accuracy by simply classifying all traffic as normal despite completely failing to identify attacks. Accuracy is calculated using the following formula: (Uddin, 2019).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2.8.3 Precision

Precision measures a model's ability to avoid falsely classifying negative examples as positive. The range is [1,0] and the closer to 1, the more secure the model is (Microsoft 2025). The formula below is the general calculation for precision (scikit, 2025).

$$\text{Precision} = \frac{TP}{TP + FP}$$

2.8.4 Recall

Recall measures a model's ability to find all true positive examples. It shows how many of the actual positive cases the model managed to find. The interval here is also between [1,0] where the closer to 1 the model hits, the better it is at not missing positive examples (Microsoft, 2025). The computation goes thus (scikit, 2025):

$$\text{Recall} = \frac{TP}{TP + FN}$$

The higher the recall, the more appropriate it is when there is a need for no threat to be missed out, despite there being lots of false alarms. This is the situation in cybersecurity since each threat that remains undetected may lead to dire consequences. Such a model will hence be beneficial since it reduces the amount of false negatives – attacks that are missed

by the model. This explanation is justified by the analysis of the different approaches used in evaluating classification models provided in Chapter 4 in the book Machine Learning Bookcamp. “Recall is the fraction of correctly classified positive examples among all positive examples...The higher the recall, the fewer false negatives we have.” (Grigorev 2021, Chap.: Evaluation Metrics For Classification, Paragraph: Precision and Recall).

The book uses a customer churn prediction problem as an example, but the principle is general: high recall means that the model is better at identifying the positive, important observations, which in security-critical environments corresponds to real threats.

2.8.5 F1 score

F1 score is the harmonic average of Precision and Recall, giving weight to both of them, which becomes very important in imbalanced data where both False Positives and False Negatives are harmful. F1 score doesn’t consider True Negative and can be used when detecting threats is more critical than reducing False Positive (Microsoft, 2025). Formula for the calculation of the F1 score is given below:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.8.6 AUC – Area Under the ROC Curve

AUC stands for “area under the curve,” showing a model’s ability to separate one class from another at any threshold. The area under the ROC graph is taken in this case, in which the dependency of True Positive Rate on False Positive Rate is plotted, see Figure 2, and falls in the interval [0,1]. When the value approaches 1, we have perfect separation; whereas, the value around 0.5 shows that the model performs even worse than random guessing.

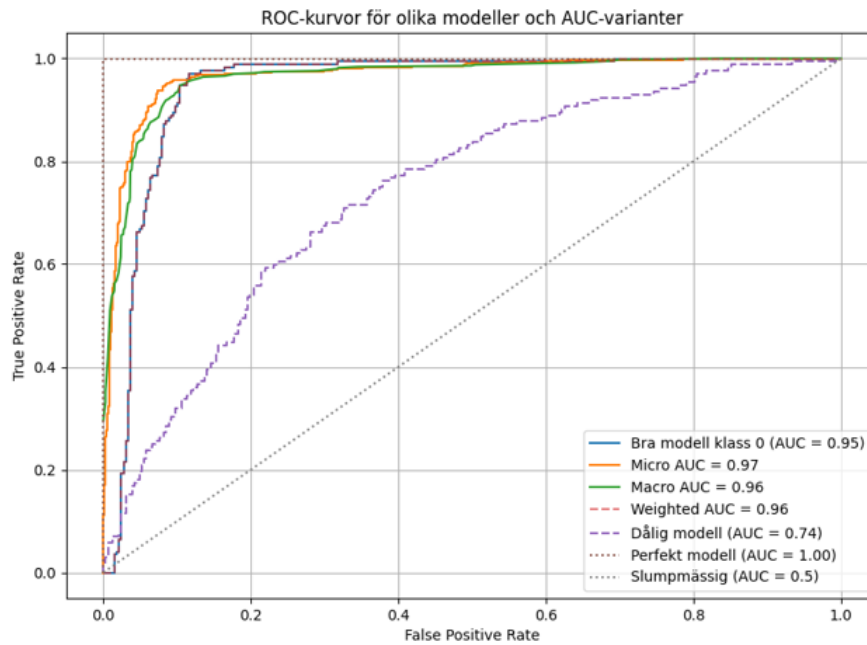


Figure 2: ROC curves for different models and AUC variants. A model with a high AUC (closer to 1.0) has good separation ability between classes, while a model close to 0.5 performs like chance

2.8.7 Confusion Matrix

The confusion matrix offers an in-depth view of the results of classification of the model. It indicates how many instances belonging to the actual class (the y-axis) have been classified under another class (the x-axis) by the model. The ideal situation for such a model is where most observations lie on the diagonal line indicating that the prediction and actual class match (Microsoft, 2025). Figures 3 and 4 demonstrate how such a matrix appears in both cases.

The term "confusion" in relation to the confusion matrix comes from the model's tendency to confuse classes (Microsoft, 2025). This could be exemplified in the case of an intrusion detection model where the true positive would indicate that an attack has been correctly detected and the false negative means that a real intrusion was left undetected.

The confusion matrix is therefore a central tool in the evaluation of models in safety-critical applications.

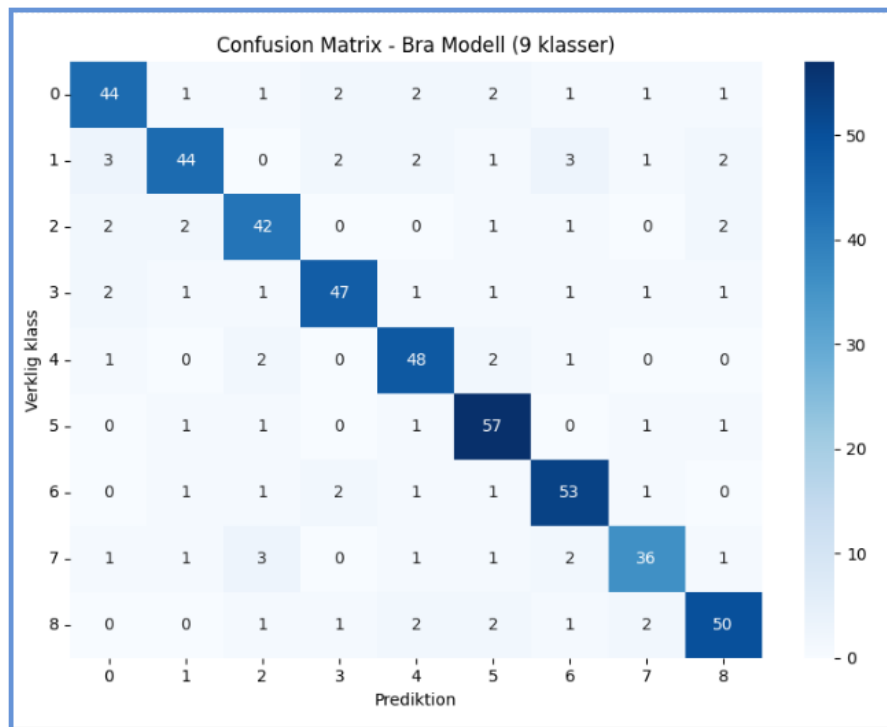


Figure 3: Confusion matrix for a well-trained model. Most observations lie along the diagonal, indicating good accuracy

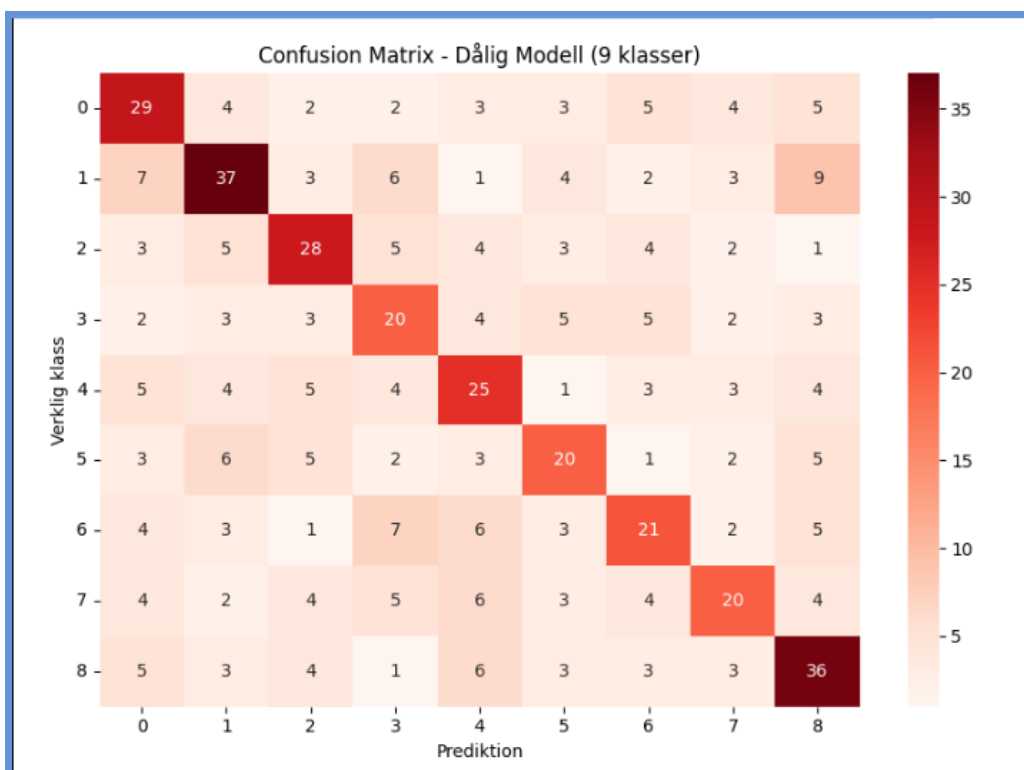


Figure 4: Confusion matrix for a weak model. Many misclassifications occur, which is visible in the off-diagonal distribution

2.8.8 Macro, Micro And Weighted Calculations

For multi-class problems or unbalanced data, three common strategies are used to summarize results (Microsoft, 2025), see a summary in Table 1.

