

A NOVEL NETWORK INTRUSION DETECTION AND APPLICATION
IDENTIFICATION APPROACH BASED ON MATRIX PROFILE AND
GRAMIAN ANGULAR SUMMATION FIELD

by

Selin Berk

B.S., Electrical & Electronics Engineering, Yeditepe University, 2021

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical & Electronics Engineering
Boğaziçi University

2025

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Prof. Emin Anarım and Prof. Mutlu Koca, for their expert guidance, invaluable insights and continuous support with patience throughout the course of this thesis. Their encouragement for every challenge has been invaluable for both development of this thesis and my personal academic growth.

I sincerely thank Assoc. Prof. Şerif Bahtiyar for being part of my thesis committee. I appreciate his thoughtful feedback and the time he dedicated to evaluating my thesis.

I would like to extend my thanks to my dear friends and colleagues, Çağatay Ateş, Ramazan Yılmaz, and Onur Tatar. I appreciate their friendship and the moments we shared along the way.

Finally, I would like to express my thanks to my mother, Sibel Berk, my father, Fatih Berk, and my sister Serra Berk. Their encouragement and support have always helped me to achieve my goals.

ABSTRACT

A NOVEL NETWORK INTRUSION DETECTION AND APPLICATION IDENTIFICATION APPROACH BASED ON MATRIX PROFILE AND GRAMIAN ANGULAR SUMMATION FIELD

Network intrusions and attacks constitute a serious threat to internet structure. It is necessary to recognize attacks accurately to maintain the integrity of communication networks. Also, network traffic needs to be accurately classified for security and quality of service considerations. Machine learning applications have demonstrated superior performance in classification problems in several areas including network security. Uncovering hidden patterns is considered complex and therefore open to several novel methods. Matrix Profile (MP) and Gramian Angular Summation Field (GASF) are such mathematical algorithms that may successfully be utilized in machine learning models. This thesis presents a novel network intrusion detection and application classification methodology that incorporates MP and GASF with machine learning classifiers. Success of Random Forest model may be enhanced with MP and accuracy of Convolutional Neural Network (CNN) classifier may be improved with GASF. Features are converted to image using GASF to visualize patterns in time-series. Images are then fed to CNN model. Similarities in time-series are computed with MP algorithm as input to Random Forest classifier. Assessment of the proposed model is accomplished with publicly available datasets that contain several intrusions and network traffic application types. Validation results certainly prove that presented approach is successful and applicable. Major achievement of this study is to demonstrate that mathematical algorithms such as MP and GASF may be merged with machine learning models to yield accurate network intrusion and application classification.

ÖZET

MATRİS PROFİL VE GRAMIAN AÇISAL TOPLAM ALAN TABANLI ÖZGÜN BİR AĞ SALDIRISI TESPİTİ VE UYGULAMA TANIMA YAKLAŞIMI

Ağ ihlalleri ve saldırılar, internet yapısı için ciddi bir tehdit oluşturmaktadır. İletişim ağlarının bütünlüğünü koruyabilmek için saldırıların doğru bir şekilde tanınması gereklidir. Ayrıca, güvenlik ve hizmet kalitesi açısından ağ trafiğinin doğru bir şekilde sınıflandırılması gereklidir. Makine öğrenmesi uygulamaları, ağ güvenliği de dâhil olmak üzere çeşitli alanlardaki sınıflandırma problemlerinde üstün performans sergilemiştir. Gizli kalıpların ortaya çıkarılması karmaşık olarak değerlendirilmekte ve bu nedenle çeşitli yeni yöntemlere açık bir alan olarak görülmektedir. Matris Profil (MP) ve Gramian Açisal Toplam Alanı (GASF), makine öğrenmesi modellerinde başarıyla kullanılacak matematiksel algoritmalarlardır. Bu tez, MP ve GASF yöntemlerini makine öğrenmesi sınıflandırıcılarıyla birleştiren yeni bir ağ saldırısı tespit ve uygulama sınıflandırma yöntemi sunmaktadır. Rastgele Orman modelinin başarısı MP ile artırılabilirken, Evrişimsel Sinir Ağları(CNN) sınıflandırıcısının doğruluğu GASF yöntemiyle geliştirilebilir. Zaman serilerindeki desenleri görselleştirebilmek için öznitelikler GASF yöntemi ile görüntüye dönüştürülmektedir. Daha sonra bu görüntüler CNN modeline beslenmektedir. Zaman serilerindeki benzerlikler, Rastgele Orman sınıflandırıcısına girdi olarak kullanılmak üzere MP algoritması ile hesaplanmaktadır. Önerilen modelin değerlendirilmesi, çeşitli saldırılar ve ağ trafiği uygulama türlerini içeren herkese açık veri kümeleri ile gerçekleştirilmiştir. Doğrulama sonuçları, sunulan yaklaşımın başarılı ve uygulanabilir olduğunu kesin olarak kanıtlamaktadır. Bu çalışmanın temel başarısı, MP ve GASF gibi matematiksel algoritmaların makine öğrenmesi modelleriyle birleştirilerek doğru ağ saldırısı ve uygulama sınıflandırması elde edilebileceğinin gösterilmesidir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF ACRONYMS/ABBREVIATIONS	xiv
1. INTRODUCTION	1
1.1. Intrusion Detection	6
1.1.1. Intrusion Detection Methods	7
1.2. Motivation	9
1.3. Thesis Contribution	14
1.4. Thesis Organization	16
2. NETWORK INTRUSION DETECTION	17
2.1. Network Attacks	17
2.2. Network Attacks: Detection,Effects and Prevention	19
2.2.1. Brute Force	20
2.2.1.1. Basic Brute Force Attacks	21
2.2.1.2. Reverse Brute Force Attacks	21
2.2.1.3. Dictionary Attacks	21
2.2.1.4. Hybrid Brute Force Attacks	21
2.2.2. Distributed Denial of Service	22
2.2.2.1. SYN Flood Attack	24
2.2.2.2. UDP Flood Attack	25
2.2.2.3. HTTP Flood Attack	25
2.2.2.4. DNS Amplification Attack	25
2.2.2.5. ACK-PUSH Flood Attack	26
2.2.2.6. ACK Fragmentation Attack	26
2.2.3. Denial of Service	26

2.2.3.1.	SYN Flood	27
2.2.3.2.	UDP Flood	27
2.2.3.3.	Slowloris Attack	27
2.2.4.	Reconnaissance	28
2.2.4.1.	Operating System Scanning	28
2.2.4.2.	Ping Sweep	29
2.2.4.3.	Port Scanning	29
2.2.4.4.	Vulnerability Assessment	29
2.2.4.5.	Host Discovery	30
2.2.5.	Web Based	30
2.2.5.1.	Browser Hijacking	31
2.2.5.2.	Command Injection	31
2.2.5.3.	SQL Injection	31
2.2.5.4.	Uploading Attack	31
2.2.5.5.	Cross-site Scripting	32
2.2.5.6.	Backdoor Malware	32
2.2.6.	Spoofing	32
2.2.6.1.	DNS Spoofing	33
2.2.6.2.	ARP Spoofing	34
2.3.	Related Work	34
3.	NETWORK INTRUSION DETECTION METHODOLOGIES	48
3.1.	Matrix Profile	48
3.2.	Time-series to Image Transformations	52
3.2.1.	Gramian Angular Summation Field	56
3.3.	Machine Learning Algorithms	61
3.3.1.	Decision Tree	62
3.3.2.	Random Forest	63
3.3.3.	Support Vector Machine	63
3.3.4.	k-Nearest Neighbor	64
3.3.5.	Convolutional Neural Networks	65
3.3.6.	Multi-layer Perceptron	65

3.4. Proposed Methods	66
3.4.1. Proposed Matrix Profile Based Random Forest Model	66
3.4.2. Proposed GASF Based Convolutional Neural Network Model	67
4. FEATURE EXTRACTION METHODS FOR NETWORK INTRUSION AND APPLICATION CLASSIFICATION	75
4.1. Statistical Features	75
4.1.1. Inter-arrival Time	75
4.1.2. Packet Size	75
4.2. Information Theory Based Features	76
5. PERFORMANCE EVALUATION OF THE PROPOSED METHOD	78
5.1. Data Sets	78
5.1.1. Boğaziçi University DDoS Attack Data Set	79
5.1.2. CIC IoT 2023 Data Set	79
5.1.3. Encrypted Network Traffic Classification Data Set	81
5.2. Performance Metrics	82
5.3. Matrix Profile Based Random Forest Model Results	83
5.3.1. Intrusion Detection	85
5.3.2. Application Identification	92
5.4. Gramian Angular Summation Field Based CNN Model Results	98
5.4.1. Intrusion Detection	99
5.4.2. Application Identification	104
6. CONCLUSION & FUTURE WORK	109
REFERENCES	112

LIST OF FIGURES

Figure 3.1.	Calculation of Euclidean distance between two points.	50
Figure 3.2.	Example of a time-series sequence.	50
Figure 3.3.	Calculation of Euclidean distance and distance array.	51
Figure 3.4.	Calculation of distance matrix and matrix profile.	51
Figure 3.5.	Time-series containing matrix profile values.	52
Figure 3.6.	Example of a sinusoidal function and related GASF transformation.	60
Figure 3.7.	Example of a rectangular function and related GASF transformation.	61
Figure 3.8.	Example of a sawtooth function and related GASF transformation.	61
Figure 3.9.	Block diagram of matrix profile based machine learning model.	67
Figure 3.10.	Convolutional layers of proposed CNN model.	71
Figure 3.11.	Pooling layers of proposed CNN model.	73
Figure 3.12.	Block diagram of proposed CNN Model.	74
Figure 5.1.	Matrix profile results of mean entropy values with different sliding window sizes.	84

Figure 5.2.	ROC Curve on mean entropy values of Boğaziçi University DDoS attack data set.	84
Figure 5.3.	Confusion matrix of attack classes using matrix profile based random forest method with CIC IoT 2023 data set.	86
Figure 5.4.	Effect of matrix profile method on F1-score for attack class. . . .	90
Figure 5.5.	Effect of matrix profile method on F1-score for attack type. . . .	91
Figure 5.6.	Performance comparison of machine learning algorithms with matrix profile method.	92
Figure 5.7.	Confusion matrix of application categories using matrix profile based random forest method with Encrypted Network Traffic Classification data set.	93
Figure 5.8.	Effect of matrix profile method on F1-score for application category.	98
Figure 5.9.	Effect of matrix profile method on F1-score for attack type. . . .	98
Figure 5.10.	Confusion matrix of attack classes using GASF based CNN model with CIC IoT 2023 data set.	100
Figure 5.11.	Confusion matrix of application categories using GASF based CNN model with Encrypted Network Traffic Classification data set. . .	105

LIST OF TABLES

Table 3.1.	Activation Functions.	72
Table 4.1.	Information theory-based Features.	77
Table 5.1.	Attack Classes with their Subclasses of CIC IoT 2023 Data Set. . .	80
Table 5.2.	Category and Applications of Encrypted Network Traffic Classification Data Set.	81
Table 5.3.	Confusion matrix for 2x2 binary classification case.	82
Table 5.4.	Performance Metrics of Matrix Profile Method using Boğaziçi University DDoS Attack Data Set.	85
Table 5.5.	Performance Table of Attack Classes using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	87
Table 5.6.	Performance Table of DDoS Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	88
Table 5.7.	Performance Table of DoS Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	88
Table 5.8.	Performance Table of Reconnaissance Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	89
Table 5.9.	Performance Table of Spoofing Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	89

Table 5.10.	Performance Table of Web-based Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.	90
Table 5.11.	Performance Table of Application Categories using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.	94
Table 5.12.	Performance Table of Audio Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.	94
Table 5.13.	Performance Table of File Transfer using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.	95
Table 5.14.	Performance Table of Text Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.	96
Table 5.15.	Performance Table of Video Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.	97
Table 5.16.	Performance Table of Attack Classes using GASF based CNN model with CIC IoT 2023 Data Set.	100
Table 5.17.	Performance Table of DDoS Class using GASF based CNN model with CIC IoT 2023 Data Set.	101
Table 5.18.	Performance Table of DoS Class using GASF based CNN model with CIC IoT 2023 Data Set.	102

Table 5.19.	Performance Table of Reconnaissance Class using GASF based CNN model with CIC IoT 2023 Data Set.	102
Table 5.20.	Performance Table of Spoofing Class using GASF based CNN model with CIC IoT 2023 Data Set.	103
Table 5.21.	Performance Table of Web-based Class using GASF based CNN model with CIC IoT 2023 Data Set.	104
Table 5.22.	Performance Table of Application Categories using GASF based CNN model with Encrypted Network Traffic Classification Data Set.	104
Table 5.23.	Performance Table of Audio Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.	106
Table 5.24.	Performance Table of File Transfer using GASF based CNN model with Encrypted Network Traffic Classification Data Set.	106
Table 5.25.	Performance Table of Text Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.	107
Table 5.26.	Performance Table of Video Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.	108

LIST OF ACRONYMS/ABBREVIATIONS

ACK	Acknowledgment
ANN	Artificial Neural Network
ARP	Address Resolution Protocol
AUC	Area Under Curve
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DL	Deep Learning
DNS	Domain Name System
DoS	Denial of Service
GASF	Gramian Angular Summation Field
HTTP	Hyper-Text Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Prevention System
IP	Internet Protocol
KKN	k-Nearest Neighbor
KL	Kullback-Leibler
LSTM	Long Short-Term Memory
MAC	Media Access Control
ML	Machine Learning
MP	Matrix Profile
MTF	Markov Transition Field
OS	Operating System
PSH	Push
QoS	Quality of Service

ReLU	Rectified Linear Unit
RF	Random Forest
ROC	Receiver Operating Characteristic
RP	Recurrence Plot
SQL	Structured Query Language
SVM	Support Vector Machine
SYN	Synchronize
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
XSS	Cross-site Scripting

1. INTRODUCTION

Over the past decade, substantial progress has been achieved in the field of computer networking. However, alongside these advancements, network security threats have also escalated significantly. As the use of networked systems continues to grow, malicious activities pose increasing risks to the stability and reliability of Internet-based applications. Network attacks have become a prevalent form of cyber threat and are now recognized as one of the most critical challenges in large-scale computing environments. These attacks have increased both in frequency and intensity, often exhibiting short durations but high packet transmission rates. Such intrusions can severely impact the performance of packet-switched networks, leading to degraded service availability and user experience. Although Intrusion Detection Systems (IDS) offer mechanisms to identify and mitigate various forms of attacks, they may be ineffective when adversaries craft packets that closely resemble legitimate traffic. This necessitates the development of more robust anomaly detection techniques. Among the most disruptive attack vectors are flooding-based assaults, wherein attackers transmit a large volume of packets to exhaust network resources and bandwidth. This leads to network congestion, service interruptions, and, in extreme cases, total denial of service. DoS (Denial of Service) attacks, in particular, exemplify such aggressive tactics and are capable of consuming significant computational and communication resources. Traditional IDS approaches largely depend on statistical analysis of network traffic. However, as the behavioral patterns of attacks increasingly mirror those of normal traffic, these methods face growing limitations. Moreover, attackers may dynamically adjust parameters such as intensity, pulse duration, and inter-attack intervals, thereby complicating detection and prevention strategies. The overarching goal of such attacks is to destabilize the target network and diminish the quality of service delivered to end users [1].

Numerous studies have examined various types of network attacks, often classifying them based on their impact on system resources such as processing power, mem-

ory, and bandwidth. These resource-depletion attacks—including flooding—aim to exhaust critical network capabilities and disrupt service. While extensive research has been conducted to detect such threats, they remain a significant risk, particularly to large-scale organizations. Some attacks, such as DoS, generate substantial traffic to overwhelm the target. However, other attacks may achieve similar disruptive effects using relatively low traffic volumes by strategically depleting key resources. This underscores the necessity of deploying real-time IDS on critical networks to effectively identify and mitigate these threats. Several approaches have been proposed to minimize the impact of such attacks and preserve service continuity. Performance metrics are essential to guide the selection of appropriate detection methods. Protecting networks from resource exhaustion remains a top priority, as successful intrusions can cause service disruptions, rapid traffic degradation, and congestion. To maintain service quality, networks must incorporate effective congestion mitigation strategies that allow performance recovery post-attack. Attackers often use spoofed source Internet Protocol(IP) addresses, random destination ports, and custom-crafted packets to evade detection [2]. The result is a surge in traffic directed at a specific target, characterized by multiple source addresses, a common destination, and a high volume of request packets. Detection relies on identifying distinguishable traffic patterns. Packet header features—extracted at defined intervals using tools such as Wireshark—serve as critical indicators in identifying anomalous behavior and facilitating timely response.

Reliable networking reports confirm that large-scale network attacks are increasingly common and difficult to detect in real-time, which poses a serious threat to the stable operation of Internet applications and services. These intrusions have become a critical issue in communication systems, significantly affecting service quality. As a result, the development of intelligent, automated detection frameworks has gained importance in order to eliminate manual errors and enhance system reliability. Accurate and timely identification of intrusions is essential for safeguarding network stability. This study introduces a comprehensive time-series transformation approach for analyzing Internet packet traffic to detect anomalies. Recent research is reviewed to explore current methods, application areas, and the challenges associated with trans-

forming time-series data into images. It is evident that novel techniques are needed to effectively recognize and classify diverse types of network attacks. The proposed method analyzes incoming traffic properties by extracting key features and evaluating parameters. In traditional statistical approaches, attacks are detected when parameters exceed predefined thresholds. In contrast, this thesis presents an innovative representation method that leverages machine learning and convolutional neural networks to improve intrusion and application recognition. The model's parameters are adaptable, and validation using public datasets demonstrated high performance. To address the limitations of traditional systems, this study explores time-series image transformation techniques such as Matrix Profile (MP) and Gramian Angular Summation Field (GASF). These methods enhance the performance of machine learning and Convolutional Neural Network (CNN) classifiers by improving pattern recognition and feature extraction. While conventional time-series analysis methods have shown acceptable results, they may fall short when applied to complex network traffic data. The unique characteristics of packet-level time-series data demand more sophisticated analysis methods, as addressed in this study.