Table 1: Macro, Micro and Weighted calculations

Metric	Description	Application/Context
Macro	Calculates the metric for each class and averages them	When each class, regardless of size, should affect the result equally, especially for unbalanced data
Micro	Treats each individual prediction (in all classes) as a single large binary problem. Adds together all TP, FP, FN, TN.	When the interest is in the overall performance.
Weighted	Calculates the metric for each class, but weights each class according to how many true examples it has.	When classes occur in different frequencies.

Azure AutoML automatically calculates several of these combinations, for example macro F1, micro precision, or weighted recall, depending on the selected evaluation strategy.

2.8.9 Choosing A Primary Metric In Azure AutoMLml

When configuring an AutoML job in Azure, a primary metric is chosen for which the model should be optimized. The choice should be adapted to the purpose of the analysis. Table 2 is inspired by Microsoft Learn's documentation on automated training in Azure Machine Learning, where different primary evaluation metrics are linked to use cases (Microsoft, 2025).

Table 2: Primary values in AutoML

Metrics	Application/Examples
Accuracy	Image classification, balanced datasets, sentiment analysis
AUC-weighted	Fraud detection, image classification, anomaly and spam detection, anomaly detection
Average precision score-weighted	Text classification, especially when interpreting sentiment in texts
Norm-macro-recall	Identification of unusual events, e.g. customer churn in unbalanced datasets
Precision-score-weighted	Text classification, e.g. topic analysis or identification of subjectivity in content

In this study, recall, F1-score and AUC are mainly used as metrics, since the goal is to maximize the detection of real threats (minimize false negatives) without overloading the system with false positives.

2.9 CLASSIFICATION

Classification is a form of supervised learning where the goal is to predict a discrete output variable based on input data represented as feature vectors. In this method, a model is trained using labeled data, where each data point consists of a pair (x_i, y_i) where x_i is a feature vector (input) and y_i is a category (output). The next step of the algorithm is training to map an observation into one of the specified classes in the output space. In contrast to regression where the target value is continuous, the classification model will produce an outcome that is comprised of a finite number of classes. Examples of classification applications include but are not limited to email spam filtering, image detection, and various network threat detection (Li, Lin, & Zeng, 2024).

2.9.1 Multiclass Classification

Multiclass classification is considered one area of machine learning where the classification algorithm aims to categorize more than two classes within a data set. Unlike in the binary

classification problem where the output of the classifier must belong to only two categories, for example, "attack" or "benign," the multiclass classifier attempts to classify the input into at least three categories. Multi-class classification plays a vital role in cybersecurity, especially during the real-time detection of IoT attacks. In this respect, multi-class classification helps to differentiate among various classes of malicious data, such as DDoS attacks, port scan, and unknown attacks, and normal data. One of the most popular techniques that can be utilized here is One-vs-Rest (OVR) classification because of its ability to create multiple binary classifiers. Each binary classifier corresponds to each class versus the remaining classes.

2.9.2 Classification Algorithms In Auto ML

The algorithms included in AutoML's classification models are presented in Table 3 (Microsoft, 2025).

Table 3: Classification models in AutoML

AI-Algorithm
Logistic Regression
LightGBM
Gradient Boosting
Decision Tree
K-Nearest Neighbour
Linear SVC
Support Vector Classification (SVC)
Random Forest
Extremely Randomized Trees
XGBoost
Naïve Bayes
Stochastic Gradient Descent (SGD)

There are other algorithms besides the ones mentioned above, such as the meta-algorithm VotingEnsemble in AutoML, which is not a machine learning model but rather an aggregate model consisting of several base models to enhance prediction accuracy. Ensemble models

are created during the last phase of the autoML process, and Azure uses Caruana's ensemble selection algorithm to pick the base models for the ensemble. The algorithm creates the initial ensemble from the best five models up to five models whose score is within 5% of the best score.

More models will then be added one by one, and each time, an improvement in accuracy is considered. VotingEnsemble is thus iteratively optimized based on which combinations give the best results (Microsoft, 2025).

2.9.2.1 Caruana Ensemble Selection Algorithm

To improve the performance of machine learning models, the Ensemble Selection method uses a library of several different models, trained with varying algorithms (such as SVM, KNN, ANN, decision trees, etc.) and different hyperparameter settings. The goal is to create a diversified set of models where some perform well on different metrics such as accuracy, F1 score or ROC-AUC. Through a so-called forward stepwise selection strategy, the models are selected that at each step improve the overall performance of the ensemble model according to a specific metric, such as accuracy or ROC area. The selection is done iteratively by successively adding the models that best complement the already existing ensemble. Initially, the process often starts with the five best-performing models to minimize the risk of poor starting performance. Each new model added to the ensemble must increase its overall performance to be included (Caruana, R., Niculescu-Mizil, A., Crew, G. & Ksikes, A., 2004).

To reduce the risk of overfitting, the method has been improved by three techniques: (1) selection with replacement, which allows weighting of particularly strong models, (2) initialization with a selection of the best-performing models before starting the greedy selection, and (3) bagging of the ensemble selection, where several sub-ensemble are created from random subsets of models and their average is used as the final result. Studies show that ensemble selection often outperforms both the individually best-performing models and other ensemble methods such as bagging, boosting, and Bayesian model aggregation (Caruana et al., 2004).

2.10 DATA SCALING AND PREPROCESSING IN MACHINE LEARNING

In machine learning, data preprocessing is a central part of model development, especially when variables have different scales. One technique that could be used in such a case would be standardization, which involves scaling of the variables in such a way that their mean is zero and standard deviation is one. Such a practice increases the efficiency of scale-sensitive algorithms like SVM, KNN, and gradient descent algorithm. The Azure Machine Learning framework uses `StandardScalerWrapper` for achieving this purpose (Microsoft, 2025). Apart from standardization, there may be other methods applied based on data. Data can be scaled to the range $[-1,1]$ by using `MaxAbsScaler`. `SparseNormalizer` is mainly used when the data is sparse, such as in cases with many zero values, e.g. in log data and normalizes each row so that its length is 1. The application of `TruncatedSVDWrapper`, which retains the most important features from high-dimensional matrices without centering the input data, may be considered an attempt to reduce the number of features used in analysis to enhance the model performance and interpretation. Particularly, this may prove beneficial for applications involving a huge amount of input data (Ahsan et al., 2022).

2.11 PREVIOUS RESEARCH ON AI IN NETWORK SECURITY

It has been well proven that artificial intelligence (AI) and machine learning (ML) play crucial roles in ensuring network security and implementing various intrusion detection approaches, anomaly-based detection, and traffic flow pattern recognition. With the growing number of threats posed to modern networks, there has been a growing need for efficient approaches aimed at ensuring security by leveraging the use of AI and ML (Ahsan et al., 2022; Zuech, Khoshgoftaar & Wald, 2015).

2.11.1 Use Of CICIDS2017 In Previous Studies

CICIDS2017 is one of the most established and frequently used benchmark datasets in intrusion detection research. CICIDS2017 dataset stands out with its ability to provide realistic network traffic within a controlled environment, which consists of pre-classified data as either normal or malicious. The diversity of traffic data and large amount of data can

be considered useful in testing both supervised and unsupervised machine learning models (Zhao, Hu, & Liu, 2023.; Khraisat et al., 2019).

In addition to the diverse data presented in CICIDS2017, there is also a considerable number of statistical properties related to the network, namely, 79 per flow, that allows using CICIDS2017 in testing machine learning models on auto ML services like Azure ML without spending too much effort on feature engineering (Sharafaldin, Lashkari, & Ghorbani, 2018). As reported by Zhao, Hu, & Liu (2023.), various deep learning models were tested on CICIDS2017, and the LSTM models demonstrated high efficiency in testing. These networks showed to be especially effective for working with sequential data where timing and order are important aspects, e.g., for detecting escalation of attacks.

Firstly, according to Ahsan et al. (2022), CICIDS2017 can be regarded as an essential element in the testing process aimed at verifying the effectiveness of developed artificial intelligence systems for network security. Nevertheless, according to Khraisat et al. (2019), the issue of the existing skew between types of attacks becomes apparent. The problem may hinder the model's detection of infrequent types of attacks, so certain measures should be employed, including oversampling, using weighted losses, or creating synthetic data. Secondly, it should be noted that some researchers have pointed out specific limitations of CICIDS2017 due to the fact that the data in question was gathered in laboratory conditions. The fact that attacks were performed in a relatively short period of time and their characteristics were predetermined means that models that rely on CICIDS2017 may experience certain difficulties while detecting prolonged or more complex attacks in real life (Zhao, Hu, and Liu, 2023).

2.11.2 Models That Have Shown High Accuracy

As reported in multiple studies, there is a high precision and recall in using different ML approaches for classifying the network traffic on the basis of CICIDS2017 dataset. As indicated by Ahsan et al. (2022), classic ML techniques like Random Forest and XGBoost prove their superiority in detecting attacks with distinct signature, i.e., DDoS, PortScan and Brute Force, where in some cases the accuracy rate exceeds 95%.

According to Belal & Sundaram (2022), one of the strengths of the ensemble methods is their good generalization capacity. Additionally, it has been claimed that hybrid approaches such as those utilizing the CNN-LSTM approach are particularly effective in recognizing spatial and temporal patterns, which may prove to be very beneficial for attacks with several phases.

Although there are many advantages of modern methods like CNN-LSTM models, they are quite complex to implement due to their high computational demands and need for specialized skills. Thus, simpler approaches such as Decision Tree or Logistic Regression remain relevant because of their ability to offer faster deployment and easier implementation, even if this means sacrificing predictive performance (Khraisat et al., 2019).

Consequently, previous research has proven that there is not one solution that may be referred to as the best model for intrusion detection as it varies based on many factors, including the type of threat, resources at hand, the requirement for model interpretability, and others. With this said, several authors recommend taking an automated approach and employing such technologies as Azure AutoML which offers a systematic approach in selecting the optimal model based on set criteria (Microsoft, 2025; Ahsan et al., 2022).

2.12 PROBLEMS RELATED TO USING ARTIFICIAL INTELLIGENCE IN THREAT DETECTION

Although artificial intelligence can improve the efficiency of threat detection significantly, the technology itself poses a number of challenges. Some of those challenges include the problem of data imbalance, lack of model interpretability, and poor generalization capacity of models.

2.12.1 Bias In Training Data

AI-based detection models depend on training on datasets that represent the variability of cases that happen in actual networks. Should one group of attacks or traffic be present in abundance in the training set, whereas other groups are highly scarce, there is a possibility of building an imbalanced predictive capacity in the model. This would lead to detecting

frequent attacks with high precision, yet leaving uncommon, although critical attacks unnoticed (Khraisat et al., 2019; Zhao, Hu, and Liu, 2023).

One example of such a problem is related to popular CICIDS2017 dataset that contains some attack classes extremely rarely, such as Heartbleed and Infiltration attacks. These attacks significantly affect the recall rate for the specific attacks in the model. The problem is described by Belal and Sundaram (2022) and includes techniques like class balancing, synthetic minority oversampling technique, and ensemble learning models.

2.12.2 Lack Of Transparency And Interpretability

Another central problem when applying AI in cybersecurity is the limited interpretability of the models. More specifically, deep learning models in general have been associated with the so-called “black box” issue, making it difficult to understand the reasoning behind a certain classification choice. The presence of such issues becomes a serious challenge in security-relevant contexts when decisions have to be justified (Ahsan et al., 2022). With that in mind, a new idea called explainable artificial intelligence (XAI) has become increasingly popular among scientists in recent times.

That is, in line with systematic studies done by Mohale and Obagbuwa (2025), SHAP and LIME can act as explanation methods to help in understanding how models operate in IDSs. While the simplicity of some models such as decision trees makes it easy to understand them, their accuracy was found to be relatively low compared to other models. In effect, the problem raised by Mohale and Obagbuwa (2025) is the absence of standard tools to use XAI.

2.12.3 Generalizability To Real-World Environments

Another issue that arises with the use of such machine learning models involves the lack of their generalization to real-world conditions since real-world networks exhibit great diversity, constant change, and include legitimate anomalies, as well as novel and unknown types of cyberattacks (Khraisat et al., 2019; Avila et al., 2021). Thus, algorithms that have

proven themselves in testing environments cannot necessarily deliver similar results in practice.

Zuech, Khoshgoftaar and Wald (2015) emphasize that this lack of transferability is a major obstacle to the practical application of AI in organizations' security work. To counteract this, it is proposed to continuously retrain the models with current traffic data. But this entails the use of substantial resources such as computational power, expertise, and data management skills, which may not be available to all companies.

2.13 SUSTAINABLE DIGITALIZATION AND CLOUD INFRASTRUCTURE

With the advent of digital services, there is a greater need for computational power. However, this entails the greater demand for energy consumption in the information technology industry. According to Buyya, Ilager and Arroba (2023), data centers, the backbone of any cloud computing infrastructure, will contribute to a growing number of electricity usage. It has been predicted that by 2025, data centers worldwide will utilize 20 percent of the total electricity used in the world and produce 5.5 percent of carbon dioxide emissions.

However, cloud-based services are considered to have the potential to reduce the IT sector's climate impact compared to traditional, locally operated systems. Through resource sharing and large-scale optimization in cloud providers' data centres, opportunities for increased energy efficiency and better integration of renewable energy are created (Yenugula, Sahoo & Goswami, 2024). These solutions also support circular economy principles through, for example, hardware reuse and more efficient workload distribution.

To enable more sustainable digitalisation, organisations and system developers should consider environmental factors in their technical choices. Loeser, Erek and Zarnekow (2017) argue that sustainability principles should be integrated already in the design phase of IT services and not only considered as a secondary requirement. This includes, among other things, energy use, carbon footprint, material recycling and the ability to manage waste in a sustainable way.

CHAPTER-3

METHODOLOGY

This research employs experimental and quantitative methodologies in developing and evaluating an automatic Intrusion Detection System based on Machine Learning on the CICIDS2017 dataset. This methodological design was employed in order to perform network traffic analysis, detect malicious actions in the data, and assess the efficiency of various machine learning algorithms in the cyberattacks detection process. The research methodology consists of several stages that include data collection, preprocessing, solving class imbalance problem, modeling and evaluation of results.

The reason for the selection of the CICIDS2017 dataset as the base for this research is that it offers real network traffic data that contain various kinds of cyberattacks alongside the benign activities. The dataset was created at the Canadian Institute for Cybersecurity. This dataset incorporates modern cyberattacks such as Distributed Denial of Service (DDoS) Attacks, Brute Force Attack, Port Scan, Web Attack, Botnet Traffic, DoS Attack. The dataset comprises network flows extracted from the network traffic by the CICFlowMeter with such metrics as flow duration, packet length, byte/s, packet/s and others. Since the complete dataset includes millions of records and involves considerable computations, roughly 20% of the data set was selected using stratified sampling technique in order to maintain original class distribution and decrease computational complexity at the same time.

In preparation of the data set, several preprocessing steps have been applied prior to model training. Duplicate values, missing data and invalid values have been eliminated to increase quality of the data and stabilize training process. Infinitely high values obtained during calculation of network flows were handled properly. Classification structure of the target variable was represented by two classification scenarios. Binary classification involved grouping all kinds of attacks to one ATTACK class and comparison with the BENIGN traffic. Multiclass classification included maintaining separate attacks for evaluating model's ability to classify different attacks.

Data normalization and feature scaling was done to normalize variation within numerical features and improve performance of machine learning algorithms. Due to imbalance nature of the CICIDS2017 dataset in which majority is made up of benign traffic that exceeds several minor attacks, RandomOverSampler was applied in order to make the dataset balanced. Over-sampling involved duplicating the minority classes and varying the samples slightly to avoid biased towards majority classes.

The resulting dataset was subsequently split into training and testing datasets at the rate of 80 percent vs. 20 percent. The stratified train-test splitting was applied in order to guarantee that there was consistent representation among both datasets. While the former served as training data for machine learning models, the latter provided means of evaluating models' performance on unseen data.

In order to have something to compare AI-driven techniques against, a rule-based approach to intrusion detection was applied to simulate traditional SIEM and signature-based detectors. As such, the baseline detector determined whether the particular network activity is suspicious based on threshold-based rules which included extremely high numbers of packets transmitted, flows with suspicious duration, and unusual byte transmission. Thus, the rule-based system was introduced as a baseline system for comparative purposes.

Supervised ML algorithms applied in the course of the project included logistic regression, random forest, XGBoost and lightGBM. Logistic regression was used as a baseline technique in terms of linear classification problems. In turn, the latter two were utilized due to their outstanding ability to work with multi-dimensional and nonlinear traffic.

The resulting dataset was subsequently split into training and testing datasets at the rate of 80 percent vs. 20 percent. The stratified train-test splitting was applied in order to guarantee that there was consistent representation among both datasets. While the former served as training data for machine learning models, the latter provided means of evaluating models' performance on unseen data.

In order to have something to compare AI-driven techniques against, a rule-based approach to intrusion detection was applied to simulate traditional SIEM and signature-based detectors. As such, the baseline detector determined whether the particular network activity is suspicious based on threshold-based rules which included extremely high numbers of

packets transmitted, flows with suspicious duration, and unusual byte transmission. Thus, the rule-based system was introduced as a baseline system for comparative purposes.

Supervised ML algorithms applied in the course of the project included logistic regression, random forest, XGBoost and lightGBM. Logistic regression was used as a baseline technique in terms of linear classification problems. In turn, the latter two were utilized due to their outstanding ability to work with multi-dimensional and nonlinear traffic.

Additionally, comparative experiments were conducted for assessing the effects of applying various imbalance management approaches on the performance of the models. There was studied how models trained through oversampling outperform other models without any balancing methods in terms of better detecting attacks. Lastly, there was discussed the results obtained concerning the effectiveness assessment of machine learning-based intrusion detection systems.

3.1 DATASET EXPLANATION

The data source used in the paper was the CICIDS2017 dataset created by the Canadian Institute for Cybersecurity (CIC). The reason behind selecting this particular dataset is its ability to provide a comprehensive picture of the current traffic network scenario including both normal and attack activity. CICIDS2017 dataset is widely used in intrusion detection research area for it simulates real-life situations including attacks. Dataset link: <https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset>.

The data is collected using network flow records created through traffic capture analysis by means of the CICFlowMeter application. Every network flow is characterized by 79 statistical attributes that characterize the network traffic. Such attributes can range from flow duration, total number of packets in the forward direction, total number of packets in the reverse direction, flow-bytes-per-second, flow-packets-per-second, packet size statistics, inter-arrival times, flag counts, header sizes, active time, idle time, and many others. Attributes provide comprehensive information regarding the communication process and can be helpful in distinguishing between normal and malicious traffic flows.

CICIDS2017 not only comprises normal network traffic but also includes numerous types of cyberattacks, namely Distributed Denial of Service (DDoS) attack, Denial of Service (DoS) attack, Brute Force attack, Port Scanning attack, Web Attack, Botnet attack, Heartbleed attack, and Infiltration attack. Thus, the presence of a variety of cyberattack types allows evaluating the effectiveness of both binary and multi-class intrusion detection systems. The binary classification involves distinguishing between benign and malicious flows whereas multi-class classification distinguishes an attack type within network traffic.

The initial data set includes millions of traffic samples gathered for several days with simulated network operations. Because of the very large number of data and processing difficulties, about 20% of the total amount of data was sampled by the stratified method. The stratification of the data helped preserve the original ratio of benign and attacked classes while simplifying computations.