Image encoding techniques are widely used in network applications for intelligent and efficient fault diagnosis. This thesis proposes a framework in which such techniques are adapted for security-focused applications, particularly for detecting malicious network attacks and classifying network traffic. Recent studies highlight the effectiveness of matrix profile and GASF algorithms in classification tasks. Motivated by these findings, this work employs matrix profile and GASF as core components of the proposed detection strategy. These time-series-to-image transformation methods produce distinctive visual representations of traffic patterns that reveal attack conditions, eliminating the need for manual detection [3]. Following model training on benchmark datasets, the neural network architecture demonstrated strong performance in identifying various attack types. The proposed multi-stage framework combines matrix profile and GASF transformations with machine learning algorithms for accurate classification. By converting time-series data into images, the model captures subtle patterns and features not evident in raw data. This allows the network intrusion detection task

to be reformulated as an image classification problem. A robust multi-label CNN is developed to detect and differentiate attack types based on these image representations. Additionally, the matrix profile algorithm enhances the performance of a random forest classifier, further validating the versatility of the proposed approach. Compared to traditional binary classification models, this framework offers a more comprehensive and scalable solution for network security applications.

Time-series-to-image conversion is widely used in signal processing and pattern recognition, particularly through the GASF algorithm. This thesis applies GASF to transform temporal network packet data into image representations, which are then used to train a CNN for intrusion detection and traffic classification. The GASF method encodes time-series data by mapping it into polar coordinates, where each element of the resulting Gramian matrix is the cosine of the sum of two normalized values. This produces visually interpretable and information-rich images [4]. While evenly spaced time-series are effective for analyzing network traffic, real-world packet transmissions often occur at irregular intervals. To address this, a fixed time window is used to convert raw packet data—captured via the industry-standard Wireshark tool—into evenly spaced sequences. Manual interpretation of GASF images is time-consuming and error-prone, especially in high-traffic environments. Therefore, automated, intelligent systems are essential for timely and accurate intrusion detection. CNN models are particularly well-suited for this task, as they can efficiently extract and classify features from the generated images without manual intervention. This thesis asserts that transforming one-dimensional time-series data into two-dimensional images enhances classification performance. The resulting representations are both distinguishable and feature-rich, enabling accurate and efficient network attack detection in real time.

Sequence data is prevalent across various domains, including network systems, and is often characterized by dynamic patterns and evolving trends—making anomaly detection particularly challenging. Effective detection requires leveraging the sequential nature of the data. In deep learning, CNNs are widely used for image-based tasks. A typical CNN architecture includes convolutional layers for feature extraction, pool-

ing layers for dimensionality reduction, and fully connected layers for classification. Despite their effectiveness, CNNs and other machine learning models such as Artificial Neural Networks (ANN) are computationally intensive and must be carefully tuned for efficient processor usage. An alternative approach for analyzing time-series data is the matrix profile technique, which identifies anomalies by calculating distances between subsequences within the series. This method excels in detecting patterns and outliers with minimal configuration and low computational cost, making it well-suited for resource-constrained environments. Moreover, integrating matrix profile features with machine learning models like Random Forest (RF) can further enhance classification accuracy.

Numerous machine learning approaches, including neural networks, have been proposed for time-series classification. Converting time-series data into images using techniques like GASF serves as an effective feature engineering step, particularly for CNN-based models. Supervised learning is widely used to train classifiers on labeled datasets, and model performance is subsequently validated through standard evaluation metrics [5]. In this study, the proposed method—combining GASF transformation with a CNN classifier—demonstrates high accuracy in detecting network intrusions and classifying application traffic. Performance is assessed using benchmark datasets commonly used in academic research, which include comprehensive traces of various network activities. The confusion matrix is also presented to provide a detailed interpretation of the model’s predictive accuracy.

Among deep learning models, CNNs have gained significant popularity for their ability to extract meaningful features from input images and outperform traditional algorithms in classification tasks. This thesis leverages CNNs in conjunction with time-series-to-image transformation, specifically using the GASF, as a powerful feature engineering technique. Rooted in computer vision, this approach enables the extraction of patterns from two-dimensional representations of time-series data that are not easily observable in one-dimensional form, while preserving temporal information. The proposed framework processes network traffic data captured from real environ-

ments and transforms it into image matrices using GASF. These are then fed into CNN and Random Forest models for intrusion and application classification. Experimental results demonstrate that this approach achieves high detection accuracy and shows strong potential for real-time implementation. CNNs are particularly well-suited for grid-like data structures such as images. Their architecture includes convolutional layers that apply filters to extract features, pooling layers that reduce dimensionality, and fully connected layers that produce classification outputs. These components work together to automatically extract deep features essential for recognizing attacks and classifying network applications. The analysis in this study shows that different types and intensities of attacks influence traffic features in distinct ways. Feature vectors derived from packet-level traffic reflect variations depending on attack scale, network size, and attacker behavior. The attack detection process in this thesis consists of three main stages: data acquisition, feature extraction, and classification. While CNNs and other machine learning models like Random Forests offer strong performance, they can be computationally intensive. Efficient use of system resources is therefore important for practical deployment. Accurate application classification, covering services such as browsing, messaging, streaming, or file transfer, is critical for network management and quality of service optimization. The combined use of CNNs and advanced mathematical techniques like GASF and matrix profile proves effective for both intrusion detection and traffic classification [6].

1.1. Intrusion Detection

Anomaly detection in time-series data has been widely studied and is typically categorized into three main approaches: distance-based, clustering-based, and prediction-based methods. Distance-based techniques rely on calculating the distance between data points using a predefined metric; objects significantly distant from others are flagged as anomalies. For instance, k-Nearest Neighbor (KNN)-based methods assign anomaly scores based on the average distances to the k nearest points. However, these methods face two key limitations: their performance heavily depends on the selected distance metric, and their computational complexity can reach $\mathcal{O}(n^2)$, making

them inefficient for large datasets. Clustering-based techniques, on the other hand, group data into clusters and treat outliers—data points that do not fit well into any cluster—as anomalies [7]. While simple and supported by extensive prior research, these methods often conflate the goals of clustering and anomaly detection. Clustering aims to categorize data, whereas anomaly detection focuses on identifying deviations. As such, clustering-based approaches may yield suboptimal results, especially when the definition of "anomaly" is vague or context-dependent. Due to these limitations, traditional techniques may fail to achieve high detection accuracy. This highlights the need for more specialized approaches—such as the GASF-CNN framework proposed in this thesis—which are explicitly designed for time-series anomaly detection.

Anomaly detection for network intrusion is highly dependent on the effective use of distance measures. Extensive recent research offers in-depth theoretical analysis of these measures and explores a wide range of techniques, including k-means clustering. Additionally, statistical methods are commonly used, where an attack is identified when computed metrics within a given time window exceed predefined threshold values.

1.1.1. Intrusion Detection Methods

Network intrusions aim to disrupt or deny services by overwhelming target resources. While many attacks generate high traffic volumes, detecting those that operate with minimal traffic—yet still impact service availability—remains a major challenge in large-scale networks. Intrusion Detection Systems (IDS) are designed to identify such threats targeting servers and network devices. IDS approaches are generally categorized into signature-based and anomaly-based methods. Signature-based IDS detect attacks by matching incoming traffic patterns to a database of known attack signatures. While effective for known threats, they struggle to detect novel or previously unseen attacks. In contrast, anomaly-based IDS model normal network behavior and flag deviations as potential intrusions [8]. These systems often employ statistical techniques such as clustering and multivariate analysis, using extracted features to detect irregular traffic patterns.

Modern IDS commonly relies on packet analysis to identify the onset of malicious network activity. Protocol information within packet headers is examined to detect anomalies. In addition to conventional methods, wavelet and spectral analysis techniques have been proposed. These approaches treat metrics like packet arrival rates as signals and apply mathematical transformations to classify traffic in the frequency domain as either normal or malicious. Spectrum-based methods provide an alternative perspective for detecting irregular traffic patterns. Compared to edge-based protection systems, network-based IDS offers improved visibility and better support for infrastructure protection and quality of service management. While raw traffic flows have been used for attack detection, distinguishing subtle attack types that do not exhibit abnormal volume often requires deeper packet inspection, placing additional strain on resource, limited systems. To address this, researchers increasingly apply mathematical techniques to extract richer features for accurate traffic classification and threat detection, enhancing the performance and reliability of IDS.

Deep learning-based models have shown strong potential for analyzing network traffic and detecting intrusions. For example, CNNs are effective in extracting packet-level features, while other models—such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Naive Bayes, Decision Trees, and k-means clustering—have also been applied successfully. Studies indicate that these methods vary in accuracy, largely depending on the quality of the training dataset. k-Nearest Neighbors (KNN), for instance, can be effective but suffers from high computational and memory requirements, making it less suitable for large-scale datasets [9].

Detection and mitigation of network attacks are critical topics that require careful consideration. Common defense strategies include filtering and blocking based on attributes such as source/destination addresses, packet counts, and lengths. When attack characteristics are clearly defined, detection algorithms can identify them more easily. Many studies aim to build platforms capable of reliably detecting and responding to such threats. Network attacks often exploit system limitations—such as memory or bandwidth—by generating high volumes of malicious packets. These intrusions can

severely impact security and service stability. Most detection methods rely on anomaly detection, which compares current traffic patterns to known normal states. Deviations are flagged as potential threats. Some approaches further incorporate traffic filtering to block malicious data. Effective detection and response must be timely to prevent service disruptions. Current techniques typically use statistical thresholds or machine learning models to differentiate between normal and malicious behavior, ensuring network performance is maintained.

Traffic attributes are computed from extracted features, often using a sliding window approach to calculate statistical metrics such as mean and variance. If attribute values exceed predefined thresholds, the traffic is flagged as anomalous. Many studies highlight traffic variation as one of the most distinguishing features for intrusion detection, making feature extraction a crucial step [10]. One challenge is identifying the optimal feature set to represent diverse attack types, as multiple attributes may change simultaneously during an attack. In the feature generation phase, traffic data—collected from raw packet captures—is segmented into equally spaced time windows. Attributes such as source/destination IPs, ports, flags, and other packet metadata are then used to construct time-series representations. Due to the wide range of available features, dimensionality can be high. Attack onset often correlates with sharp increases in traffic and abnormal variations in port activity. When attribute values exceed thresholds, the system classifies the traffic as malicious. Effective detection methods must be robust against false negatives, especially for complex or low-volume attacks. Furthermore, the inherent variability of Internet traffic, including undefined peak periods, must be considered to develop a reliable and adaptive detection methodology.

1.2. Motivation

With the rapid expansion of the Internet, networking studies now face numerous challenges—particularly regarding infrastructure and data security. Network intrusions are often analyzed through the lens of information theory, and several mathematical models have been proposed to classify and understand their complex behavior. Devel-

oping effective detection methods remains a demanding task, and this study contributes to ongoing efforts by proposing usable models and algorithms. Security concerns in communication systems are significant, as anomalies like intrusions can impact not only network performance but also broader domains such as business operations. As a result, anomaly detection has become a major research area, leading to the development of various algorithms and techniques aimed at improving network security and quality of service. The growing complexity of network environments makes effective management challenging. Application classification can aid in mitigating vulnerabilities and enhancing overall security. Modern network devices, often embedded with advanced software and connected via local or global networks, contribute to the continuous generation of large volumes of time-series data. Consequently, time-series analysis has become essential across numerous domains, including network intrusion detection.

Networking enhances device scalability, efficiency, and connectivity—but also introduces new security threats. As such, systematic approaches are essential to achieve and maintain robust network security. This thesis contributes to the growing field of intrusion detection by analyzing network attacks and aiding the development of prevention strategies. The study focuses on critically examining network intrusion and application traffic characteristics, identifying key parameters relevant to security challenges. Given the complexity of network packet time-series data and the need for real-time processing in many applications, advanced tools and platforms are required. To address these challenges, this thesis explores image transformation techniques such as Matrix Profile and GASF, which convert time-series data into visual representations. These methods not only facilitate analysis but also enable dimensionality reduction—preserving essential information while improving efficiency in processing and storage.

Traffic anomalies, particularly network intrusions, are common in modern communication systems. Ensuring server availability is critical for maintaining the Quality of Service (QoS), especially in large-scale environments such as corporate networks. Intrusions can significantly disrupt infrastructure, degrade service, and compromise

commercial transactions by flooding networks with malicious or meaningless packets. These attacks exploit protocol vulnerabilities, consume valuable resources, and lead to service downtime—making network security a critical area of research. Anomaly detection involves comparing normal and abnormal traffic states to identify harmful activity, typically through feature extraction. This study emphasizes such approaches, addressing the high frequency and impact of intrusions in today’s networks. Traditional detection methods rely on packet analysis, but newer techniques—such as time-series modeling and visualization—offer improved accuracy in recognizing abnormal behavior. Time-series representation, paired with mathematical models like Matrix Profile and GASF, enables better analysis and classification of traffic anomalies. Visualizing traffic features supports more intuitive understanding of packet behaviors and facilitates detection decisions. However, challenges such as computational complexity must be considered for real-world deployment. Indicators like abnormal Central Processing Unit (CPU) usage, excessive task counts, and high memory consumption can signal ongoing attacks. During such events, resources are often consumed by illegitimate requests, leading to service degradation or DoS—a frequently observed outcome in affected networks. This thesis explores the effectiveness of time-series analysis and visualization techniques as viable and scalable solutions for advanced intrusion detection.

Malicious attackers can initiate activities that congest network resources, degrade performance, and cause instability. While recovery is possible, it is often slow and resource-intensive. To mitigate these threats, effective defense platforms are essential. This thesis contributes to ongoing research efforts aimed at designing scalable, real-time intrusion detection and prevention systems. Existing defense mechanisms—such as firewalls, filtering, port blocking, and Intrusion Prevention Systems (IPS)—are widely deployed, but their scalability and real-time effectiveness remain limited. The diversity of network attack types, each with unique characteristics, presents a significant research challenge. A key objective of this study is to propose a system that detects and classifies attacks quickly, minimizing their impact on QoS. Intrusion detection can be approached through both real-time systems, which analyze traffic on the fly, and offline systems, which process captured data post-transmission. Real-time sys-

tems are preferred for ensuring uninterrupted service delivery. Among recent research trends, time-series analysis has proven effective in recognizing anomalies. Techniques such as Matrix Profile and GASF have shown strong potential in transforming temporal network traffic into informative visual representations suitable for machine learning (ML) and deep learning (DL) models. This thesis builds on these findings, proposing a time-series-to-image conversion framework using Matrix Profile and GASF for intrusion and application traffic classification. Carefully selected features from network packets—such as source/destination IPs, ports, and packet timing—are used to construct time-series data, which are then visualized and processed by CNNs. Validation is performed using publicly available datasets that contain a variety of intrusion and application traffic types. The results demonstrate high classification accuracy, low false positive rates, and minimal detection delays which are key indicators of an effective system. The proposed method emphasizes computational efficiency, scalability, and robustness, making it well-suited for deployment in large-scale networks. By leveraging time-series image transformations and ML models, this work offers a practical and accurate solution for detecting and classifying network intrusions. The approach provides a foundation for future real-time IDS and supports continued development of intelligent, adaptive security frameworks.

This thesis aims to assist researchers in selecting effective methods for network intrusion and application traffic classification by evaluating known mathematical techniques applied to widely used traffic features. The integration of Matrix Profile and GASF time-series-to-image transformations with Random Forest and CNN models is explored, demonstrating strong potential for accurately detecting and classifying both malicious activity and application traffic. The proposed approach highlights the applicability of time-series visualization in real-time detection platforms. While neural networks lack a predefined structure, model architecture is optimized through iterative experimentation to identify the most effective configuration. The method focuses on analyzing ordinary traffic patterns and extracting discriminative features from anomalous traffic to inform preventive actions. ML techniques are leveraged to learn and differentiate normal versus intrusive patterns, while addressing the limitations of purely

statistical methods—such as the challenge of defining accurate thresholds. The framework presented in this study builds on time-series analysis, introducing mathematical enhancements to improve classification accuracy and detection speed. Importantly, real-time recognition and response are critical to maintaining secure and stable communication infrastructures. Failure to detect and respond to attacks in time can lead to service disruption or system failure. This work contributes to advancing the field by proposing a scalable, accurate, and practical classification framework, and lays the groundwork for developing future intrusion prevention and mitigation systems. While prior studies have introduced various methods, many still require improvement in terms of scalability, robustness, and real-world applicability. This thesis aims to enhance existing approaches with novel model designs and contribute to the development of more effective and adaptive classifiers.

Intrusion detection models are typically evaluated using standard performance metrics. Numerous studies have demonstrated that applying ML techniques can effectively address network intrusion and application classification challenges. The promising results of ML-based approaches provide strong motivation for this work. A key consideration in implementing prevention systems is ensuring minimal impact on existing infrastructure, particularly in terms of resource usage such as processing power. This thesis carefully accounts for these factors in selecting a complete and practical model. The primary contribution of this work is to demonstrate that time-series processing algorithms—specifically Matrix Profile and GASF—can be successfully applied to detect intrusions and classify network applications. Given the severe consequences of such intrusions, including service disruption and server overload, the need for intelligent, scalable detection solutions is clear. As intrusion mechanisms and application behaviors grow increasingly complex, the importance of effective network management and security frameworks becomes more critical. This thesis addresses these challenges and contributes novel, ML-driven solutions for enhanced classification and real-time threat mitigation.

Therefore, it can be stated that networks remain vulnerable to various cyber-attacks due to inadequate security measures or unresolved networking challenges. These attacks are complex and have been the focus of extensive research aimed at developing effective prevention platforms. This study contributes to that ongoing effort.

A key motivation for improving a model is to enhance its accuracy, often by modifying or combining existing models. However, it is equally valuable to explore accuracy improvements without fundamentally altering the model structure. Recent studies suggest that time-series transformation, particularly into image form, can significantly boost performance. This insight forms the core motivation of this thesis. Notable advantages of representing time-series data as images include:

- It is significantly easier to visualize and analyze complex patterns or trends.
- Visual representation of temporal data is formed thus, enabling for intuitive evaluation and pattern recognition.
- Transforming time-series data properly keeps temporal dependencies as an important point which may allow for wider analysis and better discernment.
- Deep learning techniques are known to be suited better for images thus, time-series traffic data patterns may be classified accurately.

Therefore, image transformation may easily be expressed as a significant innovation with potential to enhance outcomes for networking applications. It is widely known that image conversion techniques are examined for neural network studies. This thesis also aims to contribute to such research.

1.3. Thesis Contribution

The contributions of this thesis can be stated as follows:

- Effective Time-Series to Image Transformation Using GASF: This thesis demonstrates that converting time-series data into two-dimensional image representa-

tions using the GASF algorithm significantly enhances pattern visibility. Such visual patterns, which are often difficult to identify in raw one-dimensional time-series data, become more discernible, thus improving the interpretability and classification potential of the data.