The data set is characterized by an evident imbalance between benign traffic and some attack classes on one hand, and uncommon attacks like Heartbleed or Infiltration on the other hand. Imbalance makes it difficult to train models since machine learning algorithms may get biased toward majority classes and ignore minority classes of attacks. Thus, balance procedures should be performed to enhance results.

The CICIDS2017 database is an appropriate benchmark for the evaluation of intrusion detection systems based on the realistic generation of traffic and the inclusion of various attack scenarios, allowing for an effective detection of not only well-known but unknown attacks using machine learning techniques.

3.2 PREPROCESSING

Data preprocessing is an essential step when working with intrusion detection based on machine learning as raw data might contain inconsistencies, repetitions, missing fields, and noise affecting the performance of the algorithms. For that reason, the following pre-processing steps have been conducted to prepare the CICIDS2017 data set for training.

Firstly, the data set has been loaded into the analysis environment and checked for missing fields, repetitions, and incorrect numeric fields. Then, all duplicate records were excluded to avoid a possible bias in training algorithms and enhance their generalizability. Finally, missing and undefined fields were detected and treated appropriately.

The data set had some infinities due to network flow calculations especially on packet rates and byte transmission rates. The infinities were converted to missing values which were then dropped to ensure stable model training during computation. This was necessary to make sure that there were only valid numeric values of the features in our data.

From this cleaned data set, two classification labels were made for the target. In the binary class labels, all attacks fall under ATTACK class while normal is BENIGN class. On the other hand, in the multiclass class labels, attacks have their own labels as it helps in identifying the type of attack. Categorical target labels were then encoded using label encoding.

Feature scaling and normalization were used to decrease variance between numerical features and help the models converge. Considering that some of the numerical features of the dataset are characterized by very different values, scaling would allow not letting one of the features affect the learning process too much.

Next, class imbalance in the dataset was considered. Some attack classes include far fewer examples than benign data and the majority of attacks. Therefore, oversampling technique, RandomOverSampler, was used on the training set in order to duplicate examples from the minority classes and balance them out.

Feature scaling and normalization were used to decrease variance between numerical features and help the models converge. Considering that some of the numerical features of the dataset are characterized by very different values, scaling would allow not letting one of the features affect the learning process too much.

Next, class imbalance in the dataset was considered. Some attack classes include far fewer examples than benign data and the majority of attacks. Therefore, oversampling technique, RandomOverSampler, was used on the training set in order to duplicate examples from the minority classes and balance them out.

3.3 ALGORITHM WORKFLOW

The suggested intrusion detection system utilizes a supervised machine learning approach that can be used to automatically detect malicious behavior in network traffic using the CICIDS2017 dataset. This approach is based on a series of steps such as dataset acquisition, pre-processing, imbalance handling, machine learning model training, and attack classification among others. Overall, this process was performed by utilizing machine learning libraries in Python as well as Microsoft Azure AutoML.

Step 1: Dataset Acquisition

Initially, the task was to acquire the CICIDS2017 dataset, which is a collection of network traffic flows consisting of both normal traffic and attacks. Specifically, this dataset includes statistics on flows that were gathered using the CICFlowMeter tool. The data was imported to the analysis workspace using data processing libraries such as Pandas and NumPy.

Step 2: Data Sampling

Since the original dataset consists of several millions of data entries, analyzing the entire data will need substantial computation capability. In an attempt to reduce computation cost but maintaining data class ratio at the same time, 20% of data was sampled using stratified sampling. Stratified sampling ensures equal proportions of benign traffic and attacks in the sampled data.

Step 3: Data Cleansing and Preprocessing

A number of data preprocessing methods have been applied before fitting the datasets into the model.

Data preprocessing included:

- Duplicate data elimination
- Missing value identification and imputation
- Infinite/invalid value elimination

- Data normalization
- Target encoding

However, infinite values that arose in the computation of the flows have been substituted and filtered out to avoid possible instability in the machine learning training phase. The target labels were further encoded using two classification methods:

- Binary classification (BENIGN vs ATTACK)
- Multiclass classification (each attack category)

Feature scaling and normalization methods were applied as well to guarantee numerical uniformity of the traffic data.

Step 4: Managing Class Imbalance Problem

CICIDS2017 suffers from extreme class imbalance since benign traffic data and some attack classes include significantly more instances than other attacks such as Heartbleed and Infiltration attacks.

To manage the problem of class imbalance, RandomOverSampler has been applied in the training process. Over-sampling involved duplicating instances from minority classes to ensure adequate representation among all classes. Over-sampling contributed to avoiding classifier bias against majority classes and improved detection of rare attacks.

Step 5: Train-Test Split

After data processing, the dataset was then subjected to a train-test split, with the 80/20 split ratio being applied.

- 80% of the data would be used for training the model
- The other 20% of the data would be used for testing and validating

Stratified train-test split technique would be used to ensure that the class balance would remain consistent throughout both data subsets.

Step 6: Rule-Based Intrusion Detection as a Baseline

Prior to training machine learning models, rule-based detection was performed to simulate the behavior of a traditional intrusion detection system (IDS).

A rule-based intrusion detector would identify unusual packets in terms of such threshold parameters as:

- Very high packet transmission rate
- Abnormal flow duration value
- Suspicious byte transfer rate
- Too many connection attempts

This baseline algorithm provided for comparison purposes for the performance evaluation of other models.

Step 7: ML Models Training

Supervised machine learning algorithms were trained based on the preprocessed data.

Logistic Regression

Logistic Regression was implemented as a baseline algorithm for linear classification of malicious traffic versus benign traffic.

Random Forest

Random Forest algorithm uses several decision trees to enhance accuracy of prediction and avoid overfitting.

XGBoost

XGBoost is a gradient boosting model that could learn complicated traffic patterns through iterative error reduction. The algorithm is famous for its high efficiency and good predictive power.

LightGBM

LightGBM was applied in both binary and multiclass intrusion detection problems where its efficient gradient boosting approach allowed fast model training without any loss in classification accuracy.

Step 8: Performance Metrics for Model Evaluation

- The trained models were analyzed based on various performance metrics as a way of evaluating the models' ability to perform intrusion detection.
- The performance metrics included:

Accuracy

Precision

Recall

F1-score

ROC-AUC

Confusion matrix

Precision-recall curve

Step 9: Feature Importance Analysis Using Ensemble Models

Feature importance analysis was performed using machine learning algorithms such as Random Forest and LightGBM.

The important traffic features were selected from those that influenced intrusion detection and they include:

- Flow bytes per second
- Flow packets per second
- Packet length features

- Flow duration
- Packet inter-arrival times

This made it possible to determine which features of traffic were most related to the attacks.

Step 10: Multi-Class Attack Classification

Besides binary attacks, several multiclass attacks were also tested to see whether specific categories of attacks could be detected.

The multi-class attacks include:

- DDoS
- DoS
- PortScan
- Brute Force
- Botnet
- Web attacks
- Infiltration attacks

The main idea of the multi-class classification was to evaluate how efficient the AI-based IDS could be in different situations.

Step 11: Performance Comparison

Finally, all machine learning algorithms were compared to the rule-based method for detection of cyber-attacks.

The performance comparison was based on the following criteria:

- Detection accuracy
- False positives
- Attack detection ability

- Class imbalance effects
- Computational complexity

It became evident that machine learning-based intrusion detection systems were able to detect attack patterns more accurately than the rule-based system.

CHAPTER-4

RESULTS

From the conducted experiment, one can conclude that AI-based intrusion detection systems perform very well when it comes to recognizing attacks present in network traffic. Several supervised machine learning models have been trained and tested on CICIDS2017 data, and then compared to a rule-based IDS used for solving similar cybersecurity issues. As seen from the received results, a substantial boost in detection precision and classification ability can be provided by an intelligent approach.

Both preprocessing and balancing have significantly impacted model results. Once the data had been cleaned, imbalanced, and prepared, it became evident that machine learning algorithms worked much better at recognizing both dominant and rare attack classes. It is especially worth noting that the application of RandomOverSampler helped increase recall rate of underrepresented attack types which previously lacked precision.

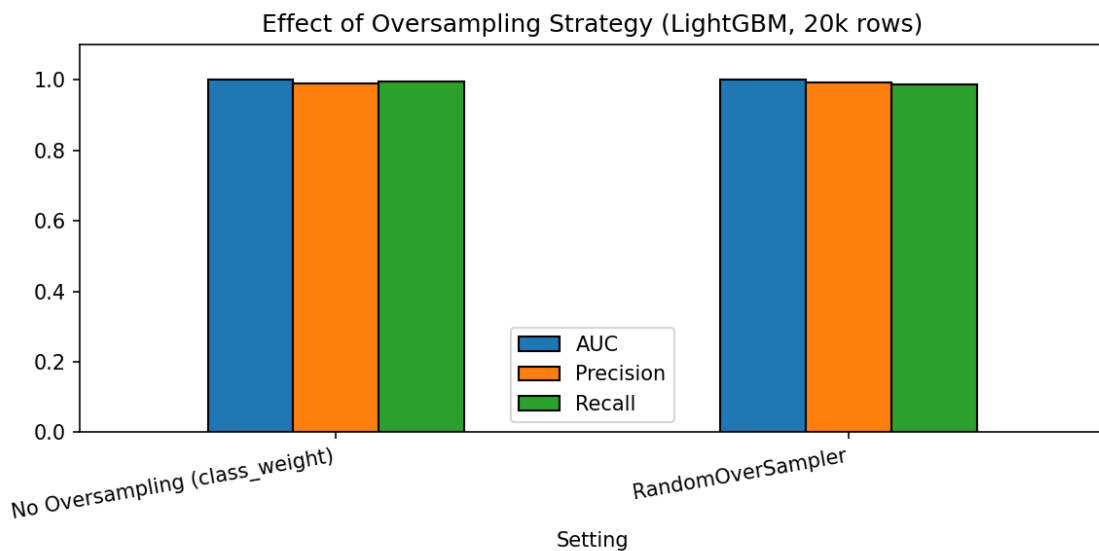


Figure 5: Effect of oversampling strategy

The baseline Intrusion Detection System using rule-based approach proved to be ineffective due to its dependency on pre-defined threshold criteria and traffic patterns. Despite being able to detect certain anomalies in traffic patterns, this baseline approach could

not identify more sophisticated patterns found in the data set. The baseline approach was characterized by higher numbers of false positives and false negatives than the machine learning approaches.

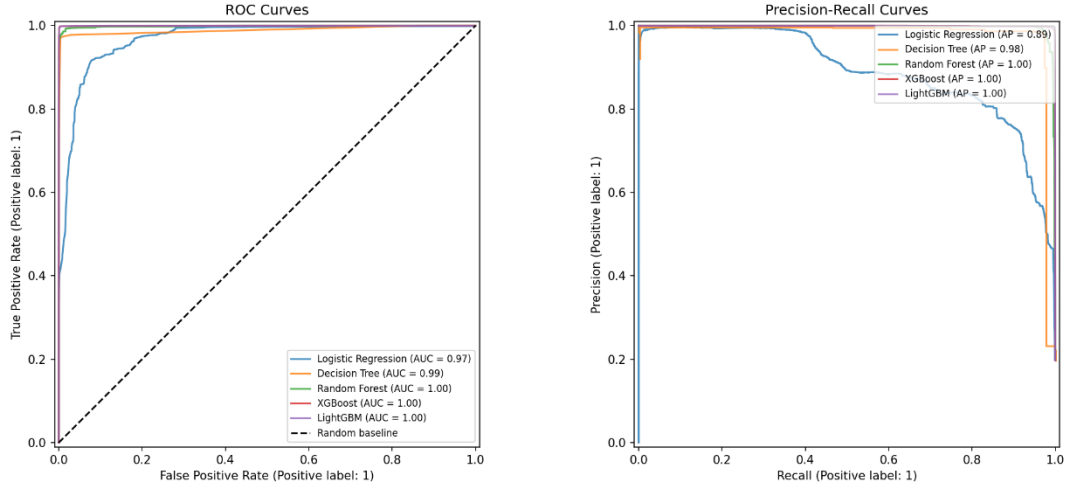


Figure 6: ROC curves and Precision-Recall curves

The graph below shows the classification performance comparison for five machine learning models including Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM. In the first subplot on the left-hand side, the ROC curves are illustrated. The True Positive Rate (TPR) is shown on the y-axis versus the False Positive Rate (FPR) on the x-axis, while the diagonal dotted line indicates the random model's performance. In terms of their AUC scores, Random Forest, XGBoost, and LightGBM obtained a perfect score of 1.00, Decision Tree obtained a close-to-perfect score of 0.99, whereas Logistic Regression scored a score of 0.97.

The right subplot shows the Precision–Recall (PR) curves, which evaluate the trade-off between precision and recall for the same models. In regard to their Average Precision scores, the Random Forest, XGBoost, and LightGBM models performed exceptionally well with a perfect AP of 1.00, while the Decision Tree model managed a good AP score of 0.98. Logistic Regression performed poorly compared to other algorithms in achieving a relatively high AP score of 0.89. Therefore, it can be observed that ensemble models, including Random Forest, XGBoost, and LightGBM, perform better than Decision Tree and Logistic

Regression.



Figure 7: Multi-class confusion matrix

In the following diagram, the multi-class confusion matrix of attack classification is presented for the network intrusion detection system. Here, the rows refer to the true class labels of the attack category, while the columns denote the predicted labels provided by the classifier algorithm. The cells along the diagonals show the number of correctly classified samples, whereas the other cells exhibit misclassified samples. Darker shading signifies greater number of samples, as indicated by the color scale at the right. The classification model shows an excellent performance rate for the BENIGN class with 68,232 classified instances.

However, some samples categorized as BENIGN fall under the wrong attack categories such as DoS Hulk, PortScan, and Web Attacks – Brute Force. The DoS Hulk and PortScan attack types comprise many samples, while there are some instances of their confusion with the BENIGN category. Small attack categories such as Bot, DoS Slowhttptest, DoS slowloris, SSH-Patator, and Web Attack – XSS have fewer samples and rare cases of being mislabeled.

From the overall perspective, the confusion matrix reflects the fact that although the algorithm faces difficulties in detecting specific types of attacks from the other traffic classes, it possesses sufficient capabilities to successfully differentiate most of the network traffic categories. The chart demonstrates the aspects related to the accuracy of the predictions for each class.

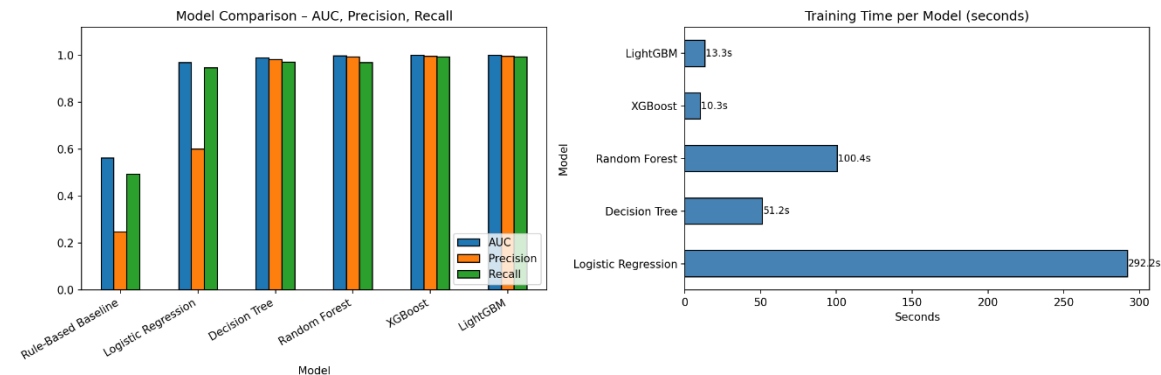


Figure 8: Model Comparison

This figure illustrates the comparison between various machine learning techniques using performance metrics and training time. The first subplot on the left side provides a comparison among the following six models through their Area Under the Curve (AUC), Precision, and Recall values: Rule-Based Baseline model, Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM. Among these models, the Rule-Based Baseline model shows the least performance, and Logistic Regression attains the best AUC value but a comparatively low precision value than other machine learning techniques. The Decision Tree, Random Forest, XGBoost, and LightGBM techniques consistently deliver outstanding performance in all three categories.

The second subplot shows the amount of time taken for training each algorithm, in seconds. The Logistic Regression model took the most training time, which was around 292.2 seconds, whereas Random Forest took 100.4 seconds and Decision Tree took 51.2 seconds. On the other hand, the training time for XGBoost was approximately 10.3 seconds and for LightGBM was around 13.3 seconds. It can be concluded that the gradient boosting algorithms are not only more accurate than others but also efficient.

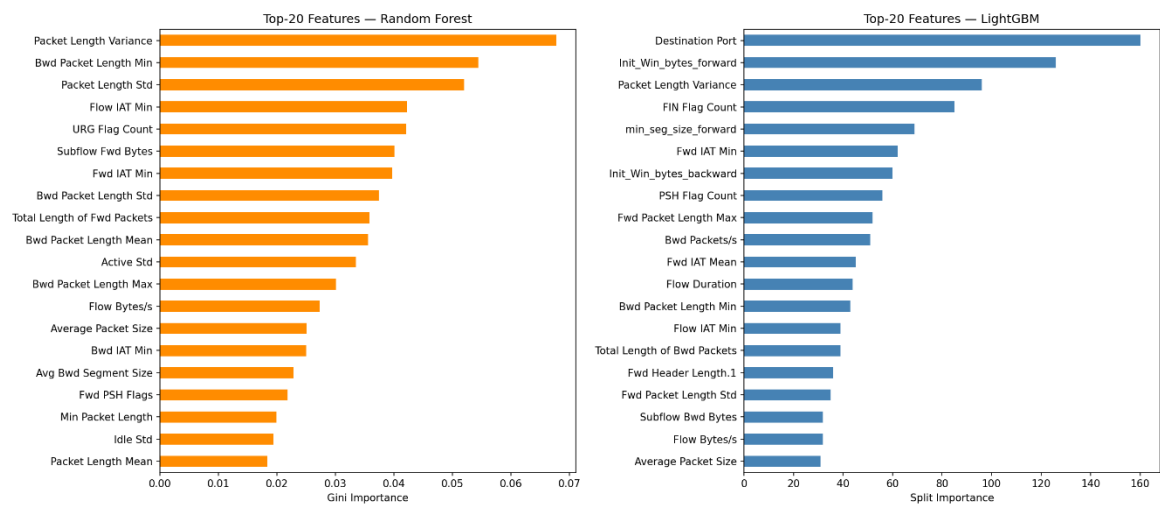


Figure 9: Features

The figure above shows the top 20 most significant features extracted from both the Random Forest and LightGBM algorithms for detecting network intrusions and classifying network attacks. Feature Importance analysis plays a critical role in understanding which network traffic features affect the prediction process the most.

The left graph shows the importance scores of all features calculated by using the Random Forest algorithm. As can be seen in the graph, Packet Length Variance turns out to be the most important feature among all, followed by Bwd Packet Length Min, Packet Length Std, and Flow IAT Min. Some other important features include some statistics-based features from traffic flows like packet lengths, inter-arrival times, and byte counts in packets.

The right subplot illustrates the importance of features as generated by the LightGBM classifier. Here, the dominant feature is Destination Port, followed by Init_Win_bytes_forward, Packet Length Variance, and FIN Flag Count. Other important

features include various forward/backward packet metrics, flow duration, and other flag-related metrics. As opposed to Random Forest, LightGBM pays closer attention to protocol and connection-related features.