- **Integration of Machine Learning and Deep Learning in Network Security:** A novel framework is proposed that integrates CNNs and RF classifiers for detecting network intrusions and classifying application traffic. The image-based representation is particularly well-suited for CNNs, while the structured features extracted using the Matrix Profile algorithm serve as input to Random Forest. The methodology is flexible and extendable to other models such as kNN and SVM.
- **Enhanced Feature Engineering via Matrix Profile:** This work introduces a feature preprocessing stage using the Matrix Profile algorithm. Time-series data is segmented into fixed-size windows, and statistical relationships are computed to highlight anomalies. These processed features are shown to improve the distinctiveness and quality of the input data, making the subsequent classification more robust and efficient. Additionally, selected features are combined to generate composite (overlay) images, further improving CNN performance.
- **Improved Model Accuracy Through Visualization:** The experimental results confirm that image-based deep learning methods (especially CNNs trained on GASF-transformed data) can significantly increase classification accuracy, reduce false positives, and shorten detection delay. The use of time-series visualization as a preprocessing method proves to be a key factor in these improvements.
- **Creation of Novel Feature Sets Based on Statistical and Information Theory:** New composite features are introduced using statistical and information-theoretic methods. These features enhance the representation of traffic behavior under normal and attack conditions, contributing to better learning and classification outcomes.

Therefore, the approach presented in this thesis substantially adds to machine learning and deep learning studies for network intrusion and application classification context.

1.4. Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 provides an overview of various network attacks, including brute force, distributed denial of service, reconnaissance, web-based and spoofing attacks. This chapter also discusses detection techniques, and prevention methods, along with a review of related work in the literature. Chapter 3 presents network intrusion detection methodologies, covering matrix profile, time-series-to-image transformations and a range of machine and deep learning algorithms. It concludes with the proposed approaches developed in this study. Chapter 4 presents the feature extraction techniques utilized for network intrusion detection. Chapter 5 evaluates the performance of the proposed methods. It introduces the datasets used in the experiments, outlines the evaluation metrics, and presents results for both the matrix profile-based and GASF-based models in terms of intrusion detection and application identification. Finally, Chapter 6 concludes the thesis with a summary of findings and suggestions for future research directions.

2. NETWORK INTRUSION DETECTION

2.1. Network Attacks

Internet applications are widely used across sectors to enhance system efficiency. However, the growing reliance on these applications has increased vulnerability to service disrupting intrusions. This has made large-scale network attacks, especially DDoS, a major concern for security professionals. Despite various preventive measures, such threats continue to pose risks to communication services. The evolving network packet architecture, while expanding Internet capabilities, has also become a target for malicious actors. The rapid growth of Internet-connected devices has further amplified the threat. These attacks typically consume processing power, memory, and network available sockets. Several studies categorize network attacks according to network layers such as application layer attacks and infrastructure layer attacks. Some major attacks exploit vulnerabilities in transport and network layers of widely known general model. Examples of such attacks are different type packet floods. There are also divisions that classify network intrusions and attacks based on volume of malicious requests that are sent. Thus, high volume network attacks and low volume network attacks are noted. It is well-known that, network attacks are cyber-attacks that target the availability and service structure of servers. As a result, normal users are not allowed have access to service delivered. Therefore, it is very important to develop defense mechanisms against such attacks. However, the interconnected computer systems and network devices are significantly vulnerable to a very large number of attacks such as denial of service attack as major one [11].

Network attacks are malicious attempts to disrupt services by exhausting a target system's resources such as CPU, memory, or bandwidth. These attacks significantly change the statistical distributions of network traffic, making them a major concern for organizations like Internet service providers. Since such attacks occur frequently and are difficult to block in real-time, efficient, network-based detection methods are

essential. For example, a denial-of-service attack can disable servers, leading to hours of service disruption and degraded quality. Typically, these attacks involve an attacker depleting resources to impair normal operations. Effective defense requires accurate traffic analysis, anomaly detection, and reliable profiling of service behavior.

Distinguishing between normal and malicious network traffic is often challenging, as attack packets can closely resemble legitimate ones, creating uncertainty in detection. Various mathematical methods have been proposed to address this problem, aiming to improve identification accuracy and optimize the use of network resources. By design, such attacks disrupt services by overwhelming servers with simultaneous requests from multiple sources, reducing access for legitimate users. Distributed Denial-of-Service (DDoS) attacks are particularly difficult to detect and prevent due to their use of numerous devices and locations. As the most prevalent form of network disruption, DDoS attacks often exploit normal-looking requests to consume extensive computing and bandwidth resources, blocking access to authorized users. These attacks can target different layers of the network layer, including the network, transport, and application layers [12].

Network attacks, a growing threat to the Internet, are increasing in size, frequency, and complexity. DDoS attacks often render resources inaccessible and may even cause system failures by overwhelming targets with protocol requests, leading to severe performance degradation for legitimate users. This highlights the urgent need for robust network intrusion detection systems to mitigate malicious activities and safeguard critical information. There are basically two main approaches for detecting network intrusion: signature-based and anomaly-based. Signature-based detection compares current traffic to known attack patterns, but struggles to detect novel threats. In contrast, anomaly-based detection identifies deviations from normal behavior, making it more effective against unknown attacks—though it demands significant computational resources for training. These anomaly-based systems often rely on artificial intelligence algorithms or statistical techniques. While artificial intelligence methods are powerful, they require high processing power and long training times. Statistical approaches, on

the other hand, face challenges in defining accurate thresholds. Both methods analyze traffic features to identify patterns, supporting more effective detection and prevention of a broad range of network attacks [13].

2.2. Network Attacks: Detection, Effects and Prevention

With the rapid growth of Internet applications, security threats, particularly network intrusions and attacks, have escalated significantly. These threats compromise system availability, degrade service quality, and can even lead to complete shutdowns. A common method involves attackers using compromised devices to flood a target server with spoofed, meaningless packets, depleting its resources and denying service to legitimate users. DDoS and DoS attacks are especially damaging, leveraging large numbers of controlled devices to overwhelm targets by exhausting bandwidth or processing capacity. Modern attack strategies increasingly use automated tools to launch large-scale, distributed floods that are harder to detect and stop. These tools are inexpensive, easy to access, and often require minimal technical knowledge, making them highly effective in the hands of even novice attackers. Typically, attackers first identify systems with unresolved security vulnerabilities. Compromised systems then act as handlers to scan and infect other devices across networks, often using scripts to propagate the attack.

Protocol-based packet flooding can be detected by analyzing specific packet features, as network attacks often exhibit anomalies in these characteristics. This approach has proven effective in identifying and preventing intrusions. Recent studies suggest improvements to detection algorithms and propose using traffic flow features for attack identification, offering lower overhead compared to traditional methods. Techniques like Kullback Leibler (KL) distance and generalized information distance are commonly applied to detect various types of network threats. Additionally, several models have been developed for preprocessing traffic to identify large-scale attacks. However, there remains significant room for advancement in intrusion and attack detection methods.

However, these detection approaches often require significant computational time, making real-time detection of large-scale attacks challenging. As the frequency and impact of such attacks grow each year, organizations without adequate security measures risk service disruption and financial loss. Network attacks typically aim to overwhelm target servers by generating excessive requests that exhaust processing resources. Although these attacks produce high volumes of malicious traffic, they often mimic normal patterns, complicating detection in large-scale networks. Intrusion detection systems are widely used to identify such threats. IDS methods generally fall into two categories: anomaly detection, which identifies deviations from normal traffic patterns, and misuse detection, which matches traffic against known attack signatures. Hybrid systems combine both methods for improved accuracy [14]. These systems help detect anomalies caused by attacks that exploit unresolved vulnerabilities in network protocols, aiming to drain server resources with illegitimate traffic. This thesis focuses on methods for detecting and classifying such attacks effectively.

The widespread use of Internet applications has introduced various security vulnerabilities that can be exploited to disrupt service structures. Intrusions typically aim to degrade or disable services by exhausting system resources such as CPU, memory, and bandwidth, rendering services unavailable. Intrusion detection remains a fundamental challenge in cybersecurity, often evaluated using publicly available attack datasets. Many large-scale attacks, particularly those focused on resource and bandwidth depletion, can last for days and generate a high volume of requests in a short time by flooding networks with malicious protocol packets. Despite the growing sophistication of detection systems, network attacks continue to pose significant threats—especially in sectors like finance—resulting in substantial financial losses.

2.2.1. Brute Force

Brute force attacks use a technique that employs trial and error to break encryption keys, login information and passwords. It is pretty simple and straightforward method for getting unauthorized access privileges to networks and related resources.

For this malicious attack, all possible combination of encryption keys or passwords are tried until correct one is finally discovered. The attacker checks all possible inputs systematically, so the method relies on making several attempts.

Attack type refers to utilization of exhaustive effort to gain critical and sensitive information. Although approach may be dated it is still common form of intrusion and widely used. This attack is typically used when weaknesses of the system is not expected to be taken advantage of that would make the intrusion easier [15].

Attackers may utilize a variety of techniques listed below:

2.2.1.1. Basic Brute Force Attacks. Attacker does not make use of any complex tools and simply tries to guess or uses some scripts to get credentials manually. Common password combinations are mainly applied to discover correct information.

2.2.1.2. Reverse Brute Force Attacks. Attacker runs a certain known password either gathered with an intrusion or widely used. It involves systematically trying many usernames until a successful combination is found. These attacks are considered reverse in that attacker begins with known password and proceeds further.

2.2.1.3. Dictionary Attacks. This is considered as a major and sophisticated type of brute force attacks. Attacker runs potential passwords from a list or so-called dictionaries to discover the correct one. Some words are modified to get close or added certain numbers and symbols to reach a successful credentials.

2.2.1.4. Hybrid Brute Force Attacks. This is normally dictionary attack merged with a simple brute force attack. The attacker first gains access to target username and then runs through mentioned certain dictionaries with slight modifications to reach account credentials.

2.2.2. Distributed Denial of Service

A DDoS attack is a malicious attempt to disrupt the normal functioning of a network by overwhelming it with meaningless or harmful traffic. Its primary goal is to render the target server unavailable by flooding it with massive packet communications from multiple sources. As a major cybersecurity concern, DDoS has attracted significant research attention. Various methods have been proposed to detect and mitigate DDoS attacks, and organizations must adopt effective security strategies to counter them. Advancements in processing and networking technologies have also enabled attackers to launch increasingly complex and powerful attacks. As a result of DDoS activity, servers become overwhelmed and unable to respond to legitimate client requests due to resource and bandwidth exhaustion. Attackers deliberately flood the server with illegitimate traffic, leading to dropped connections and service failures in distributed environments with multiple compromised nodes [16]. Therefore, main motive for the attack is to deplete target system resources so that service is deliberately halted leading to service unavailability for network clients.

Although several classifications are proposed, DDoS attacks may be broadly classified into three categories as volume-based attacks, protocol attacks and application-layer attacks.

- Volume-based attacks: These attacks target servers by overwhelming them with large volumes of distributed malicious traffic, making them unavailable to legitimate users. This is typically achieved by sending a surge of meaningless packets to consume bandwidth or by flooding the server with Synchronize (SYN) requests, which are part of the Transmission Control Protocol (TCP) three-way handshake. Volume-based attacks aim to overwhelm the target's bandwidth by sending massive volumes of traffic from multiple sources, rendering the service inaccessible. Examples include User Datagram Protocol (UDP) floods, Internet Control Message Protocol (ICMP) floods, and other high-volume traffic attacks.

- Protocol attacks: Protocol attacks largely consume server resources such as processing power and memory. They exploit vulnerabilities in various protocols. Protocol attacks exploit vulnerabilities in network and transport layer protocols to exhaust server resources like memory or CPU. Examples include SYN floods.
- Application-layer attacks: They aim at vulnerabilities in application structures. Attackers send malformed or spoofed requests to exploit software bugs or overload processing, potentially affecting other applications. Application-layer attacks target specific vulnerabilities in web applications or services, often using legitimate looking requests to overload server functionality. Examples include HyperText Transmission Protocol (HTTP) floods, Slowloris, or attacks exploiting SQL (Structured Query Language) injection.

Effective strategies need to be employed to secure the system from harmful consequences of DDoS attacks such as firewalls to detect and filter malicious traffic or fabricated packet groups. More complicated security platforms may be deployed for overall end-to-end protection from DDoS attacks. Recognized vulnerable system issues should be cleared as soon as possible to sustain service effectively. A comprehensive and well-documented plan with proper approach surely helps to respond serious DDoS attacks quickly and keeps network security intact. Note that continuous monitoring against attacks and implementations for infrastructure resilience is definitely required to prevent DDoS attacks and distinguish between malicious and normal traffic.

Also, employees of organizations may be trained so that suspicious or malicious activities may be spotted in time and reported to resolve prior to harmful effects. Following approaches are valuable to protect the system from DDoS attacks:

- Deployment of web application-based firewall that filters or blocks malicious unwanted traffic.
- Intrusion detection and prevention systems to reinforce network security.
- Well-structured and correctly applied plan with full procedures to identify and respond to attacks accordingly.

- Regular and complete revisions of all operating systems and software to address and clear known vulnerabilities that has the possibility to be used by attackers to start major attacks.

Given the serious risks posed by DDoS attacks, organizations must implement effective security measures and establish clear response procedures. Researchers across disciplines are actively exploring solutions, with machine learning emerging as a leading approach due to its ability to analyze traffic in real time and detect anomalies. While ML models offer promising results, they require continuous tuning to adapt to evolving attack patterns. Hybrid approaches that combine multiple algorithms and datasets are increasingly favored for building end-to-end detection platforms. Neural network-based methods are among the most recommended for DDoS detection, offering better performance than traditional techniques. However, certain attacks—such as amplification-based DDoS—remain difficult to detect, as they resemble normal traffic but exploit disproportionate response sizes to exhaust resources. While some flood attacks can be managed with real-time monitoring, they still pose significant threats. Effective defense requires not only accurate detection but also rapid response mechanisms to prevent service disruption. Ultimately, successful DDoS detection depends on collecting and analyzing sufficient network traffic data to distinguish between normal and malicious behavior. Machine learning classifiers continue to gain traction for their effectiveness in identifying and mitigating these threats.

Most prominent attacks are listed and explained below:

2.2.2.1. SYN Flood Attack. TCP protocol uses a three way handshake method for establishing a connection between nodes. Attacker starts multiple connections to server by sending spoofed SYN packets continuously. These packets are not acknowledged deliberately and therefore, server keeps ports open and waits for response for inconclusive amount of time. Therefore, message handshake is not completed. Server becomes overwhelmed with incomplete connections. Consequently, SYN flood depletes target server of its resources such as connection table memory that is generally allocated for

processing incoming packets. Finally, performance of the server is degraded resulting rejection of incoming normal requests or server may completely be shut down for service loss.

2.2.2.2. UDP Flood Attack. For UDP flood, attacker sends large number of spoofed packets containing UDP datagrams to arbitrary ports of target host in large quantities with variety of source addresses. These UDP packets overwhelm target host. System tries to check for application listening mentioned ports. If no application is assigned, host replies with destination unreachable ICMP message. Thus, given large number of UDP packets, the target system is forced into sending many ICMP packets. Attack exhausts network resources and bandwidth. Eventually host becomes unreachable to clients packet. There is a lack of a full handshake method in UDP. Therefore, UDP attacks may be considered as not easy to detect. Processing large number of UDP packets consumes resources of server.

2.2.2.3. HTTP Flood Attack. For this type of attack, fabricated HTTP requests are sent massively to attack a web server or a certain application. Many compromised devices are generally used. Attacker occupies target server with meaningless HTTP requests to allocate valuable resources thus normal client access to server resources are denied. Attackers also attempt to use random HTTP headers to avoid detection.

2.2.2.4. DNS Amplification Attack. This attack uses UDP packets and therefore, handshake in protocol is not applied. The source address validation is avoided. Attacker uses open Domain Name System (DNS) resolving systems to overload the target machine with DNS responses. Spoofed DNS requests are sent to DNS resolving server and when resolving system replies, response frame is dispatched to target server. It is called amplification attack because DNS server is sent many queries and response increases payload size of the packet which as a result congests target server bandwidth. Server is unable to discern which packets are coming from a normal system and responds all requests. Consequently, network resources are depleted and system may be shut down.

2.2.2.5. ACK-PUSH Flood Attack. This attack affects the network service by consuming available resources and bandwidth. Push (PSH) and Acknowledgment (ACK) flags of the transmission protocol are associated with this malicious activity. ACK and PSH packets are sent continuously to target servers. Working prevention mechanisms may be useless. The attack may be a starting point for other harmful activity. Many PSH and ACK packets diminish server processing power, and normal users may not have required service. Handling flags may require additional processing power therefore, it is quite possible to overload host service structure via modifying mentioned flags with many harmful requests. The protocol requires that transmissions be established with a certain handshake procedure that needs processing. Therefore, complete consumption of target server resources may be the result due to unnecessary processing of sent packets.

2.2.2.6. ACK Fragmentation Attack. For this type of attack, a small number of packets with maximum allowable size are used to adversely affect the network service structure. Available bandwidth may be consumed. These packets go through several devices such as routers and attack prevention systems due to reassembling of fragmented packets are not achieved. Content of mentioned packets are basically random data. The attacker has the purpose to utilize all available bandwidth of target system. Therefore, attack diminish performance of available servers in the network.

2.2.3. Denial of Service

Denial of Service attacks are among the most common and damaging cyber threats, aiming to render target servers unavailable to legitimate users. These attacks disrupt normal operations either by overwhelming systems with malicious traffic or exploiting known vulnerabilities, often leading to service outages and financial loss. Unlike DDoS attacks, which involve multiple sources, DoS attacks are launched from a single source yet can be highly destructive, particularly against poorly secured networks [17]. DoS attacks typically exploit configuration flaws or known network vulnerabilities to exhaust system resources such as CPU, memory, or bandwidth. The

primary goal is to disrupt server functionality and deny access to authorized users. Attackers may employ various techniques, making detection challenging. These attacks deplete processing capacity and block normal service access, posing a significant threat to service continuity [18].

Widely known types of DoS attacks are explained below:

2.2.3.1. SYN Flood. Protocol handshake may be used by attackers to send SYN requests without the connection is completed. Therefore, target connection table may be affected adversely. For SYN Flood, the attacker makes use of the functionality of protocol by dispatching SYN packets with spoofed address to target system to start transmission procedure. However, attackers do not send all required replies according to protocol. Therefore, connection status is open and server waits for acknowledge packets. This attack is specific type that targets starting handshake of connection protocol. SYN flood dispatches large number of packets to target system, but result acknowledge packet is not routed.

2.2.3.2. UDP Flood. Attacker dispatches many UDP packets to several ports that results in the server processing unnecessary traffic and sending replies accordingly. For this type of attack, attacker may attempt to send many UDP packets with spoofed addresses to varying ports of target system. Target system examines each port for application information but reaches to no related program. Finally, target server dispatches ICMP packets as response to spoofed IP address but no response is received.