Generally speaking, from the above illustration, it is evident that both models view packet length parameters and flow-related metrics as the main criteria for intrusion detection. At the same time, the differences in feature importances reveal the differences in the learning patterns generated by these two models.

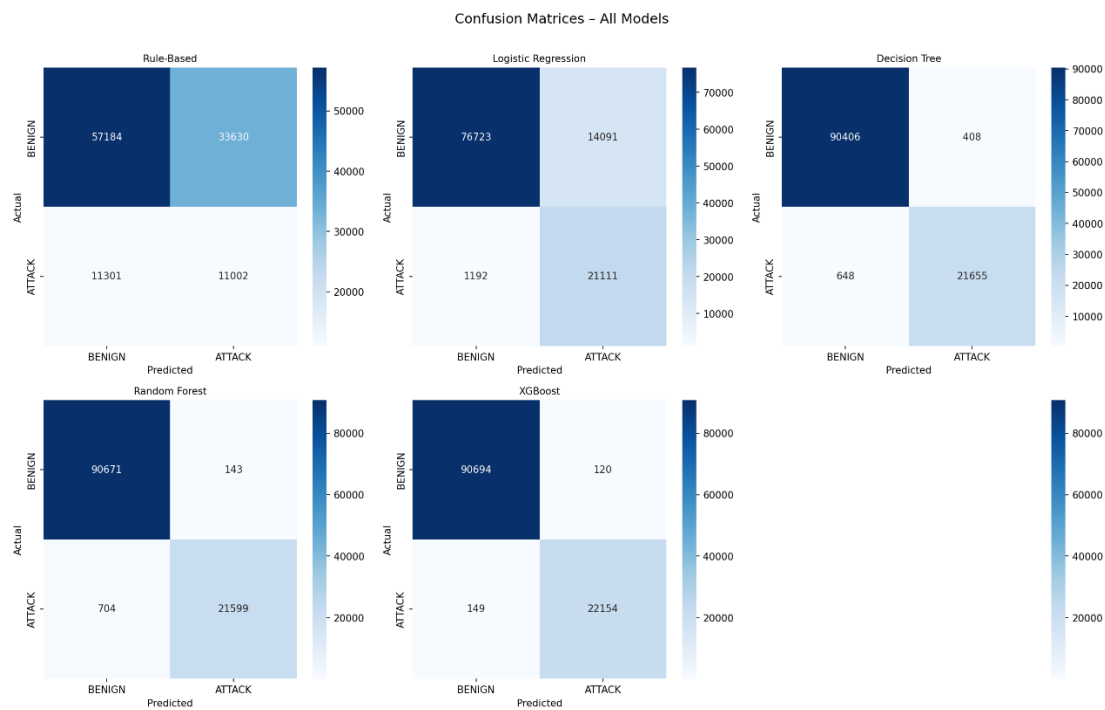


Figure 10: Confusion Matrices

Figure above shows the confusion matrixes of the five classification models namely Rule-based Baseline, Logistic Regression, Decision Tree, Random Forest, and XGBoost used for binary classification intrusion detection, where the network traffic can be either class BENIGN or ATTACK. The confusion matrix illustrates the connection between actual classes and predicted classes with the diagonals showing the correct classification and non-diagonals showing the wrong classification, where darker color intensity signifies a higher number of samples.

Rule-Based classifier model shows lower performance compared to other classifiers, with many benign samples being wrongly classified as attacks and also many attacks being classified as benign traffic.

The Decision Tree model achieves substantially higher accuracy, with very few misclassifications compared to the previous models. The algorithm Random Forest boosts the model by reducing the cases of misclassification to achieve very high accuracy in separating benign and malicious traffic. Of all the models, the one that performs the best is the XGBoost with the least number of errors in classification.

Overall, the graph shows an increase in effectiveness in detecting intrusions from simpler to more complex models. Ensemble methods like Random Forest and XGBoost have improved classification, making them more reliable in detecting malicious activities within the network.

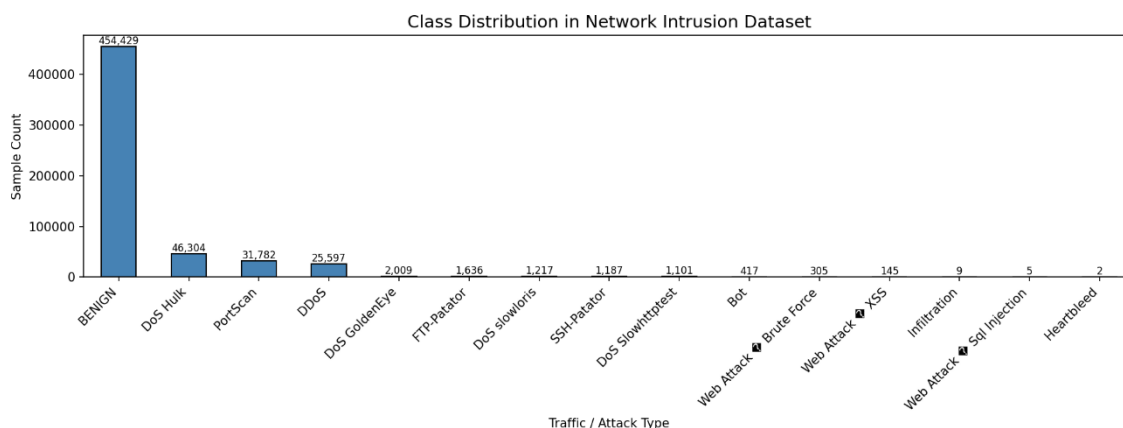


Figure 11: Class Distribution

As depicted in the figure below, there is an obvious imbalance between various classes in the Network Intrusion Dataset. The largest number of samples belongs to the BENIGN class (454,429), which is obviously the largest of all attack classes. On the side of the malicious traffic classes, DoS Hulk (46,304), PortScan (31,782) and DDoS (25,597) stand out as they include quite high numbers of data, while some classes such as Heartbleed (2), SQL Injection (5), and Infiltration (9) have very low data values.

The best results have been achieved through the application of ensemble methods such as Random Forest, XGBoost and LightGBM. All these models exhibited good capacity to learn non-linear relationships among the different network traffic features, and could classify normal and abnormal patterns accurately.

Logistic regression algorithm performed relatively well in classification but it did not produce better results compared to ensemble learning algorithms due to its inability to model complex patterns of network traffic.

Random Forest performed with high accuracy while showing robust resistance to overfitting due to its decision-tree-based ensemble architecture. It coped efficiently with the problem of processing numerous input features and performed stably in terms of different types of attacks.

XGBoost additionally improved intrusion detection system performance by continuously minimizing classification errors during the training process. It performed with high precision and recall rates, which proves its high capability of detecting attacks without false positives. LightGBM proved to be the most performant algorithm out of all considered approaches. Its gradient boosting framework ensured efficient and highly accurate training process while maintaining a relatively low amount of computation. LightGBM proved its high performance on multiclass attack classification task by performing well at classifying several types of attacks into distinct classes.

As it was shown by evaluation metrics, machine learning approaches were characterized by high accuracy, precision, recall and F1-score values. High values of precision imply relatively small number of false positive cases, whereas recall demonstrates the ability to detect attacks correctly. High F1-score indicates that precision and recall of the models have been properly balanced, which is crucial in this application.

The confusion matrix analysis further substantiated the efficiency of machine learning models by showing that most of the benign and malicious traffic samples were classified accurately while only a few attacks samples suffered from classification mistakes. The ensemble learning models showed substantially less classification mistakes than Logistic Regression and rule-based detection system.

The ROC-AUC analysis showed high capability of separation between benign and malicious traffic. The ensemble learning models generated the highest possible area-under-curve values indicating their ability to differentiate attack traffic on different decision thresholds. At the same time, precision recall graphs indicated that despite having problems of dealing with imbalanced classes, the models performed quite stably at that point.

Multiclass classification analysis showed that the developed models can recognize multiple categories of attack with good accuracy. The DDoS, DoS, PortScan and Brute Force attacks have been classified accurately with good prediction performance. However, the rare categories of attack like Heartbleed and Infiltration attack have shown relatively lower classification performance due to insufficient sample representation.

Feature importance assessment revealed several traffic characteristics that played an important role in intrusion detection. Traffic characteristics that included packet rate, flow duration time, packet size, and byte rate exhibited the most significant impact on attacks' classification. They provided meaningful insights in terms of determining whether traffic behavior is malicious or benign.

Comparative assessment of different models indicated that intrusion detection systems based on artificial intelligence principles proved to be superior to those based on rules in regard to their performance. Machine learning models were capable of detecting complex attacks that were not easily detected with threshold-based approaches.

In summary, the outcomes confirmed the efficiency and effectiveness of machine learning and AI-powered intrusion detection systems.

CHAPTER-5

CONCLUSION

This work was aimed at the use of machine learning and artificial intelligence approaches to detect intrusions automatically by analyzing network traffic using CICIDS2017 data set and applying the methodology similar to that used in Azure AutoML approach. The main aim of this research was to test whether machine-learning intrusion detection systems would be able to detect cyber attacks in large network traffic effectively reducing reliance on rule-based security monitoring practices.

The results of this study show that machine-learning algorithms are capable of detecting intrusions with high accuracy, reliability, and preciseness. Such machine-learning algorithms as Logistic Regression, Random Forest, XGBoost, and LightGBM were successfully used to classify between normal and malicious traffic based on statistically derived network flows from the CICIDS2017 data set. Among different models, those employing ensemble learning technique, i.e., Random Forest, XGBoost, and LightGBM proved to outperform other algorithms as they allowed handling nonlinear relations and patterns within cyber-security data.

It should be mentioned that the conducted study stressed the relevance of data preprocessing and balancing when enhancing intrusions detection effectiveness. Thus, duplicate deletion, invalid values handling, feature scaling, and application of the RandomOverSampler tool helped increase the quality of input data, thus enabling the models to better detect minority classes of intrusions. It has been proven once again that imbalance handling is a prerequisite for obtaining reliable classification results for cybersecurity-related datasets.