2.2.3.3. Slowloris Attack. For the attack, attacker causes many connections to be open by sending partial HTTP requests. Therefore, server is prevented from serving normal users. Attacker relies on employing partial HTTP requests to target server with open transmission connections. Slowloris is one of the variants of HTTP attacks. Note that it is a harmful attack which opens many HTTP connections to target web server. Also, partial HTTP requests are sent at certain intervals to keep connection statuses open.

Therefore, processing power and resources of target are largely consumed. They may be completely exhausted. As a result, system discards normal user requests. It may be significantly difficult to prevent such attack.

2.2.4. Reconnaissance

Reconnaissance or scanning attacks are often used as a preparatory phase for larger, more destructive cyberattacks. Their primary goal is to identify system vulnerabilities, active devices, and open ports [19]. The information gathered during this stage can later be exploited in severe attacks such as DoS or DDoS. While reconnaissance attacks may not cause immediate damage, they are critical components of broader attack strategies. To prevent them, organizations should monitor for suspicious activity, disable unnecessary services, restrict traffic to essential ports, enforce strong authentication mechanisms, and isolate critical systems to minimize exposure. Several known techniques are used to carry out reconnaissance attacks, with some particularly harmful types described below:

2.2.4.1. Operating System Scanning. Operating System Scanning attack is typically used to determine properties of the operating system running on targeted server. Attacker employs several methods to gather information about the target such as analyzing certain responses of transmission packets and exploring typical behavior of open ports and running services. This scanning is expected to collect detailed information about target system. Resulting data is used for further planned attacks to exploit vulnerabilities. Main purpose is to define attack characteristics. It should be noted that active scans determine Internet protocol behavior of target whereas passive methods largely depend on analyzed traffic. The method involves applying some networking tools for such scanning tasks. Identifying or deciding about the strategy concerns many as to prevent such attacks.

2.2.4.2. Ping Sweep. Ping Sweep attack is reconnaissance attack employed to determine effective servers or devices in a certain network. It involves dispatching series of ICMP Echo Request packets to a range of network addresses and then examining the ICMP Echo Reply packets that are sent to determine which hosts are active and running. This method identifies effective servers in target network. Therefore, attackers form a map for a further harmful activity. With this type of attack, network mapping may be considered convenient and appropriate since active servers send replies. Main idea for this scan effort is to find effective and running devices with their addresses in certain network to gain important information that may have several malicious effects such as another attack.

2.2.4.3. Port Scanning. Port Scanning basically identifies open and active ports on target server. As a working procedure, the attacker sends several packets to various ports of mentioned target server. Main aim for the attempt here is to form an effective connection. Responses to packets are then analyzed to determine port statuses. This scanning detects working ports on system and also gathers information about the services running on mentioned ports of the server. Mentioned attack basically makes use of the fact that open ports of a server are potential entrance points for malicious activity. Therefore, services are specified and applications are identified. It is noted that configuration issues of services and applications are certain vulnerabilities that may cause wide problems. This attack also searches to gain further information for planned attacks that may exploit some configuration issues or service problems.

2.2.4.4. Vulnerability Assessment. Vulnerability assessment attack is basically preparing evaluation or analysis that covers possible security issues and vulnerabilities that largely involves running automated tools in target network. This vulnerability scan identifies several security problems that may readily be exploited by attackers to have unauthorized access or gain privileges to mentioned system. Also, some valuable information may be captured. Vulnerability Assessment scans a target system for known problematic security related issues. Task is mainly automatic with the use of several

networking tools that gather related information. Servers or devices with not revised software or configuration errors are determined. Also, some unresolved vulnerabilities that are not resolved may also be determined for further attack planning. Identified issues of the target network may be exploited later for malicious activity. Therefore, quick precautions are required for vulnerabilities of a system.

2.2.4.5. Host Discovery. Host Discovery attack has the purpose to identify active servers or hosts in a certain network. It involves using various methods to determine network addresses of several devices that are connected and running. This attack maps servers and devices with their addresses in a certain network to discern the structure of existing transmission. Overview of all active devices are gathered and therefore, targets may be prioritized easily. Information provided may be used to plan further harmful attacks. It may be stated that such attacks assist in to discern overall network structure with possible host relationships and determine important items for further attack planning. Therefore, it is considered as the initial stage of procedure of mentioned attacks. Since such attacks are main concern for sustainable service, preventive measures that detect such attacks should be deployed and constant monitoring of resources should be planned.

2.2.5. Web Based

Web-based applications are widely used for delivering various services and are gaining increasing complexity, which can introduce critical security vulnerabilities. These weaknesses—often stemming from development errors or insufficient input validation—make web servers common targets for attackers. Web-based intrusions now account for a significant portion of modern network security issues. Attackers may exploit these vulnerabilities to gain full control of servers and access sensitive data [20].

Implementing robust intrusion detection systems is essential to prevent such attacks. Vulnerabilities must be identified promptly, and administrators should continuously monitor for suspicious activity and apply corrective actions without delay.

Some major types of these attacks are listed below:

2.2.5.1. Browser Hijacking. Attacker manipulates user web browser to redirect to malicious servers and install unwanted application to gain access to sensitive information. This type of attack is often achieved through certain malicious scripts. Users are directed to malicious websites that spread mentioned harmful scripts and applications. Browsers should be kept up to date with relevant security settings to avoid intrusions. Downloading from untrusted sources are not recommended.

2.2.5.2. Command Injection. This attack allows attackers to execute some system commands effectively from server. Method involves inserting commands in input field of sites to gain access to system resources. Vulnerabilities are exploited and server is compromised to some extent for execution of malicious commands. As input fields of web sites are prone to intrusion and several attacks, executing user inputs in system commands should be avoided strictly. Least privilege principles may be employed to limit and control impact of compromised account for security reasons.

2.2.5.3. SQL Injection. This attack uses code injection method to have privileges for web sites. Unsecured sites are directly affected with the attack. Database dense applications are main target for this type. Attacker may inject malicious SQL queries into input fields of sites to gain access to database to gather sensitive information. Authentication process may be bypassed and database content may be totally extracted. These attacks may not be totally avoided by firewalls or intrusion detection systems may lack protection. Parameterized queries and prepared statements may be used to protect against such attacks. Software and database management systems need to be regularly revised to avoid unauthorized access to system resources.

2.2.5.4. Uploading Attack. Uploading attack sends malicious files such as scripts and related programs to a target server. These files are normally disguised as harmless files. This attack is an attempt to gain unauthorized access to resources or execute arbitrary

code for malicious activities. As a result of the attack, complete server control may be achieved. Allowed file types for the server should be restricted and content of received files needs to be validated thoroughly to protect against such attacks.

2.2.5.5. Cross-site Scripting. With this attack, attacker is allowed to inject some code into site server. Malicious scripts are put into web pages that are viewed by general users. Therefore, the attackers gain sensitive information and largely control user sessions. There are basically two types of Cross-site Scripting attacks: For Stored Cross-Site Scripting (XSS), mentioned malicious scripts are stored in server and then delivered to normal users. For Reflected XSS, malicious scripts are reflected from a web application to browser of the normal user. Certain security measures need to be implemented to avoid such attacks such as validating all user inputs and protecting sessions.

2.2.5.6. Backdoor Malware. For this type of attack, a backdoor malware or program is sent and installed in site server or relevant application. Therefore, a continuous unauthorized access is provided to attackers. An entry point to system is generated and security measures are unusable. Attackers are able bypass authentication easily and gain privileges to gather critical information and execute arbitrary commands for malicious activity. This attack may cause complete server compromise or long-term unauthorized access to resources.

2.2.6. Spoofing

Spoofing attacks involve an attacker impersonating another entity to gain unauthorized access to a network or intercept data transmissions. These intrusions disrupt normal server operations by exploiting identity verification weaknesses, often aiming to collect sensitive information. Many Internet protocols lack robust authentication mechanisms for verifying the source or destination of messages. Without proper safeguards, servers become vulnerable to spoofing attacks, which can be leveraged in man-in-the-

middle scenarios. These attacks often exploit protocol-level flaws, but can be mitigated using advanced firewalls with deep packet inspection and authentication measures to validate communication endpoints [21]. DNS spoofing is a common type, where domain name records are falsified to redirect users to malicious websites. While website spoofing is widely recognized, the broader impact on domain-dependent services can be severe without appropriate countermeasures. The two major types of spoofing attacks commonly used in man-in-the-middle attacks are DNS Spoofing and Address Resolution Protocol (ARP) Spoofing.

2.2.6.1. DNS Spoofing. This type of attack manipulates Domain Name System to transfer typical users from normal service sites to sites with malicious purpose. The attack makes use of process DNS translates domain names into network addresses that are used for device transmission. Affected and compromised DNS resolver directs request to address of the attacker. Method involves attackers targeting a certain DNS server to send false DNS responses to resolver that in turn caches incorrect information. The resolver may map a normal domain name to an attacker address which results adding malicious records in the system. When users attempt to visit a normal site DNS resolver return the spoofed address therefore, traffic is transferred to attacker-controlled malicious server.

There are several impacts of spoofing that are worth mentioning. Many users are directed to fake sites to gather credentials or important information as a major issue. Disguised harmful files may be downloaded as normal ordinary downloads. Also, it should be noted that many users are blocked access to intended service. This attack allows interception of message transmission between users and normal service as harmful purpose. Prevention methods are basically checking or validating mentioned DNS replies using cryptographic signatures. Security methods like DNS over HyperText Transfer Protocol Secure (HTTPS) or DNS over Transport Layer Security (TLS) may clearly be used to prevent attacks. Also, it may be useful to reduce the service life of cached DNS entries. Also anomalies in DNS responses may be monitored or controlled.

2.2.6.2. ARP Spoofing. This type of attack exploits Address Resolution Protocol. This protocol maps network addresses to Media Access Control (MAC) addresses in networks. Attackers send fake ARP messages to associate their MAC address with network address of target server. Normally, attack is started with sending forged ARP responses to victim and the gateway. Target supposes that attacker MAC address is related with the gateway network address. Target updates its ARP table, mapping gateway network address to attacker MAC address. Therefore, all traffic from target to the gateway now goes through attacker as a clear intrusion. Traffic flow is intercepted for valuable information. This way, attacker is able to read or modify important data. This type of spoofing is mainly utilized to perform man-in-the-middle attacks therefore, attackers are allowed to dispatch harmful packets. There are several impacts of such spoofing that are noted. Intrusion allows attackers to gain sensitive information like access credentials. User sessions may be totally controlled. By disrupting ARP tables, attackers may block normal traffic to servers. Protection mechanisms against the attack are basically configuring ARP entries manually for critical devices. Switch-level monitoring and validation may be used to block spoofed ARP packets. Encryption methods may surely be employed to secure packet transmission. Also, it should be noted that isolating sensitive servers in network may help to restrict such spoofing opportunities.

2.3. Related Work

Many deep learning-based methods are developed for anomaly detection in time-series data with promising performance figures. Article in [22] addresses difficulties in understanding complex relationships of time-series data and distinguish anomalies from normal data that present similar distribution. An approach is proposed that employs time-series to image transformation with adversarial autoencoder. Method presents image transformation algorithms and CNN-based adversarial learning. The approach uses image transformation to capture local features of near time points. Method consists of two CNN-based adversarial autoencoders that competitively learn to differentiate normal and abnormal data. Several real time-series datasets are analyzed, and reports

show that model achieved significant anomaly detection performance if compared with currently available methods. It is added that, model with GASF achieved the best performance figures. Model provides intuitive interpretation of results and thus improves usability of the work. It should also be added that, a valuable method is presented to improve robustness of anomaly detection in time-series data in [23] by using ensemble learning combined with several time-series to image transformation procedures. The need for more advanced methods that improve accuracy is obvious. Presented aim is to gather wide range of patterns in transformed time-series data. The method combines powerful characteristics of multiple models to increase accuracy of anomaly detection. Obtained results provide a reference for selecting appropriate time-series transformation method based on specific properties of utilized dataset. This reference may be useful to select proper transformation method based on statistical parameters for a particular dataset. Importance of correct time-series transformation algorithm for reaching significant outcomes is clearly shown. Performance of various time-series to image conversion procedures are analyzed and a reference to select appropriate algorithm is presented in such studies. Considering several research and applications in the area over years, it is obvious that deep learning studies achieved superior performance using time-series data in classification. However, it may be difficult to uncover hidden patterns in time-series. In [24] authors mention that, converting time-series data to image may improve performance of the neural network model. A comprehensive review of several notable studies which employ time-series to image conversion techniques and application cases are required for developing the argument. Therefore, various time-series to image algorithms such as GASF, Markov Transition Field (MTF) and Recurrence Plot (RP) are discussed. They argue that, process enables better analysis and interpretation. Visual presentation of temporal data may certainly enable an intuitive interpretation and pattern recognition. It may be stated that, temporal dependencies are kept as major benefit for better analysis. Thus, it is clearly understood that, integration of time-series data as images with deep learning models may be readily implemented for anomaly detection and classification.

Deep learning is currently an exciting subject area which is frequently studied by researchers to develop or advance intrusion detection systems used for computer networks. In a notable survey [25], deep learning-based intrusion detection systems are detailed. Mentioned survey provides a thorough analysis and classification of several implemented procedures. Major background concepts about system architecture are expressed. Classification is achieved according to type of deep learning model utilized in systems. Evaluation metrics and datasets are compared for categorization effort. It may be useful to understand how mentioned deep learning methods recognize several types of intrusions. Comprehensive analysis of several intrusion detection frameworks is presented. Advantages and limitations of applied methods are expressed. Also, future works are recommended. It should be noted that, developing accurate and superior performance intrusion detection system to handle several types of network attacks is really challenging. Therefore, deep learning is a significant area to consider for network intrusion research which presents substantial opportunities. Note that, anomaly detection is studied in various of application domains and therefore, significant progress is achieved. Deep learning-based approaches are attracting a considerable amount of interest recently due to their superior capability to resolve anomaly related problems. After CNN gained considerable success in the field of computer vision, researchers readily implemented CNN for time-series data analysis. Several papers considered deep learning as useful context for anomaly detection in time-series data as in [26]. Significant background information about anomaly detection in time-series data is provided and latest real applications are carefully examined. Many deep learning based anomaly detection methods for time-series data need to be examined. Also, characteristics of time-series data need to be discussed with anomalies that may exist. Authors explain how deep learning methods acquire complex relations in time-series data. Such models make no prior assumptions about underlying patterns and this characteristic makes them attractive option for time-series analysis currently. Validation that deep learning methods outperform several classical methods is presented. It is suggested that, such models are substantially capable of processing large sequences of data. Traditional approaches also provide useful information. Several deep learning models for time-series analysis are compared using well known available datasets. Guidelines

for proper model selection and training strategy for deep learning-based time-series anomaly detection is considered helpful. Applications of anomaly detection in several industries are shown. It is claimed that, CNN-based fault detection and diagnosis models achieved success in analyzing time-series data. With the significant progress in neural network research, several learning-based models are developed with emphasis on clustering and classification to distinguish normal traffic and malicious attacks. For instance, article in [27] analyzes several learning-based models for network traffic classification problem. Such studies also present a certain roadmap for research in network traffic classification subject. Authors categorized existing studies as supervised and unsupervised based on learning models for network traffic recognition. Technologies and methods for network traffic classification in anomaly detection studies are examined. Recommendations and limitations are presented for mentioned learning methods. It is stated that, supervised learning models and machine learning algorithms may properly be applied for traffic classification. Network traffic classification problem thoroughly investigated and several procedures are recommended to reach a successful outcome. Note that, supervised learning-based network traffic classification requires a reliable dataset with labeled traffic types. Dataset certainly should include several traffic types. Preparing labeled data may result in certain costs. Also, a certain feature engineering is needed to be achieved. Therefore, a certain preprocessing is required for accuracy. However, it is clear that, learning-based models contribute to solution for network traffic anomaly detection.

In a research article [28] concerning DDoS attacks, many incidents spanning 10 years are analyzed in detail. Significant characteristics of network attacks are presented with their major impacts. It is found that, frequency and sophistication of attacks are noticeably increased. DDoS attack trends with historical data are presented. Dynamic and quite adaptive security strategies need to be implemented. It is claimed that, diverse attack techniques, not just typical ones, should be addressed for better security implementations. Approaches based on data analysis are required to diminish cyber-threats. This notable study presents several clues that may be used to design effective prevention platforms. It is mandatory to explore further than conventional detection

methods given the complexity of current DDoS attacks. For this aim, thorough research articles are considered very useful to understand several aspects of network attacks and possible security solutions. A detailed survey [29] mentions various properties of network DDoS attacks with a relevant classification. Typical network vulnerabilities and security issues are discussed. Current network attack detection solutions with emphasis on approaches that utilize machine learning are examined. Certain limitations that may reduce effectiveness are expressed. It is clear that, such articles provide a certain help to grasp network intrusion context and related research efforts especially for DDoS attacks. Also, contribution is obvious for the development of security solutions. Such an article [30] discusses several machine learning methodologies. Many researchers utilize mentioned methods for network intrusion detection to improve accuracy. They provide a review of literature for the area spanning years. Several models are constructed to evaluate the effectiveness of mentioned algorithms. It is stated that, success of deep learning methodologies surely depends on proposed model and tuning of related parameters. Such articles are really useful for research activities that may achieve thorough analysis of the subject area.

An effective DDoS attack detection method based on information entropy and deep learning is presented in [31]. Accuracy of traditional methods may not be significant enough for practical applications. It is stated that, malicious traffic may be detected through information entropy analysis. So, a CNN model is implemented to differentiate between normal and malicious traffic by examining network packets. Validation of the approach is performed by comparing with SVM and Decision Tree. Accuracy of mentioned method reaches almost 99%. Therefore, capability to detect DDoS attacks is expressed. Traditional methods require intervention and intrusion detection systems significantly experience false indications. Also, it may not be easy to distinguish real attacks from false indications. Authors tried to merge advantages of information entropy and deep learning studies. Method is claimed to have low complexity of information entropy detection and superior accuracy of deep learning model. It is presented as proper network security implementation. Thus, neural networks may achieve superior performance figures for network attack detection. It is also added

that, design of secure systems against network attacks are typically important for large enterprises or organizations. A recent paper [32] discusses a DDoS attack detection system. Existing solutions have several disadvantages such as static thresholds and therefore, may not adapt to variations in network traffic. This article presents an entropy-based dynamic threshold detection system. System consists of sections such as feature extraction, malicious activity detection, indication generation and threshold update. In the study, source address is selected as random variable for calculation of normalized Shannon entropy. Malicious activity is detected when calculated entropy is lower than a certain threshold. System generates a certain indication. Authors propose a dynamic threshold algorithm for variations in network traffic. Chebyshev's theorem is used as basis for procedure of mentioned threshold update. Validation is achieved with a known available dataset. Proposed detection system is shown to have significant performance metrics. Note that, [33] argues, there is not adequate research concerning time-based features for classification of several DDoS attacks. Many time-based features for DDoS attack detection are examined. Deep learning classifiers are very valuable in accurately detecting several DDoS attacks. Several notable research concerning deep learning suggest the advantages of smaller feature sets especially in reducing training time. Findings indicate that, majority of models provide almost 99% accuracy. It is indicated that, model training time for time-based features is considerably lower. Authors note that, accuracy is not compromised and a small time-based feature subset may be adequate for attack detection. DDoS attack severity may be managed by using traffic classification to characterize malicious network traffic. After properties of attack are determined, relevant strategies may be employed to mitigate the attack. Thus, a small number of time-based features may be implemented for successful DDoS attack detection. Large-scale defense mechanisms for network attacks need to be explored. In [?], authors claim that, current DDoS prevention methods generally increase controller occupancy and consume valuable resources. Therefore, a delay may be experienced in recognition of attacks. It is noted that, such schemes are typically suited for single controller systems. If scalability considerations are taken into account, it is really a significant limitation for large-scale networks. A learning approach to resolve mentioned limitations may be required. Proposed method is a de-

centralized machine learning approach that distributes the task to several controllers. Aim is to prevent DDoS attacks especially in large-scale networks with multi-controller architecture. Methodology is validated with publicly available datasets. An accuracy exceeding 98% is reached. It is noted as a significant success if compared to other related works. Proposed approach achieves scalability for network attack detection and lowers consumption of processing resources. Also, a recent article [34] discusses an interesting neural network methodology to detect DDoS attacks. Usage of Graph Neural Networks is presented. Multiple models are added together to improve success figures. It is argued that, such methods perform better than machine learning models based on statistical features. Fairly simple graph networks may be utilized in place of computationally intensive models with straightforward structure. Approach consists of studying network topology that may be represented as graphs. Therefore, a certain traffic representation is expressed. Mentioned neural network processes graphs. Fewer convolutional layers for neural network model are used. Predictions of several models are combined in a way to enhance accuracy. Accuracy reaches 99% for publicly available datasets. Another article [35] proposes a Naive Bayes-based machine learning method for network DDoS attack detection. Fast and reliable detection of intrusions is really important. It is stated that, Naive Bayes algorithm is well-known for classification problems in machine learning studies. It is widely used and fairly simple to implement. Also, training requirement is quite lowered. However, it is argued that, method runs fast and performance may be considerable for typically difficult classification tasks. Success rates reaching 99% is reported for detection of DDoS attacks with application of mentioned method. Applicability of machine learning methods for network intrusion detection is noted for possible research efforts in the area. [36] studies one of several DDoS attack detection methods called as behavior-based approach. The research article aims to analyze user behavior to decide whether network attack occurred. User behavior shows differences when various time intervals of the day are considered. Network packet traffic is clustered to determine usage patterns. Therefore, malicious activity may be detected. Authors use machine learning algorithms such as Support Vector Machine and Random Forest for classification. Publicly available datasets are utilized to validate mentioned approach. It is argued that, method per-

forms better than implementations that are based on certain thresholds. Significant figures for performance metrics are reported. Another research article [37] proposes a DDoS attack detection mechanism using entropy as the major property to study. A calculation method which combines several entropy types for network packet traffic is offered. It is stated that, traditional methods that employ entropy has several issues such as slow response and low accuracy. Entropy value of the attack traffic is significantly lower than normal figure with proposed method. Therefore, it may be possible to determine network attack with this property. It is argued that, method is fast compared to traditional approaches and may be used as a reliable network attack detection mechanism. However, it is noted that, lower density network attacks may not be discerned due to not obvious entropy difference. Statistical methods for time-series analysis may also be utilized with the aim to detect DDoS attacks as investigated in [38]. Traditional detection mechanisms use packet-based analysis for extraction of features. It is noted that, these methods may have success issues with new types of DDoS attacks. Therefore, authors argue that, time-series analysis may be a significant alternative to traditional approaches. Some fundamental statistical parameters that may successfully be used to recognize network attack patterns. A dataset is used to form time-series and statistical analysis is achieved. Thresholds are determined for validation efforts. Accuracy figures reaching 98% are reported. It may be stated that, time-series analysis is clearly a proper area for research efforts. Also, [39] argues that, several machine learning methodologies process features and produce better outcomes if compared with traditional applications that make use of certain indications or patterns. It is noted that, feature selection is important part of properly implemented detection process. Therefore, authors apply a certain preprocessing for the dataset that may be denoted as feature engineering. Validation of the model is achieved with publicly available dataset. Significant accuracy is reported for convolutional neural network model. It may be stated that, usage of convolutional neural networks with a suitable preprocessing for features definitely improves detection success. Also, several research articles indicate that, deep learning methods are preferred regarding accuracy figures for network attack detection.

As already described, DoS attacks are a major security issue. Many researchers in subject area clearly note that there exist many types of DoS attacks. In [40] it is mentioned that existing approaches may not be able to recognize various types of DoS attacks. Therefore, a new approach is proposed by extracting multi-features of traffic and then using a backpropagation neural network for classification. Several features of traffic are collected. It is noted that, low number of features may result in inaccurate outcome. However, large number of features definitely increase overhead. Therefore, a fair number of features are utilized for the study. Authors indicate that, selection of suitable features is the major part of mentioned method. The backpropagation neural network classifier is preferred as it is claimed to have better accuracy. Findings indicate that, accurate result probability of proposed approach is near 99% which is considerable for distinguishing several DoS attacks. Feature-based detection in general has better performance figures such as detection rate if compared to threshold-based detection. Another approach that employs deep learning is examined in [41]. A hybrid deep learning model that includes CNN to detect DoS and DDoS attacks is implemented. It is argued that, traditional intrusion detection systems have certain limitations such as low accuracy and inability to clearly distinguish new types of attacks. Therefore, it is claimed that, advanced approaches such as deep learning models are required to handle such issues. Since such models do not depend on some set of clearly defined rules or signatures, new types of intrusions may be discerned. CNN with typical layers is used and feature extraction is achieved in the study. Long Short-Term Memory (LSTM) cell is added to enhance the proposed model in a way to recognize long-term patterns. Success rates reaching 99% are achieved. However, it is noted that, success depends largely on the quality of training dataset and proper tuning of mentioned model. Also, real-time issues need to be addressed. In the article, several machine learning approaches are reviewed that may provide some useful information for research purposes. Also, in [42], anomaly detection systems based on several machine learning methodologies are proposed. Several feature selection mechanisms to improve accuracy of the outcome is examined. Validation is performed with a publicly available dataset. Significant accuracy figures are reported that indicate success of mentioned methodologies. Such articles provide a valuable information for current status of network attack detection

research that utilize machine learning approaches. It may be stated that, current research in learning-based models really contribute to success of network attack detection solutions.

A typical network intrusion detection system is investigated in [43]. Aim is to utilize a deep learning model to implement a system that detects and classifies various network attacks. Emphasis is put on developing an adaptive system that analyzes network behavior to prevent new and known intrusions. Intrusion detection systems may have difficulty differentiating anomaly cases exactly. Therefore, intelligent solutions need to be deployed. Proposed deep learning system has CNN with required convolutional layers, pooling layers and fully connected layer. Features in dataset that represent traffic characteristics are used for CNN processing to reach outcome. A certain tuning approach is also incorporated. Authors report 95% accuracy for the dataset utilized. It is claimed that, CNN with tuning approach showed significant success compared to similar studies. Also, a distributed machine learning-based approach for intrusion detection may be utilized as presented in [44]. An anomaly-based method with autoencoder is implement that reaches significant accuracy. Possible anomalies are detected by computing difference from a preset level of normal traffic. Training of the model is decentralized and achieved by distributed processors. Performance figures are significant with publicly available datasets. Therefore, it is argued that, autoencoders may be used for intrusion detection especially for large scale implementations. Further research and validations may be performed to gather better accuracy. It may be proper to express that, distributed machine learning-based algorithms certainly need further study. However, a deep learning model for intrusion detection is studied in [45]. A comprehensive analysis of several parameters that underline the working structure of the model is achieved. Patterns from network packet traffic are acquired. It is mentioned that, current intrusion detection systems have substantial limitations that novel malicious activity may not be detected clearly. Therefore, research community is spending efforts to incorporate neural network models with the aim that, better success rates may be achieved especially to for complex and not previously encountered network attack profiles. Authors indicate that, deep learning models show greater ca-

pability to recognize and reveal complicated patterns. Publicly available datasets are used to present the performance of mentioned model. Accuracy figures reaching 98% for classification are reported. It is claimed that, for model studies, computational complexity and ease of training are really considered as these parameters play a major role for real-time applicability. Also, in [46], machine learning methodologies are studied. Decision Tree and Random Forest algorithms are implemented for network attack detection. Several preprocessing for input data is applied to reach better outcomes. Also, effort is spent for feature selection. Authors report significant results for well-known accuracy metrics. Therefore, it may be stated that, machine learning methods may successfully be used for network intrusion detection with certain forming effort for input data. Note also that, several deep learning models are discussed in [47]. It is mentioned that, successful outcomes are reached with deep learning methodologies recently. A typical CNN model is formed with many layers. Model is used for classification of network attacks. Validation of proposed method is achieved with a publicly available dataset. Accuracy figures reaching 98% are reported for network attack detection. Therefore, deep learning methodologies may be studied further for intrusion detection as clearly indicated with presented achievements. In another article, authors in [48] propose a network attack classification methodology. Existing systems may have difficulties for new types of network attacks. Several machine learning systems are reviewed. A model based on Decision Tree is utilized. Accuracy figures reaching 99% are presented for a publicly available dataset. Also, in [49], a comparison of several machine learning methodologies is presented. Related models are implemented with mathematical toolsets. Also, certain preprocessing is achieved and features are obtained. Classification results are discussed. Authors note that, Random Forest is quite successful if accuracy is concerned. A different approach is also proposed that has ensemble learning model in [50] to resolve class imbalance problem. It is stated that, classification performance may be improved with this methodology. A deep learning model is constructed. Many classifiers are used. Validation is performed and authors report significant figures of accuracy for the method. Also, in [51], a detailed survey is presented that provides notable information about deep learning-based classifiers. It is argued that, CNN is known to handle complex tasks. Also, these neural network

models are really employed in network attack detection platforms. Therefore, such articles provide a thorough analysis of CNN implementations and add to understanding of models.

[52] proposes a deep learning scheme for traffic classification. It is noted that, although several algorithms are examined for the task, process depends mainly on carefully selected or designed features which may restrict expected performance especially in different network conditions. A CNN with a certain mechanism to uncover intrinsic relations and features is used. Different size kernels are used as part of convolution process of CNN to extract defining features. It is claimed that, validation with used dataset express significant classification results. Therefore, requirement for expert efforts in feature extracting is lowered. Authors argue that, study may be considered as an improvement for generalization that is expected to cover various network conditions. Merging deep learning methods with time-series analysis algorithms is currently a trend for solving various challenges in traffic classification area. In [53], it is noted that, generalization property of a representation is very important aspect to consider for such studies. A deep learning implementation for traffic classification based on graph convolutional neural network is proposed. This model determines traffic features hidden in time-series with graph representations. Therefore, problem is expressed as graph classification effort. It is claimed that, a significant accuracy improvement is achieved compared to current well-known methods. Thus, effectiveness of convolutional neural networks is expressed for detection and classification tasks. With the growth of Internet, traffic analysis and classification has become a very important task from management perspective. A recent article [54] state that, research community is spending serious effort to implement intelligent systems for effective network management and improved security. Certain mechanisms that are able to characterize traffic accurately, are required to make proper decisions and resolve issues accordingly. A valuable security implementation requires assessment of threats which definitely leads to attack detection and classification. A methodology is proposed to examine network traffic with machine learning schemes such as KNN and SVM. Statistical L-moments that give several information about distributions are also applied. Authors argue that,

method utilizes fewer data and produces substantial classification results with a publicly available dataset. Also, computational complexity is less. It is noted that, method requires careful feature selection and suitable preprocessing for better results. Also, number of scenarios studied is limited thus, further studies are necessary. Note that, correlation between network traffic and related application is required to guarantee quality of service. Also, such recognition is important to determine malicious behavior. In [55], a useful review of traffic classification algorithms such as deep packet inspection and machine learning approaches are provided. Several implementations are also discussed. Authors argue that, deep learning methodologies present substantial accuracy for traffic classification. However, computational resource requirements are noted. Also, in [56], deep packet inspection methodology to recognize malicious payload is proposed. Pattern-based methods may have some issues with dense traffic. Therefore, machine learning methodologies are implemented. Validation is performed with certain datasets and significant accuracy figures for detection are reported. In another article [57], network attack detection system based on deep learning is investigated. A typical neural network model is proposed and certain feature processing is achieved. Validation of the model is performed with publicly available dataset. Also, parameter tuning of the mentioned model is implemented. Authors report significant accuracy figures. However, computational requirements need to be considered. An evaluation of several machine learning approaches for classification of network attacks is presented in [58]. Also, preprocessing is performed and feature vectors are obtained with the publicly available dataset. Significant performance figures are reported. It is noted that, LSTM is particularly successful if accuracy is considered.

In [59], it is stated that, several deep learning-based anomaly detection methods for time-series are proposed recently due to significant success of studied models. A thorough comparison of conventional methods and several machine learning models is achieved. Authors argue that, although substantial success of deep learning models, conventional models such as Matrix Profile may also be used properly for anomaly detection studies. Also, research article [60] argues that, matrix profile may be used for anomaly detection of time-series data. Analysis of subsequences for the series may be

used as an indicator in methodology. Authors note that, although supervised methodologies present excellent outcome, unsupervised approaches may be preferred due to significant issues such as training cost of models. Network performance indicators are analyzed with matrix profile for anomalies. Quite successful results are presented with application of matrix profile. Another study [61] investigates matrix profile method for detection of network anomalies. It is noted that, several anomaly detection algorithms have significant computational cost. Matrix profile method may provide a proper resolution for such issues. Authors validate the method with publicly available dataset. Significant results are reported. Therefore, it may be stated that, matrix profile method is quite applicable for network anomaly recognition. Also, utilization of matrix profile method for anomaly detection in data expressed as time-series is investigated in [62]. Authors note that, the method is effective for recognizing patterns in time-series so that, anomalies may be captured. Validation for the method is also achieved. Article provides a comprehensive and useful information of matrix profile method that may be used for application implementation.

3. NETWORK INTRUSION DETECTION METHODOLOGIES

3.1. Matrix Profile

One of the algorithms used for network anomaly detection in time-series based traffic is the matrix profile method. Matrix profile is a time-series based analysis method used to recognize network intrusions and attacks. Note that some examples of network attack symptoms include high processor usage, high memory usage or certain increase in workload [63]. Commonly used features for the analysis are forward packet length (maximum, minimum, standard deviation, mean), reverse packet length, reverse inter packet arrival time, and forward inter packet arrival time. Since the procedure is performed on one dimensional time-series, the features will be validated by selecting them one by one and the most effective features in anomaly detection will be considered. In addition, matrix profile method and other distance based analysis techniques usually perform quite well on systems with relatively low capabilities such as Internet of Things (IoT) devices. Network time-series feature data may be analyzed with several mathematical methods [64, 65]. Test data for evaluation contains labeled attacks, so performance may easily be studied. Matrix profile as one of such methods performs effectively for analysis with low required parameters. It should be stated that, matrix profile is utilized to detect anomalies in time-series. Method basically calculates similarities in time-series. A sequence from time-series is compared to every other sequence of the same length within time-series. Distances are then calculated and stored. Distance is widely considered as metric for similarity. If a sequence has low minimal distance, a characteristic is determined to be present in time-series. If minimal distance of sequence is relatively high, it is accepted as unique in time-series. This property is very well suited to find outliers that may indicate attacks.

As established process of matrix profile method calculation, each windowed sequence of length n is compared with every other sequence of length n in time-series.

Note that, a sequence has a distance of zero from itself which is a trivial match. After distances are calculated, they are stored in a matrix. The minimum of each row is stored. This value indicates minimal distance of given sequence from any other sequence in time-series. Matrix profile as a detection method is significantly effective for recognizing anomalies while allowing a certain level of typical noise. Major benefit of the method is low parameter requirements as there is only one parameter n to be defined for analysis. The algorithm is very robust if several values of parameter n is assessed. Therefore, fine-tuning is considered mostly unnecessary. It should also be mentioned that, concept of distance as examined in the method, may not easily correspond to accepted metrics of qualification for a typical classifier. Therefore, a relevant threshold is determined that is minimal. However, this threshold should be able to successfully identify attacks as the main objective.

This method has three basic components: Euclidean distance, distance matrix and matrix profile. For points in an n dimensional Euclidean space, the formula is expressed as

$$\mathbf{d}(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}. \quad (3.1)$$

Euclidean distance between points p and q is represented in Figure 3.1. When calculating distance matrices, the Euclidean distance of a time-series with itself is calculated using sliding window algorithm. Sliding windows proceed one by one. The matrix profile contains the minimum Euclidean distance value in each row of distance matrix. Anomaly detection is performed using these values. After the matrix profile is constructed, a threshold value is selected and values above this threshold are labeled as the attacks. Values below this threshold are labeled as normal. An example time-series of length m can be seen in Figure 3.2.

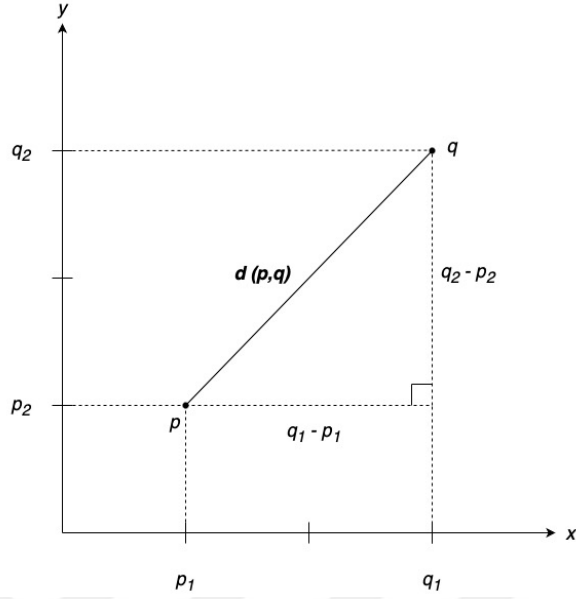


Figure 3.1. Calculation of Euclidean distance between two points.