Finally, comparing the results obtained by applying machine learning methods with those attained during implementation of the rule-based model proved superiority of AI-powered solutions. As compared to rule-based models which are capable of detecting only predefined attack patterns, machine learning models demonstrated the ability to detect unknown intrusion types. This fact confirms that machine learning-based IDS is more flexible and scalable than traditional systems.

It has been also proven that machine learning techniques can effectively classify a dataset consisting of five categories of cyber intrusions: DDoS, DoS, PortScan, Brute Force, and Botnet attacks. Despite the difficulty in identifying several infrequent attack types due to insufficient training examples, the obtained results have demonstrated high performance of the implemented solution.

The analysis showed that the parameters which describe network traffic flows are important for detecting malicious activities. Thus, the significance of the statistical flow-based analysis was shown for detecting intrusions in computer networks. In general, the research proved that AI and machine learning are effective approaches that could be used to build efficient and scalable IDS systems.

In summary, the investigation confirms that machine learning-based and AI intrusion detection systems are an effective and efficient solution for modern computer network security. The introduction of AI technology into modern cybersecurity solutions is expected to increase efficiency in threat detection by reducing the number of false positives and negatives in real time. At the same time, the use of such approaches can help to decrease the workload of security specialists in cybersecurity companies.

CHAPTER-6

DISCUSSION

Due to the emergence of new technologies, such as digital technologies, cloud computing, IoT, and AI applications, there have been tremendous changes in the amount of complex traffic created by companies. The current cybersecurity systems that mostly depend on manual monitoring and signature-based detection become less efficient in dealing with cyber attacks. Thus, in order to find out whether the machine learning algorithms and AI could solve the issue of intrusion detection, this paper has investigated the applicability of these technologies in network security.

According to the results of the analysis, machine learning technology is able to detect intrusions by analyzing network traffic at scale and finding malicious patterns with a high degree of precision. When comparing machine learning and rules-based system, it can be claimed that the former one proved its effectiveness in detecting both known and unknown attacks. Consequently, machine learning technology offers a much better solution when dealing with cybersecurity.

Among the notable observations from the study is the superior performance of ensemble learning models such as Random Forest, XGBoost, and LightGBM. This is attributed to the fact that they can learn complex relationships between traffic features. As such, they were able to perform better in classification compared to Logistic Regression since they could capture nonlinearity among traffic features. In addition, the combination of weak learners by ensemble learning models made them more effective at intrusion detection. It should be noted that LightGBM had an advantage in terms of efficiency when working with large data sets.

Another important observation from the study concerns the preprocessing of traffic data and feature engineering in intruder detection systems. CICIDS2017 was observed to have duplicate records, missing values, infinitely valued fields, and an extremely skewed class distribution which made training difficult. However, removing these problems led to enhanced performance of the models. In addition, it was found that statistical characteristics

such as the rate of byte transfers, the packet transfer rate, and the duration of flows are crucial factors in detecting attacks based on traffic features analysis.

Furthermore, another important issue in discussing is the effect of class imbalance on machine learning algorithms. Given the presence of underrepresented attack classes in the CICIDS2017 dataset, the tendency of machine learning algorithms to predict majority classes appeared. It should be noted that using the RandomOverSampler algorithm helped to leverage the information of minority classes and reduce misclassified attacks.

Nevertheless, using an oversampling technique may be risky, as it may cause overfitting. Thus, choosing proper approaches to achieve balance is still critical.

The comparison between machine learning models and the rule-based baseline approach demonstrated significant drawbacks of conventional detection techniques. Being based on threshold values and known attack signatures, rule-based solutions have a number of limitations. For instance, they cannot detect new attack types since no corresponding signatures exist yet. On the other hand, machine learning techniques help to identify patterns of network activity. As a consequence, machine learning algorithms can cope much better with evolving threats and zero-day attacks.

Even with these positive outcomes, there are still some limitations that are present. In this research, experiments were performed using the CICIDS2017 dataset in a controlled lab environment. Practical network settings can be less predictable compared to the simulation environment, leading to potential performance deterioration when applying ML solutions built using historical data due to the appearance of new types of attacks in the wild. Regular model updates are needed in order to provide reliable threat detection.

The next limitation is connected with the explanation of advanced machine learning models. Ensemble methods and boosting can be seen as black box solutions where it is hard to understand how particular predictions were made. This issue is crucial in enterprise cybersecurity since analysts have to prove security-related decisions by explaining certain alerts and events. That is why explainability is required in practice.

In addition, the study underlines the practical relevance associated with the incorporation of AI-based intrusion detection systems into the cloud-based platforms like Microsoft Azure. The use of automated machine learning processes and highly scalable cloud infrastructure

will allow processing large amounts of data concerning network traffic and deploying intrusion detection systems in the real time environment. Moreover, such implementation may considerably facilitate work of security experts while working in such conditions. Conclusively, the analysis clearly demonstrates that machine learning and AI-based intrusion detection systems present a considerable breakthrough in comparison with other cybersecurity solutions. Although a number of research issues related to the problem include such questions as class imbalance, model interpretability and applicability to the real-life environment, AI-based intrusion detection system proves its great potential as an advanced solution.

CHAPTER-7

FUTURE SCOPE

Increasing sophistication of cyberattacks together with the constantly growing amount of data processed in terms of network traffic provides an excellent ground for further progress in terms of AI-based intrusion detection systems. Even though the current research showed how machine learning techniques are able to identify malicious activities within network traffic, there are still certain areas of improvement.

An interesting direction to consider for future work will be the use of deep learning algorithms for intrusion detection purposes. Such algorithms include CNNs, RNNs, LSTM networks, as well as transformer architecture, which may help to achieve a higher level of complexity in terms of attack detection by taking into account both temporal dependencies and more complicated patterns in traffic. Moreover, such algorithms may help to detect very sophisticated traffic patterns.

Another important point to consider in future research would be real-time intrusion detection systems that integrate data from active network traffic monitoring platforms. Currently, the study has been performed on offline CICIDS2017 dataset in a controlled environment. In practice, any intrusion detection system has to process streaming traffic constantly and react immediately in case of detecting malicious activity. Machine learning algorithms may be seamlessly integrated with real-time cloud-based systems like Microsoft Azure.

A related issue that requires future consideration is the improvement of techniques used to handle class imbalances. While RandomOverSampler has been shown to boost minority attack detection in this study, other balancing techniques such as SMOTE, ADASYN, and combination of resampling techniques can be explored for better classification performance with rare attacks. Comparative assessment of various methods can help in finding an optimal approach to dealing with class imbalance in cybersecurity data.

The concept of Explainable Artificial Intelligence (XAI) is another critical aspect that should be considered in future research. Advanced machine learning models used in practice often

tend to be black boxes. This makes it difficult to understand decisions made by algorithms used in intrusion detection systems. XAI can help improve trust in intrusion detection systems by providing explanations about the attacks detected by the system. Future research can also be done on the incorporation of intrusion detection systems with SIEM solutions and automated incident response. The combination of AI-driven threat detection together with automated response can help organizations react rapidly when cyber attacks happen. Cybersecurity technologies that work autonomously will have a significant contribution to cybersecurity management.

Federated learning and distributed AI models can be incorporated into cybersecurity practices for collaborative threat detection. Federated learning refers to the process of training machine learning models using data from different parties without the exchange of sensitive network data. The incorporation of federated learning will be beneficial to large cybersecurity networks.

Future research can also consider the application of hybrid detection systems which will incorporate signature-based detection alongside anomaly-based machine learning models. Moreover, future research can analyze the performance of the intrusion detection models based on recent and more varied cybersecurity datasets. With time, new methods of attacks emerge, and hence training of the models on more recent datasets that capture the current threat landscape would become necessary.

Lastly, optimization for efficiency is another area for future research in this field. For instance, when deploying AI-based intrusion detection systems on large corporate networks, efficient use of resources becomes a priority since the system would have to handle a huge volume of network traffic with minimal latency and computational cost. Future studies could, therefore, develop lightweight machine learning algorithms for edge and cloud computing cybersecurity applications.

In conclusion, there are numerous possibilities for the future development of AI-driven intrusion detection systems. Improvements in machine learning, cloud computing, explainable AI, and real-time cybersecurity analytics would only improve the effectiveness and efficiency of automated network security systems.

REFERENCES

- Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F., 2022. Cybersecurity threats and their mitigation approaches using machine learning – A review. *Journal of Cybersecurity and Privacy*, 2(3), 527–555. doi.org/10.3390/jcp2030027
- Alrefaei, A. & Ilyas, M., 2024. Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time. *Sensors (Basel)*, 24(14), p.4516. doi.org/10.3390/s24144516
- Avila, R., Khoury, R., Khoury, R. and Petrillo, F., 2021. Use of security logs for data leak detection: A systematic literature review. *Security and Communication Networks*, 2021, pp.1-22. doi.org/10.1155/2021/6615899
- Bartley, K., 2024. Big data statistics: How much data is there in the world? Rivery. <https://rivery.io/blog/big-data-statistics-how-much-data-is-there-in-the-world/> [Retrieved: 15 March 2025].
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D., 2001. Manifesto for Agile Software Development. <https://agilemanifesto.org/> [Retrieved: 2 June 2025].
- Belal, S. & Sundaram, S., 2022. Comprehensive review on intelligent security defenses in cloud: Taxonomy, security issues, ML/DL techniques, challenges and future trends. *Journal of King Saud University – Computer and Information Sciences*. doi.org/10.1016/j.jksuci.2022.101587
- Beigi, G., Shu, K., Zhang, Y. & Liu, H., 2018. Securing social media user data – An adversarial approach. In: *Proceedings of the 29th ACM Conference on Hypertext and Social Media (HT '18)*, Baltimore, USA, 9–12 July. New York: ACM, pp. 165–173. doi.org/10.1145/3209542.3209552
- Borgman, C.L., 2015. Big data, little data, no data: Scholarship in the networked world. Cambridge, MA: MIT Press.

Buyya, R., Ilager, S. & Arroba, P., 2023. Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data center resources and workloads. Preview at arXiv:2303.10572. <https://arxiv.org/abs/2303.10572> [Accessed: 2 June 2025].

Caruana, R., Niculescu-Mizil, A., Crew, G. & Ksikes, A., 2004. Ensemble selection from libraries of models. Paper in conference proceedings of the 21st International Conference on Machine Learning (ICML), Banff, Canada, 4–8 July 2004. <http://www.niculescu-mizil.org/papers/shotgun.icml04.revised.rev2.pdf> [Retrieved: 5 April 2025].

Chio, C. and Freeman, D., 2018. Machine learning and security: Protecting systems with data and algorithms. Beijing: O'Reilly Media.

Fakhraee, S., Balakreshnan, B. & Masanz, M., 2023. Azure Machine Learning Engineering. Birmingham: Pact Publishing. [362 pp].

Feurer, M., Klein, A., Eggenberger, K., Springenberg, J., Blum, M. and Hutter, F., 2015. Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems (NeurIPS 2015), 28, pp.2962–2970.

Grigorev, A., 2021. Machine Learning Bookcamp: Machine learning for classification. Manning Publications Co. LLC. https://learning.oreilly.com/library/view/machine-learning-bookcamp/9781617296819/OEBPS/Text/04.htm#heading_id_19 [Retrieved: 28 April 2025]

Géron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Second edition. <https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/> [Retrieved: 4 May 2025].

Goodfellow, I., Bengio, Y., & Courville, A., 2016. Deep Learning. MIT Press. Google Developers, 2024. AutoML: Benefits and limitations. <https://developers.google.com/machine-learning/crash-course/automl/benefits-limitations> [Retrieved: 10 April 2025].

Han, Y., Guo, Y., Wang, W. and Zhang, X., 2023. Anomaly Detection for Encrypted Traffic Using a Deep Neural Network. Sensors, 23(7), p.3408. doi.org/10.3390/s23073408

Joshi, I., Ghuge, T., Anjarlekar, S. and Chavan, S., 2025. Advanced Security Information and Event Management Tool. *International Research Journal of Modernization in Engineering Technology and Science*, 7(4). doi.org/10.56726/IRJMETS71666

Kallio, H., Pietilä, A.-M., Johnson, M. & Kangasniemi, M., 2016. Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, 72(12), pp.2954–2965. doi.org/10.1111/jan.13031

Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1), pp.1–22.

Li, H., Lin, L. and Zeng, H., 2024 *Machine Learning Methods*. 1st ed.

Loeser, F., Ereke, K. & Zarnekow, R., 2017. Sustainable IT services: A framework for designing sustainable IT service processes. *Journal of Cleaner Production*, 144, pp.248–257.

Mahboubi, H., Kalutarage, H., Bagheri, H., Shaghaghi, A., Faisal, A. and Jolfaei, A., 2024. Machine learning-based network intrusion detection systems: A comprehensive and systematic review of trends, challenges, and future directions. *Journal of Network and Computer Applications*, 226, pp.103932. doi.org/10.1016/j.jnca.2024.103932

Meho, L.I., 2006. E-mail interviewing in qualitative research: A methodological discussion. *Journal of the American Society for Information Science and Technology*, 57(10), pp.1284–1295. doi.org/10.1002/asi.20416.

Microsoft, 2025. Managed network isolation for desktop virtual networks. https://learn.microsoft.com/sv-se/azure/machine-learning/how-to-managed-network?view=azureml-api-2&WT.mc_id=Portal

Microsoft_Azure_MLTeamAccounts&tabs=azure-cli [Retrieved: 22 April 2025]. Microsoft, 2025. Evaluate automated machine learning experiment results. <https://learn.microsoft.com/en-us/azure/machine-learning/how-to-understand-automated-ml?view=azureml-api-2> [Retrieved: 22 February 2025]. Microsoft, 2023.

Azure Machine Learning Documentation. <https://azure.microsoft.com/en-us/services/machine-learning/> [Retrieved: 14 March 2025].

Microsoft, 2025. Azure Data Lake <https://azure.microsoft.com/en-us/solutions/data-lake> [Retrieved: 21 March 2025].

Microsoft, 2024. Getting started with Microsoft Defender. <https://support.microsoft.com/en-us/topic/getting-started-with-microsoft-defender-9df0cb0f-4866-4433-9cbc-f83e5cf77693> [Retrieved: 10 April 2025].

Microsoft, 2024. Deploy and score a machine learning model by using an online endpoint. <https://learn.microsoft.com/en-us/azure/machine-learning/how-to-deploy-online-endpoints?view=azureml-api-2&tabs=cli> [Retrieved: 11 April 2025].

Microsoft, 2025. What is automated machine learning (AutoML)? <https://learn.microsoft.com/sv-se/azure/machine-learning/concept-automated-ml?view=azureml-api-2> [Retrieved: 10 April 2025].

Microsoft, 2025. What is a data lake? <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake> [Retrieved: 21 March 2025].

Microsoft Learn, 2025. Proactively hunt for threats with advanced hunting in Microsoft Defender. <https://learn.microsoft.com/en-us/defender-xdr/advanced-hunting-overview> [Retrieved: 26 March 2025].

Microsoft, 2025. Set up AutoML training for tabular data with the Azure Machine Learning CLI and Python. SDK <https://learn.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-train?view=azureml-api-2&tabs=python#primary-metric> [Retrieved: 28 April 2025].

Microsoft, 2024. 2024 Environmental Sustainability Report. Available from: <https://www.microsoft.com/en-us/sustainability> [Retrieved: 14 May 2025].

Mohale, V.Z. & Obagbuwa, I.C., 2025. A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhance transparency and interpretability in cybersecurity. *Frontiers in Artificial Intelligence*, 8. doi.org/10.3389/frai.2025.1526221

Ortiz-Garcés, I., Govea, J., Sánchez-Viteri, S. & Villegas-Ch., W., 2024. CyberEduPlatform: An educational tool to improve cybersecurity through anomaly detection with Artificial

Intelligence. PeerJ Computer Science, 10, e2041. doi.org/10.7717/peerj-cs.2041Palinkas, L.A., Horwitz, S.M., Green, C.A., Wisdom, J.P., Duan, N. & Hoagwood, K., 2015.

Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. Administration and Policy in Mental Health and Mental Health Services Research, 42(5), pp.533–544. doi.org/10.1007/s10488-013-0528-y. Raschka, S. and Mirjalili, V., 2020.

Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. 3 edition. Birmingham, UK: Pact Publishing. Scikit-learn, 2025. recall_score-scikit-learn 1.4.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html [Retrieved: 6 May 2025].

Scikit-learn, 2025. Precision recall - scikit-learn https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html [Retrieved: 6 May 2025].

Simon, H.A., 1983. Reason in Human Affairs. Stanford University Press. Shaukat, M.W., Butt, U.A., Mehmood, M., Shah, S.B.H., Amin, R., Raza, S.M., Suh, D.Y. and Piran, M.J., 2020. A review of machine learning algorithms for cloud computing security. Electronics, 9(9), p.1379. doi.org/10.3390/electronics9091379

Tang, M., Alazab, M. and Luo, Y., 2017. Big Data for Cybersecurity: Vulnerability Disclosure Trends and Dependencies. IEEE Transactions on Big Data. https://www.researchgate.net/publication/318116583_Big_Data_for_Cybersecurity_Vulnerability_Disclosure_Trends_and_Dependencies [Retrieved: 2 May 2025].

Uddin, M.F., 2019. Addressing Accuracy Paradox Using Enhanced Weighted Performance Metric in Machine Learning. 2019 Sixth HCT Information Technology Trends (ITT), IEEE 10.1109/ITT48889.2019.9075071 [Retrieved: 13 May 2025].

Van de Ven, A.H., 2007. Engaged scholarship: A guide for organizational and social research. Oxford: Oxford University Press.

Vargas, V. W. de, Aranda, J. A. S., Costa, R. dos S., Pereira, P. R. da S., & Barbosa, J. L. V., 2023. Imbalanced data preprocessing techniques for machine learning: a systematic mapping

study. Knowledge and Information Systems, pp. 31–57. doi.org/10.1007/s10115-022-01765-4

Yenugula, M., Sahoo, S.K. & Goswami, S.S., 2024. Cloud computing for sustainable development: An analysis of environmental, economic and social benefits. Journal of Future Sustainability, 4, pp.45–60

Zhao, Y., Hu, Z. and Liu, R., 2023. TBGD: Deep Learning Methods on Network Intrusion Detection Using CICIDS2017 Dataset. J. Phys.: Conf. Ser., 2670, 012025. doi.org/10.1088/1742-6596/2670/1/012025

Zamanzadeh Darban, Z., Webb, G.I., Pan, S., Aggarwal, C., & Salehi, M., 2024. Deep learning for time series anomaly detection: A survey. ACM Computing Surveys, 57(1), pp.1–42. doi.org/10.48550/arXiv.2211.05244

Zhou, F., Wang, G., Zhang, K., Liu, S. & Zhong, T., 2023. Semi-Supervised Anomaly Detection via Neural Process. IEEE Transactions on Knowledge and Data Engineering, 35(10), pp.10423–10435. doi.org/10.1109/TKDE.2023.3266755

Zuech, R., Khoshgoftaar, T.M. & Wald, R., 2015. Intrusion detection and Big Heterogeneous Data: a survey. Journal of Big Data, 2(1), pp.1–41. doi.org/10.1186/s40537-015-0013-4

Zöller, M.A. and Huber, M.F., 2021. Benchmark and survey of automated machine learning frameworks. Journal of Artificial Intelligence Research, 70, pp.409–472.