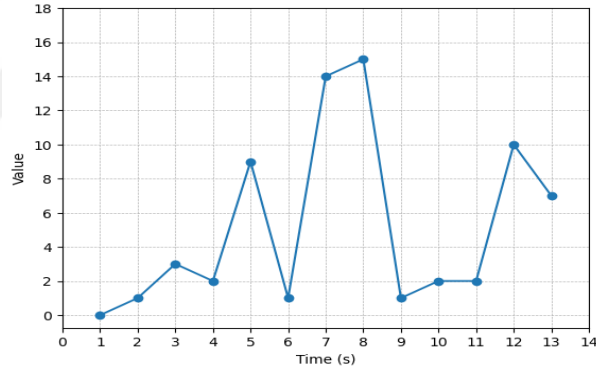


Figure 3.2. Example of a time-series sequence.

As seen in Figure 3.3, the Euclidean distance of given time-series with respect to itself is calculated using sliding windows that iterate one by one. The sliding window size n is defined as 4. In the first iteration, two vectors in the time-series data are compared. The sliding window which is denoted in blue has size 4. The red sequence has the same size as the sliding window and is fixed in the first iteration. In each step, the sliding window shifts to right by one cell. The Euclidean distance between these 2 vectors is calculated and the values are kept in an array called the distance array. The distance array has size $n - m + 1$. In the second iteration, the red sequence shifts to right by one cell. The sliding window is reset. The same steps are applied until the end

of the array and similarly the Euclidian distances are written into the distance array.

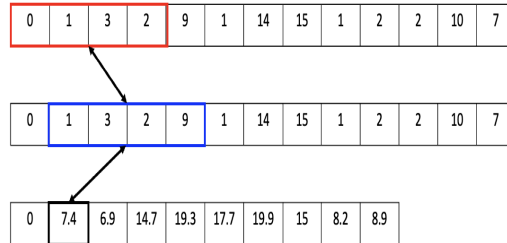


Figure 3.3. Calculation of Euclidean distance and distance array.

When the distance array is calculated for the entire time-series, the distance matrix and matrix profile are constructed as seen in Figure 3.4. Euclidian distances are combined in a matrix called the distance matrix. The minimum element of the distance matrix except zero is taken in each row and added to the matrix profile. Therefore, the matrix profile is constructed. As seen in Figure 3.5, matrix profile values are high in time intervals $t = 7$ and $t = 8$ that may be classified abnormal in the time-series. Matrix profile detects the anomalies in the time-series.

0	7.4	6.9	14.7	19.3	17.7	19.9	15	8.2	8.9	6.9
7.4	0	10.9	7.9	15.7	18.8	19.1	15.8	1.4	8.4	1.4
6.9	10.9	0	16.8	16.1	13.6	18.8	14.1	11.6	6.2	6.2
14.7	7.9	16.8	0	16.8	19.8	18	19.4	8.2	13.4	7.9
19.3	15.7	16.1	16.8	0	20.7	23.6	18.7	15.3	11.4	11.4
17.7	18.8	13.6	19.8	20.7	0	19.2	23.1	19.8	14.4	13.6
19.9	19.1	18.8	18	23.6	19.2	0	14.1	20.1	20.5	14.1
15	15.8	1.4	19.4	18.7	23.1	14.1	0	16.2	16.1	1.4
8.2	1.4	11.6	8.2	15.3	19.8	20.1	16.2	0	8.6	1.4
8.9	8.4	6.2	13.4	11.4	14.4	20.5	16.1	8.6	0	6.2

Figure 3.4. Calculation of distance matrix and matrix profile.

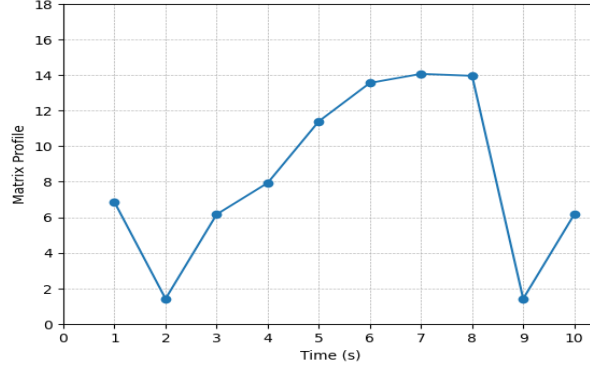


Figure 3.5. Time-series containing matrix profile values.

3.2. Time-series to Image Transformations

Valuable data expressed as time-series plays a major role in network security area by delivering significant insights about behavior and patterns of network devices that may occur as anomalies. Therefore, network packet time-series data may be utilized for intrusion and attack detection. A time-series is a sequence of data points collected at regular intervals over time which can be given as

$$X = \{(t_1, x_1), (t_2, x_2), \dots, (t_n, x_n)\}, \quad x_i \in \mathbb{R}^m \quad (3.2)$$

where n is the number of time-series data points and m is the vector dimension. A time-series may be stated as univariate or multivariate. If m equals 1, X is univariate. Therefore, univariate time-series includes a single variable observed over time. If m is greater than 1, X is called multivariate. Therefore, it is said that multivariate time-series has multiple variables observed over time. Note that, it may be difficult to gather relationships between variables in multivariate accurately. Regarding the fact that these entries have high dimensionality, certain challenges may be experienced when it comes to analyzing such data. Therefore, univariate series is easier to implement than multivariate. Although multivariate is more complex, it may be more accurate due to the fact that more characteristics may be analyzed with available variables.

Network time-series data is generated from large number of features that are captured from actual traffic. Such time-series data is widely used in various network issues such as anomaly detection. Time-series traffic data from a typical network may be significantly large and difficult to analyze to reach results for security related decisions. It may be stated that large scale network time-series data has several typical characteristics that separate it from other types of data and affect through interpretation. Simultaneous monitoring of many transmission lines and connections may be achieved with advanced tools and platforms. It is proper to state that, network packet time-series data include temporal correlations and complex spatial correlations. Also, data may seriously be affected by noise and missing values which frequently occur due to transmission issues. Conventional time-series analysis techniques are not directly applicable due to properties of network time-series data as already mentioned. Examining characteristics of network time-series data is necessary for intrusion analysis and therefore has substantial effect in decision-making for management perspective. Note that packet data is continuously produced, which means that real-time processing is required to reach the outcome [66].

Recently, subject of network anomaly or attack detection has gained considerable attention in researcher community. This is partly due to increasing complexity of communication networks that contribute to difficulty in detecting several attacks. Several studies have explored diverse approaches to enhance the accuracy and efficiency of anomaly detection mechanisms with a particular interest in leveraging representation of data as image. GASF is one of the popular and well-researched techniques for time-series to image transformation as also revealed with this study [67]. For efficient network management practice, it is very valuable to identify type of the attack in addition to knowledge of whether attack occurred so that appropriate measures may be deployed quickly to prevent or diminish the attack. If machine learning principles are concerned, a certain binary classifier that shows only if attack occurred needs to be enhanced as to handle multi-label classification. This area continues to produce considerable academic research effort due to its complexity and broad application possibilities.

Currently, many studies have focused on time-series data transformation into an image format due to substantial success achieved in industrial applications, particularly such as anomaly detection and fault diagnosis. As such, GASF is a commonly used image transformation method for anomaly and fault resolution studies in general [68]. Image transformation methodologies need to be analyzed and detailed from several aspects. As first step, existing studies are scanned based on transformation technique and application areas. This analysis work mainly details the utilization of a certain image transformation technique in networking. This thesis work presents through examination of current research for intrusion and attack classification applications. Addressing mentioned security issues, a certain two-stage GASF and ML-based algorithm is proposed and studied thoroughly. Once features extracted from network transmission packets are encoded as time-series, GASF algorithm that is applied to polar expression achieves image conversion as first stage. Image data may emphasize and uncover some characteristic attributes that may lead to multi-label classification by CNN as second stage. Therefore, more information is gathered by image conversion of captured data for attack detection purposes. Note that, time-series conversion approach is justified with neural network studies. Deep learning models such as convolutional neural networks are widely used and effectively uncover hidden patterns and features from data thus making them favorable for complex anomaly detection. Therefore, these models have gained popularity and replaced manual feature extraction. Better feature extraction capabilities are considered as an important aspect for powerful performance figures.

As an important point to mention, GASF visualization of distinguishing properties for time-series data really has great effect on correct categorization outcome as image progresses through convolution layers of neural network. Visualization concept as applied to CNN feature extraction remarkably enhances overall recognition process that leads to correct attack classification. Achieved classification accuracy clearly indicates reliability of this work. Methodology consists of conversion of time-series effectively and application of precisely adjusted certain deep learning model that handles the process. The fact that neural network architectures produce correct classification

once proper inputs are determined and applied proves favorable aspect of deep learning studies. Many researchers explored the use of GASF as time-series to image conversion algorithm for anomaly detection studies in general. Mentioned approach achieved notable success with certain accuracy to identify anomalies especially in fault diagnosis and failure prevention issues with industrial applications. Researchers captured fault issues with image transformation and application of CNN. Results demonstrated that image representation is powerful in identifying attributes that are not readily reached and applicable to solve such industrial process and quality issues. Some studies utilized GASF as time-series to image transformation method. GASF algorithm as applied in this work transforms raw time-series data into a visual representation that may be further analyzed. Unlocked latent patterns and relationships express more details and contribute to a thorough analysis to reach conclusions of the issue encountered. Application of the algorithm to convert time-series into image provides a valuable view for uncovering hidden elements that may not be apparent in raw data. Therefore, diverse application of GASF by researchers for detecting and qualifying anomalies in such a broad context illustrates usability of transformation of raw time-series data to image for further processing. While several studies explored application of GASF in industrial anomaly detection, there absolutely exists a certain need to expand research and studies into wider aspect. This study addresses this object and contributes to widen the context by evaluating a well-known and studied image conversion method within network attack detection.

Critical network packet data characteristics may be expressed and understood properly and easily with the help of image-based algorithms that utilize Gramian matrix. Such images then may be fed to CNN for deep learning attack characteristics for classification from actual data without dealing with complex statistics. It is the expectation that, multi-label deep learning studies combined with visual representation of time-series considerably enhance overall effectiveness of intrusion detection as clearly obtained with numerical analysis of the results. Transformation of raw time-series data into clearly understood visual representation for capturing uncovered or hidden patterns that indicate attacks is aimed by this work. Utilizing powerful as-

pects of image transformation combined with deep learning models to achieve superior anomaly detection performance is the major motivation for the study. Overall, this thesis adds a deep learning layer to time-series to image conversion algorithm that achieves a valuable contribution to the subject area considered.

There are various image transformation techniques described in literature. While these methods are directly applied to univariate series, typically they are not employed directly on multivariate series. Various fusion methods are discussed in studies to address such issue. Feature or image fusion is basically a process that is applied to merge necessary information from several features. When converting multivariate data into images, fusion methods may be used to combine information from different variables or data sources to create a single image representation. Note that, each variable may be converted to images separately and then these images may be combined at later stage for instance in decision. Hybrid fusion is also performed in both decision and feature levels as encountered in some studies. Also, novel application platforms may easily be formed that demonstrate success of findings of studies. Presentation of proposed algorithm for application in detection platforms is substantial achievement of this work. GASF method involves transforming scaled time-series into polar coordinate system where numerical values are expressed as cosine values and timestamps as radius. GASF adds new characteristics to advance performance of CNN [69]. Therefore, it is considered that mentioned approach suggests implementation of advanced classifier for type detection. Using image generation as extra layer in the system leverages current research on successful image-based ML methods. Therefore, this thesis utilizes such findings to classify diverse network attacks effectively using a GASF-based time-series to image transformation and CNN deep learning classifier.

3.2.1. Gramian Angular Summation Field

Image transformation converts time-series data to visual representation, such as image. It is considered a very valuable and useful method within networking area. The method reduces network packet data overall dimension into compact visual format and

also extracts key features and patterns from time-series. Additionally, as an important aspect it integrates effectively with machine learning algorithms such as CNN. Therefore, image transformation enhances analysis, interpretation, and utilization of time-series data for networking studies. Transformation process of time-series data into image is application of mathematical procedure. Note that, grid structure and characteristics of images make it attractive for network anomaly detection using neural networks. Initial step consists of converting time-series to polar coordinates. Then, image transformation computation is performed. As final stage, convolutional neural network is run phase and evaluation metrics are examined to emphasize the strength of mentioned process [70].

Datasets contain relevant data of attack types that are used as input to methodology. Features that are expressed as time-series are processed with GASF mathematics to reach image outcome. Considering the given analysis, this study converts extracted traffic time-series features based on idea of inner product and Gram matrix, and then further processes to help for attack detection. It should be stated that, GASF is a method of processing one-dimensional time-series data based on the idea of the Gram matrix. First, whole process may be stated as follows: As start of conversion of traffic packets to GASF image, feature time-series is scaled to interval of $[-1, 1]$. Method continues with assigning scaled elements to their corresponding polar coordinates. Gramian matrix is formed by cosine sum operation of polar values [71]. Finally, Gramian matrix will be used to visualize time-series. GASF offers angular values in $[0, \pi]$ range and granularity of data is maintained. Polar coordinates are derived from the normalized time-series values which are radius from the time stamp and cosine of angle in series. Span of the polar coordinate system is adjusted and temporal relations are maintained.

Polar transformation of time-series maintains both values and temporal positions. Therefore, information loss is avoided. GASF transformation retains temporal correlation of raw data. Time-series of length n is encoded as matrix of size $n \times n$. Main diagonal of the matrix contains original content of time-series. Therefore, overall trans-

formation process retains the complete information of time-series. Time-series vector is normalized and then represented in a polar coordinate system. Note that, core of the feature representation technique has Gramian matrix. Gramian matrix consists of elements that are actually cosine of summation of relevant time-series vector angles.

It should be stated that, different feature time-series images may be stacked together to form a three-dimensional image for classification through designed CNN model. Labeling is performed for different categories to be discerned by CNN model [72]. Generated image with GASF transformation depends on input vector which is time-series representation. Output GASF image that has width and height equal to size of original time-series vector. Size reduction of the image may be performed as regarding vast training samples to ease computation load of the model. After passing through a convolution layer, the image becomes abstracted to a feature map [73]. It should be mentioned that, GASF is a mathematical means that combines coordinate transformation and Gram-like matrix to transform time-series to image. As an important point, temporal dependence of time-series is maintained and spatial transformation is performed. Following is the explanation of procedure. Assume that a certain feature sequence including n entries is expressed as time-series X . There are several steps to encode time-series as images. Any time-series sequence can be represented as

$$X = x_1, x_2, \dots, x_n, \quad (3.3)$$

where x_i , $i \in \{1, 2, \dots, n\}$ is the i^{th} element of the vector X . Initially, X time-series is scaled to interval $[-1, 1]$. Benefit of this scaling algorithm is that, original characteristics of features may be retained with small variance. Scaling is combined with inner product to maintain output range. Normalization formula is expressed as

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)}, \quad (3.4)$$

where $\min(X)$ and $\max(X)$ are minimum and maximum values of X .

Scaled sequence is converted to polar coordinates as next step, which is new representation of time-series. Note that, arccos function bijection is used in this study.

Each entry of sequence has related timestamp and polar angle is between $[0, \pi]$. Angle ϕ and radius r are calculated to represent rescaled time-series as polar coordinates that can be computed as

$$\begin{cases} \phi = \arccos(\tilde{x}_i), & -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, & t_i \in \mathbb{N} \end{cases} \quad (3.5)$$

where t_i is the time stamp, and N is a constant factor to regularize span of the polar coordinate system.

Temporal relations of series elements that represent different time intervals may be emphasized for produced image with the implementation of addition between each point of series. Thus, a field of summation is formed for final image conversion. We now define an inner product expressed as

$$a \oplus b = \cos(\theta_1 + \theta_2), \quad (3.6)$$

where θ represents the angle of entries, which produces the Gram-like matrix that can be computed as

$$G = \begin{pmatrix} \cos(\theta_1 + \theta_1) & \cos(\theta_1 + \theta_2) & \cdots & \cos(\theta_1 + \theta_n) \\ \cos(\theta_2 + \theta_1) & \cos(\theta_2 + \theta_2) & \cdots & \cos(\theta_2 + \theta_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\theta_n + \theta_1) & \cos(\theta_n + \theta_2) & \cdots & \cos(\theta_n + \theta_n) \end{pmatrix}. \quad (3.7)$$

Constructed gram-like matrix may be considered as two-dimensional representation of time-series. Note that, Gram matrix is constructed for time-series and elements on diagonal axis retain time dependence. Also, remaining elements provide time-series correlation. GASF transformations of several basic functions are also included to emphasize superior usability of images for time-series studies. This is supposed to help in the understanding of defining features that are not easily grasped.

The sinusoidal function and corresponding GASF transformation is given in Figure 3.6. Since the input function is periodical, resulting GASF image has also repeating patterns. As indicated in the colorbar of GASF image, the high values represent similarities in the time-series, whereas low values indicate low similarity entries of the time-series. Therefore, GASF transformation presents similarity of the time-series.

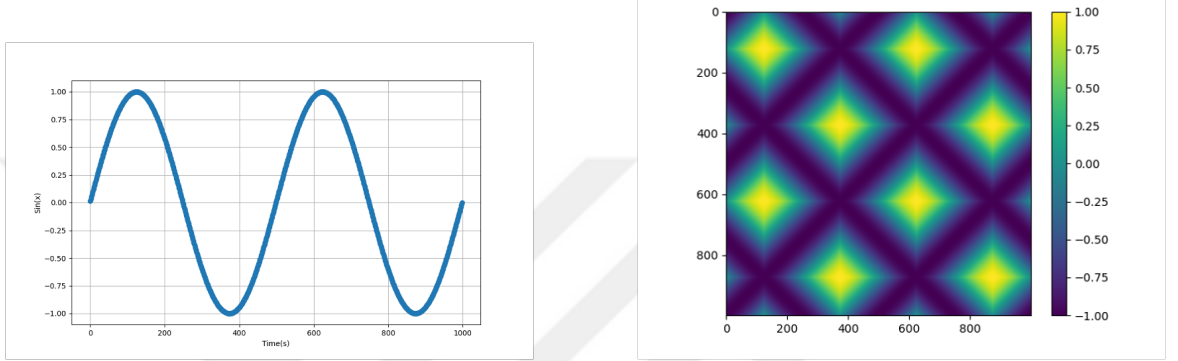


Figure 3.6. Example of a sinusoidal function and related GASF transformation.

The rectangular function and corresponding GASF transformation is given in Figure 3.7. As can be seen from the GASF image, the intersection area in GASF when dissimilar data points of the time-series are taken, is expressed in purple. Also, the intersection area in GASF when similar data points of the time-series are taken, is expressed in yellow. The resulting matrix clearly captures the abrupt transition in the signal, producing distinct quadrants that reflect the sharp change in amplitude.

The sawtooth function and corresponding GASF transformation is given in Figure 3.8. Since the sawtooth function is periodic, resulting GASF image has also repeating patterns that indicate similar and dissimilar areas. The diagonal dominance and repetitive pattern reflect the cyclic nature of the input signal. This transformation provides a structured and visually distinguishable pattern, which can be effectively used by convolutional neural networks for classification tasks.

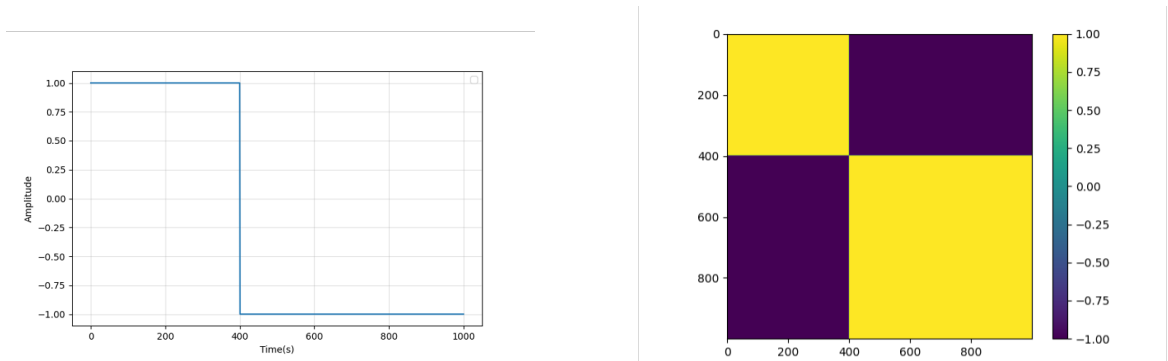


Figure 3.7. Example of a rectangular function and related GASF transformation.

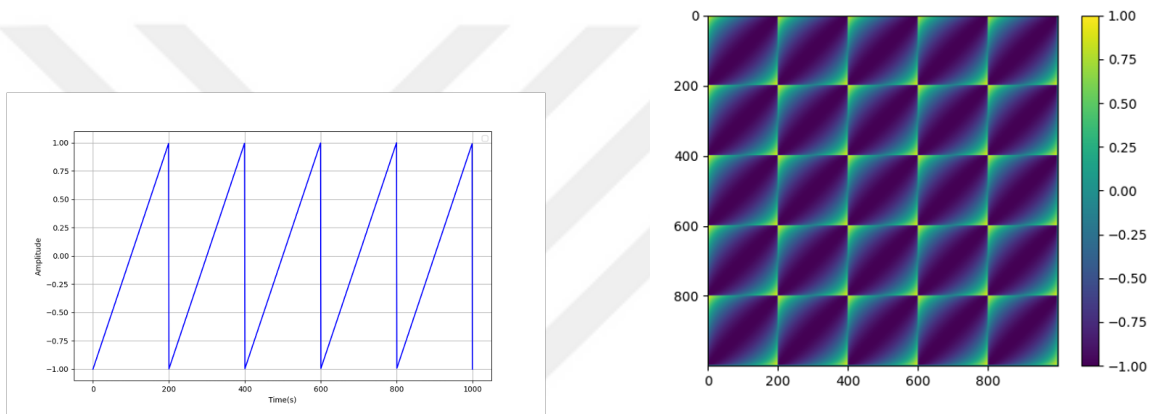


Figure 3.8. Example of a sawtooth function and related GASF transformation.

3.3. Machine Learning Algorithms

Machine learning is a major part of artificial intelligence studies that covers methodologies that provide learning of machines without requiring an explicit programming. Normally, a model is created by feeding a large amount of relevant data as input to certain set of ML algorithms and features are gathered. For network security context, ML approaches may have better ability to adapt to many variations of intrusions or attacks as that are certainly encountered over time. This is the significant strength of ML algorithms. Deep learning is listed as a branch of ML. It is mostly based on artificial neural networks and advanced learning methods that are developed as part of intensive several researches spanning areas. For some suitable cases, better results are achieved than classical ML applications. There are some variants that multiple processors join and collaborate for feature training of specific ML models to

achieve quick results. When these models are compared with classical approaches, several advantages are noted such as classification accuracy. Therefore, novel research efforts are considered to expand model design context to develop better classifiers.

ML models recognize intrusions in several network structures of varying complexities. These approaches may be scalable and are suitable for real-time detection platforms with quick response characteristics. It is very important to reduce false positive indication as part of mitigation efforts. Neural network models may easily make use of historical attack data to refine its working. Parameters may be tuned or model may be retrained with current latest information to enhance accuracy and effectiveness in recognizing and adapting to new attack patterns or schemes. ML is argued to be a very valuable mechanism and provide a context for detecting and mitigating intrusions for communication networks.

Supervised ML approaches have data given as input to mentioned models to train with labels as distinguishing property. Therefore, input sets are mapped to known pre-determined outputs for required classification. The reason is that, method achieves learning process from available dataset with labels and generates proper features to aid classification. An applicable classifier model is carefully designed and formed to use consequently for real-time intrusion detection [74].

3.3.1. Decision Tree

Decision Tree is a well-received algorithm used mainly for classification and regression work. In a tree structure like a flowchart, internal nodes are set to represent defined features. Each branch of tree symbolizes a rule for decision, the leaf symbolizes the outcome as result. Decision tree learns using available training dataset by partitioning data-based feature values [75]. Most proper and informative feature is selected in successive steps to divide data into subsets smaller in size. Therefore, information gain is maximized or the impurity is decreased as the major aim for the subsets.

Decision trees are relatively easy to understand and may capture complex nonlinear relationships up to certain extent of the features. But decision tree may experience overfitting easily and there may be bias in certain models. It is validated in many approaches for recognition of network attacks. With other mentioned ML classifiers, mainly satisfactory results are reached with the relevant implementations.

3.3.2. Random Forest

Random forest is largely accepted learning methodology used in ML for both classification and regression tasks. Concept has roots in accumulation of predictions from several decision trees to generate an accurate outcome. Every decision tree is formed by selecting subset of the training dataset through a procedure named bootstrapping random with replacement. After decision trees are built, predictions are gathered from outcome of individual trees in the model. Random forest model is said to be less sensitive to overfitting and noted as very powerful compared to single decision tree. Random forest model occupies large memory and requires much computation power specifically if quite large datasets are used and model has lots of trees [76]. Several surveys note that, random forest method seems to be most popular and well-suited ML algorithm in studies. Many researches validated its accuracy and combined with different approaches to detect network attacks.

3.3.3. Support Vector Machine

Support vector machine has widespread usage in various areas. Although it is regarded as a powerful method, implementation may be very computation intensive for large datasets. This model is particularly useful and applicable in handling very complex datasets and also easily manage high-dimensional dense feature spaces. The working idea and main process for support vector machine model is to discover a defined plane that separates the data points of various classes in mentioned feature space [77]. This model assumes that points of data may clearly be linearly separated. Support vectors are expressed as data points that have the position closest to defined decision

boundary. Such points have crucial role in making accurate and precise predictions.

SVM has been accepted as major algorithm in many implemented solutions throughout the literature to detect attacks. Quite many researchers structured their solution approach especially on this model. It is argued that support vector machine is better suited for recognition of major intrusions started using multiple nodes compared to other discussed ML classifiers such as Naive Bayes and KNN. Comparison studies regarding the accuracy of several classifiers seems to support this claim largely after validation experiments executed on popular classifiers. Results show that approach generally performed quite accurate, but lower performance figures are also reported in some studies.

3.3.4. k-Nearest Neighbor

KNN is used mainly for classification tasks. This model does not make use of any assumptions about distribution of data. For a new prediction of a data point, this method looks for the K-nearest neighbors to mentioned point in the given feature space. Neighbors are determined with the distance metric used. Majority class among neighbors is set as the predicted label for new data point. For this model, proper value selection for K is crucial [78]. A small K may cause model sensitivity to noise whereas large K is expected to introduce certain bias. Also, it should be noted that the choice of distance metric has effects on how algorithm evaluates similarity of data points.

KNN is accepted as quite easy to understand and relatively straightforward to implement. However, performance is very much dependent on dimensionality appropriate pre-processing and parameter selection is required as very important issue. KNN is researched widely and tested to detect several attacks types and mainly compared to diverse classifiers as a performance metric. Model performance is noted to be fairly acceptable in those approaches for most applications. However, this algorithm may not be completely recommended for wide context of intrusion recognition due to mentioned computation dense property.

3.3.5. Convolutional Neural Networks

Convolutional Neural Networks are deep learning models widely used for image recognition and computer vision tasks which involve grid-like structure for content data. CNNs are also influenced by functioning of human neural structure. Major structure of CNN is designed to automatically learn representations or features of input content through many layers of interconnected nodes. Mentioned layers are expressed as convolutional layers, pooling layers, and fully-connected layers. However, CNN does not include ability to process temporal information. For attack detection context, CNN has been tested and results show various performance figures such that some are acceptable and applicable [79].

3.3.6. Multi-layer Perceptron

Multi-layer Perceptron is a type of artificial neural network that comprise of several layers of artificial neurons or so-called perceptions. It is considered as the simplest and widely used neural network structures. Each layer contains several neurons that are connected with neurons in near layers. This model may learn difficult non-linear relationships and is mainly employed for non-sequential data applications [80]. But model does not have the ability to capture temporal dependencies. Multi-layer Perceptron is validated to handle several network intrusions. It is determined that the model has average recognition performance and success figures if compared to other ML classifiers. It should be noted that these classifiers have superior classification time performance figures like decision tree as major strength. Model outperforms classical approaches with respect to this aspect. Results of the model performance for adding historical context into the learning process is not promising and overall success figures of the model is mixed indicating requirement of further research.

3.4. Proposed Methods

3.4.1. Proposed Matrix Profile Based Random Forest Model

Matrix profile is used in integration with various machine learning methods. These machine learning methods include Random Forest, Support Vector Machine, Multi-Layer Perceptron, KNN and XGBoost. The aim of using CIC IoT 2023 data set is to determine the attack category and attack type. A time-series based approach is proposed for detecting and classifying network traffic attacks. The incoming network traffic is divided into predefined window lengths for analysis, and within each window, an attempt is made to determine whether an attack is present and, if so, its type. In this process, statistical and information theory based features are extracted from the packets in each window. Statistical features are computed using inter-packet arrival times and packet sizes, while information theory based features help reveal relationships between packet header information. Subsequently, matrix profile is applied to these features, which are structured in a time-series format. Matrix profile is performed on features using a predefined window size of 100 and a sliding window size of 20. In this way, features become more distinctive compared to their direct use. The matrix profile is utilized to quickly identify similarities in the time series, detect anomalies, and find recurring patterns, thereby enhancing the discriminative power of the features. Finally, the matrix profile-applied features are fed into a machine learning-based classification algorithm to perform attack classification. This method improves the effectiveness of the attack detection process by extracting meaningful patterns from time-series data. The block diagram of matrix profile based machine learning model is given in Figure 3.9. It illustrates the proposed matrix profile-based machine learning architecture for detecting various network attacks using the CIC IoT 2023 dataset. The model extracts 29 time-series features, applies matrix profile transformation to capture temporal patterns, and performs classification with Random Forest across seven attack classes.

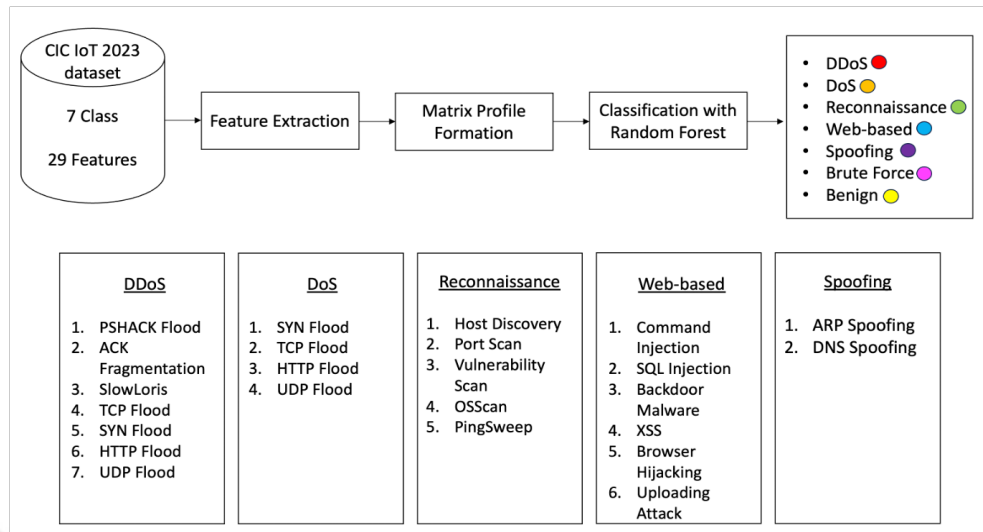


Figure 3.9. Block diagram of matrix profile based machine learning model.

3.4.2. Proposed GASF Based Convolutional Neural Network Model

CNN models already revealed effectiveness especially in anomaly detection tasks when applied to converted sequence data. Neural networks differentiate minor patterns with particular ease and help to discern complex features, thus overall success improved for classification. In this study, dataset as the source is preprocessed to construct features and sequence data is converted to image. Therefore, a preprocessing layer is added to improve performance figures. Valuable features that are known to contain important characteristics are determined for selection and superior performance figures are ensured. Constructed data in this way is used for validation and tuning of implemented model.

Note that, CNN is widely used by researchers especially after transforming raw data to image for further processing. It is widely known to be successful in learning tasks that include images as inputs. Therefore, CNN may be successfully utilized for anomaly or attack detection basically following image production. CNN learns relationships from image that is generated for anomaly detection. This study evaluates the model for network attack detection success.

CNN model has significant benefits for processing images. Therefore, many studies suggested converting raw data to images as input to CNN model. Classifier then achieves required categorization as outcome. Implemented CNN model structure gathers local features by several convolution layers and stores attributes as feature maps. Local feature capture capability of neural network greatly contributes to successful achievement. Therefore, selection of GASF as time-series conversion method is definitely justified as results clearly show.

Several researchers who apply machine learning methodologies for anomaly detection area frequently struggle to meet the requirements of scalability and real-time performance for complete usability of approaches. Researchers experiment on several transformation schemes to extract hidden or complex information from images while retaining temporal correlation and sufficient feature content which is clearly very important point to consider. Therefore, various approaches contain a certain raw data to image transformation methodology with CNN models to satisfy such constraints.

It is stated clearly that, CNN structure is recognized for its effectiveness in accurately defining relationships in image recognition. Since time-series may be expressed as image via the conversion, network attack detection problem may be quickly viewed as an image recognition and classification task with available powerful toolset of algorithms. Proposed framework discovers and makes accessible feature traits that were not readily used thus allows for further performance enhancement in detection. Also, traditional binary classification approach is enhanced as multi-label attack classification security solution. CNN when implemented as multi-label classifier, leads to network attack detection and classification simultaneously that may help to develop very valuable security solution. Success of methodology builds on effective formulation of extracted time-series for multi-label classification problem. This thesis presents the effectiveness of GASF time-series to image transformation algorithm for network attack detection when combined with CNN machine learning model. Model differentiates various forms of attacks as the outcome. GASF effectively retains existing properties of feature content and CNN achieves classification of types.

CNN is considered successful in solving several recognition problems such as image classification. It should be stated that, CNN is basically a neural network model and main structure consists of stacking convolutional and pooling layers. Note that, complex CNNs also have a similar structure with different number of convolution layers and pooling layers for feature extraction enhancements. Input data is first fed to combination of convolution and pooling layers before sent to layer that mainly performs classification task that produces the outcome. Each convolution layer performs mathematical convolution operation on inputs and has activation threshold similar to actual neural structure. These layers have defined structure that feature spaces inside filter or extract features to be processed by following pooling layer. Pooling layers of neural network are normally used to reduce dimension size of convolution layer outputs. This is achieved such that, values are evaluated together and either minimum, maximum or average is used to reach a defining number. Pooling layer output is sent by rectified linear unit to next convolution layer in the network. This scheme continues until reaching fully-connected layer.

Note that, designing proper and effective neural networks requires a lot of carefully selected parameters, such as topology details and threshold values. It should also be noted that learning time may be very long to implement large scale training samples. Computing power and memory consumption may be considered as expensive or not feasible. Although such facts may be seen as limitation, final outcome and use cases may compensate for such difficulties. Therefore, neural network classifiers are clearly an option for further study.

Traditional CNN models are known to be very effective in extracting local attributes through convolution operations and capturing features at several such levels. Therefore, proposed implementation has feature maps generated via stacked convolutional layers. Architecture consists of convolution and pooling layers with different kernel sizes. Convolution layers involve size (3,3) with stride of 1 followed by max-pooling layer with size (2,2) and stride of 2. Convolution layer has activation function implemented. Therefore, the model aggregates feature content of the entire image

with subsequent processing layers. Finally, classification result is reached through fully-connected layer of neural network. Therefore, extracted features reach to end of CNN to produce the result. Proposed multi-label CNN-based classifier is trained using features as images and produce classifications.

CNN model takes GASF images as input. Presented CNN model consists of convolution layers, pooling layers, and fully-connected layer. Fully-connected layer at the end of model reveals the outcome and result. Structure of the neural network model may be expressed as follows:

(i) Convolutional Layer

Each image consists of pixels. Considering a time series of size t , using sliding windows of size m will produce a matrix of size $m \times m$. If there exist k features, concatenating them will produce a 3D matrix of size $m \times m \times k$. The depth of this matrix depends on the feature size. CNN utilizes kernels or also known as filters to extract important features from an image. Filters perform convolution operation on the image. The extracted features may be textures, edges etc. This operation reduces the dimension of the image without losing its characteristics by extracting meaningful features. After convolutional layer, feature maps are formed. There are some parameters used in convolutional layers which are stride and padding. Stride is defined as the amount of steps the kernel moves in each step of the convolution. Padding operation is performed on the input image in order to maintain the original size of the image after convolution.

In our proposed CNN model, first convolution layer consist of n filters and kernel size of 3×3 . Note that the channel size of the input model should match the depth of the filters since each filter performs convolution operation on each channel of the input. Therefore, the kernel size can also be represented as $3 \times 3 \times k$. Padding is used in order to maintain the original size of the input. The output of the convolutional layer which is defined as the feature map will be of size $m \times m \times n$. Convolutional layers of proposed CNN model is given in Figure 3.10. The size of the feature map can be calculated as

$$\frac{m + 2 \times p - k}{s} + 1, \quad (3.8)$$

where m denotes the input size, p denotes the padding size, k denotes the kernel size and s denotes the stride size. In the first convolutional layer padding is applied to the input and 64 filters and a kernel size as 3×3 is applied. The second convolutional layer uses 128 filters and kernel size as 3×3 . Padding is used. In our model, there exist 3 convolutional layers. In the final convolutional layer, 256 filters and kernel sizes of 3×3 are implemented. Similar to other convolutional layers, padding is applied.

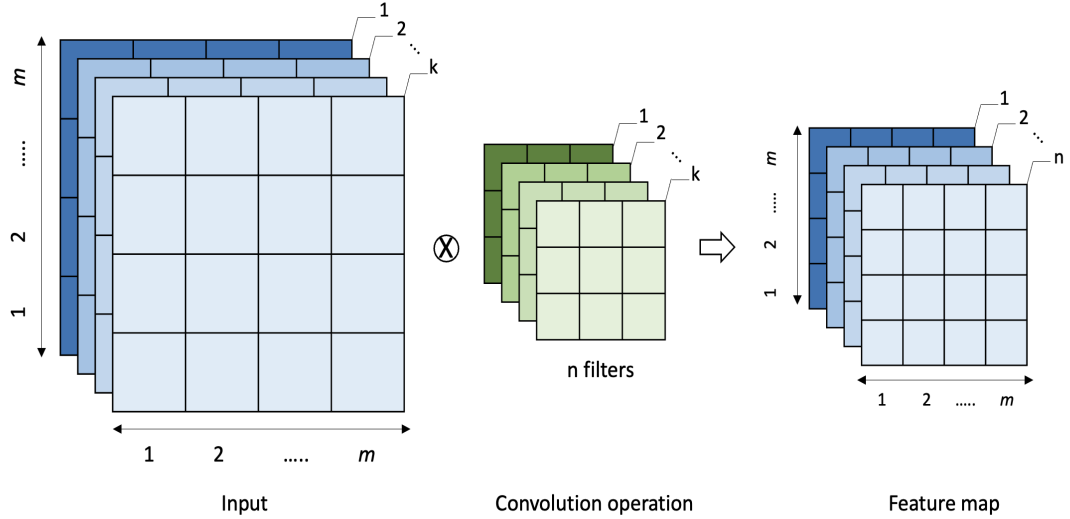


Figure 3.10. Convolutional layers of proposed CNN model.

After each convolutional layer, ReLU activation function is implemented. Activation functions enable the model to learn the nonlinearities. Several activation functions may be used in a CNN model, which are given in Table 3.1.

Table 3.1. Activation Functions.

Activation Function	Formula	Range
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$(0, 1)$
Tanh	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$
ReLU	$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$[0, \infty)$
Softmax	$x_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}, (i = 1, 2, \dots, N)$	$(0, 1)$
Unit Step Function	$u(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$\{0, 1\}$

(ii) Pooling Layer

Pooling layers are used to reduce the size of the feature maps. There are several types of pooling, the most commonly used ones are maximum pooling and average pooling. In our proposed CNN model, maximum pooling with a size of 2×2 is used. Stride size is also defined as 2. Maximum pooling is implemented after each convolutional layer. Pooling layers of proposed CNN model can be seen in Figure 3.11. This layer achieves max-pooling to further down sample local feature maps formed in previous convolution layer. The maximum element in the area where the pooling matrix is located is taken and written to the pooled feature map. In average pooling, the average of the values in this 2×2 matrix is taken. Note that pooling layers do not reduce the number of channel size. Pooling layer is applied to each channel of the feature map separately. The size of the output image with input size m , pooling kernel size k and stride s can be calculated as

$$\frac{m - k}{s} + 1. \quad (3.9)$$

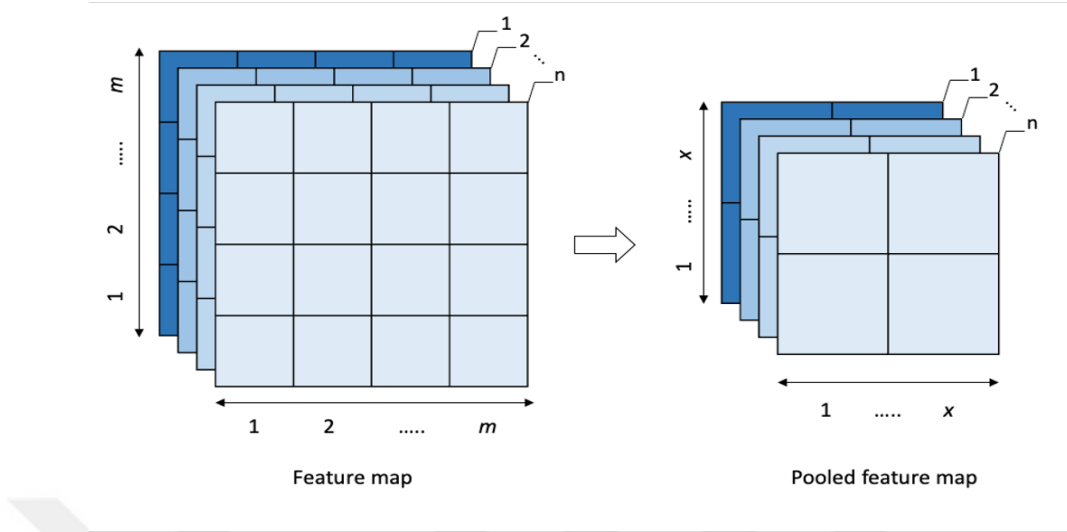


Figure 3.11. Pooling layers of proposed CNN model.

(iii) Flatten Layer

Flatten layer converts feature maps to vector prior to fully-connected layer. The flattened matrix as feature vector moves through fully-connected layer and classification of image is thus achieved.

Proposed model readily performs classification of packet traffic, recognition proceeds with features that are extracted as time-series and thus converted to image for neural network processing. Selection of image-based approach for network attack detection is mainly encouraged considering frequently mentioned impressive success of machine learning algorithms in computer vision area studies. Note that, data set selection is also an important aspect of the study to train mentioned model and utilized dataset provides a wide range of network attacks to validate performance thoroughly. Therefore, attack classification may be accomplished with great success rates. Block diagram of proposed CNN model is given in Figure 3.12. For the procedure of CNN, time-series features gathered from data sets are converted to images with GASF method. Produced data moves through CNN model layers. Convolutional layers successively extract local features and related feature map is formed. Feature sequence is represented with a certain category symbol. Several attack categories are considered. Fully-connected layer of model produces required attack category as the final result.

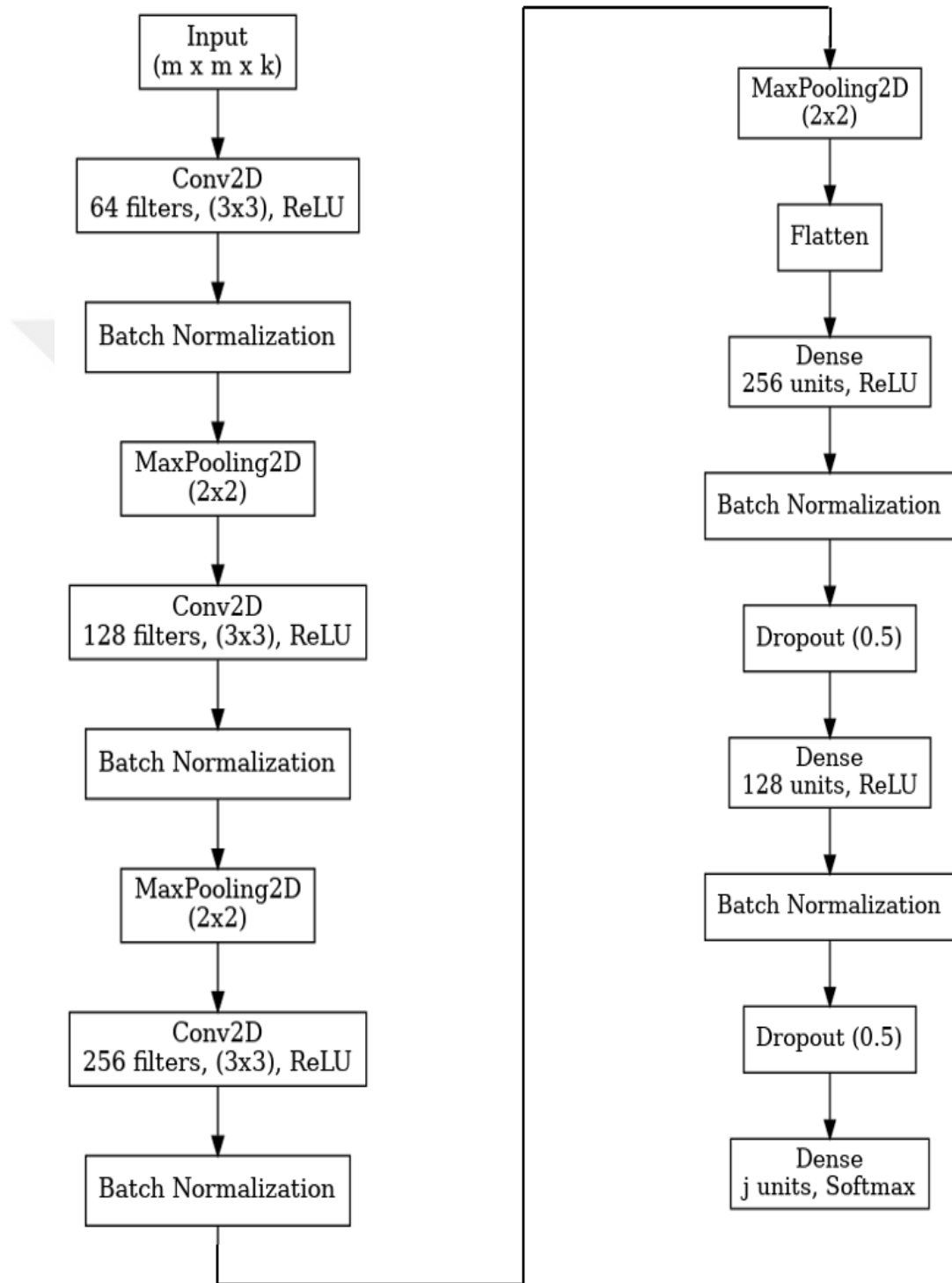


Figure 3.12. Block diagram of proposed CNN Model.

4. FEATURE EXTRACTION METHODS FOR NETWORK INTRUSION AND APPLICATION CLASSIFICATION

4.1. Statistical Features

4.1.1. Inter-arrival Time

Inter-arrival time between packets is an essential parameter commonly used in network traffic classification methods. Using these parameters, it is possible to model essential characteristics of network traffic such as traffic speed. The inter-arrival time is defined as the time difference between the arrival of consecutive packets. The inter-arrival time vector of a window consisting of N packets can be calculated as

$$\Delta = [\delta_1, \delta_2, \dots, \delta_{N-1}], \quad (4.1)$$

where the inter-arrival time vector Δ has size $N - 1$. Each element of this vector is calculated as

$$\delta_i = T_{i+1} - T_i, \quad \forall i \in \{1, 2, \dots, N - 1\}, \quad (4.2)$$

where T_i is the time stamp when the i^{th} packet comes. Similarly, T_{i+1} is the time stamp when the $(i+1)^{th}$ packet comes. δ_i represents the time difference between these two consecutive packets. The features are extracted by computing the minimum, maximum, mean, standard deviation, first, second, and third quartiles, skewness, kurtosis, and median absolute deviation statistics of the inter-arrival time vector. These features are named as iat_{min} , iat_{max} , iat_{mean} , iat_{std} , iat_{q1} , iat_{q2} , iat_{q3} , iat_{mad} , iat_{skew} and iat_{kurt} .

4.1.2. Packet Size

The amount of data transferred within a window of packets is also another essential parameter in order to classify the network traffic. This parameter enables to model the amount of data transmitted per unit time. Packet sizes within a window

are considered. For a window containing N packets, the packet size vector can be expressed as

$$P = [p_1, p_2, \dots, p_N], \quad (4.3)$$

where p_j , $j \in \{1, 2, \dots, N\}$ represents the size of the j -th packet within the window, and P denotes the packet size vector. Statistical features such as minimum, maximum, mean, standard deviation, first, second, and third quartiles, skewness, kurtosis, median absolute deviation, bytes per second, and total number of bytes are extracted from the packet size vector. These features are called as $byte_{min}$, $byte_{max}$, $byte_{mean}$, $byte_{std}$, $byte_{q1}$, $byte_{q2}$, $byte_{q3}$, $byte_{mad}$, $byte_{skew}$, $byte_{kurt}$, $byte_{sec}$ and $byte_{total}$.

4.2. Information Theory Based Features

In addition to statistical features, information theory-based features have also been used to model network traffic behavior. The most important of these features is entropy, which measures the disorder in the system. Low entropy indicates regularity, while high entropy indicates instability. Information theory-based features are calculated using packet header information. For example, in DDoS attacks, when heavy traffic is sent from different source IP addresses to the target IP, the disorder in the source IPs increases while it decreases in the target IPs. Entropy is used to measure this change. To use the entropy value, for a given time window, the network packets are gathered. Then, for each packet header value, source and destination IP addresses, source and destination port numbers, the probability distribution of the header value is constructed. It is constructed according to the number of packets they are in. The size of the probability distribution is equal to the unique number of the selected packet header value given a time window. As in the DDoS example, the entropy of source IP addresses increases because multiple source IP addresses are used to execute the attack. On the other hand, the entropy of destination IP addresses decreases because most of the packets belong to the victim IP address. The entropy values of those packet header values are used to model the network traffic behavior. Another information theory metric used in feature calculation is the greedy distance [81]. In order to determine

the distances between packet header information, the probability distribution of packet header information within each window is constructed according to the connections and the distance between these distributions is calculated by the greedy method. For example, in DDoS attacks, the distance between the source and destination IP addresses is expected to increase at the time of the attack. The distance between packet headers is important in modeling network traffic behavior and is used to detect different types of attacks. While the entropy values of the source and destination IP addresses and connection points are added to the feature set, the greedy distances between the binary combinations of these header information are used as a total of six additional features. The description of the information theory-based features constructed using entropy concept and greedy algorithm is given in Table 4.1.

Table 4.1. Information theory-based Features.

Features	Description
ent_{SIP}	Entropy of source IP address
ent_{DIP}	Entropy of destination IP address
ent_{SP}	Entropy of source port
ent_{DP}	Entropy of destination port
$\text{greedy}_{\text{SIP-DIP}}$	Greedy distance between source IP and destination IP address
$\text{greedy}_{\text{SIP-DP}}$	Greedy distance between source IP and destination port
$\text{greedy}_{\text{SIP-SP}}$	Greedy distance between source IP and source port
$\text{greedy}_{\text{SP-DP}}$	Greedy distance between source and destination port
$\text{greedy}_{\text{SP-DIP}}$	Greedy distance between source port and destination IP
$\text{greedy}_{\text{DIP-DP}}$	Greedy distance between destination IP and destination port

5. PERFORMANCE EVALUATION OF THE PROPOSED METHOD

In this chapter, the simulation results of the proposed time series-based approaches for both network intrusion detection and application classification are presented. Two distinct methodologies have been used: the first integrates Matrix Profile with the Random Forest algorithm, while the second employs Gramian Angular Summation Field (GASF) with a Convolutional Neural Network (CNN) model. The CIC IoT 2023 data set is utilized for the evaluation of intrusion detection performance, whereas the Encrypted Network Traffic Classification data set is used for application classification. Accordingly, the results are categorized into four parts:

- Intrusion detection using Matrix Profile and Random Forest.
- Application classification using Matrix Profile and Random Forest.
- Intrusion detection using GASF and CNN.
- Application classification using GASF and CNN.

All of the proposed solutions are developed based on time-series techniques, and a comprehensive analysis of their performance is provided in the following sections.

5.1. Data Sets

In this section, the data sets used to evaluate the proposed methods are introduced. Three data sets were utilized: Boğaziçi University DDoS Attack data set, Encrypted Network Traffic Classification data set and the CIC (Canadian Institute for Cybersecurity) IoT 2023 data set. Both Boğaziçi University DDoS Attack data set and CIC IoT 2023 data sets include various types of Distributed Denial-of-Service (DDoS) attacks with different traffic characteristics and attack intensities. Moreover, the CIC IoT 2023 data set also incorporates other types of network threats such as DoS, brute force, spoofing, reconnaissance, and web-based attacks, providing a comprehen-

sive environment for evaluating intrusion detection systems. Encrypted Network Traffic Classification data set includes different applications such as Discord, GoToMeeting, Instagram, Line, Microsoft Teams, Snapchat, Tango, Slack and TikTok. All details regarding these data sets are presented in the following subsections.

5.1.1. Boğaziçi University DDoS Attack Data Set

Boğaziçi University DDoS Attack Data Set is generated with a traffic generator software in Boğaziçi University [82]. Resource depleting DDoS attacks on university network are formed. Generator produces many TCP SYN, and UDP packets. Data set also includes attack-free traffic. Therefore, many types of traffic are included. Traffic is recorded in backbone router with around 2000 active hosts. Industry standard Wireshark software achieved the traffic recording. It may be used to evaluate DDoS attack detection methodologies. As the scheme, a victim server is affected by the network attacks. Generated attack packets have spoofed source IP addresses. Note that, recording mechanism gathers exact copies of incoming and outgoing packets flowing through related router interfaces. Network traffic rate reaches 1800 packets per second. Implemented attack scenarios include TCP SYN and UDP flood. Data set has 8 minutes of network traffic collected. For both TCP SYN Flood and UDP Flood, DDoS attack is executed four times. DDoS attacks have several packet rates. Note that, TCP SYN and UDP floods include 1000, 1500, 2000 and 2500 packets per second rate characteristics. Data is recorded as 80 s waiting and 20 s attack period intervals. Traffic packets are saved as well-known pcap file format and also converted to csv file format. Therefore, data set may be used for research community for network attack detection studies.

5.1.2. CIC IoT 2023 Data Set

Main goal of CIC IoT 2023 data set is to provide a comprehensive IoT attack information for research efforts. In the data set, a total of 25 different types of attacks are divided into six categories. These are DDoS, DoS, Recon, Web-based, Brute force,

and Spoofing. Attack classes and attack types in this data set are shown in Table 5.1. A certain network device in the presented topology of the data set stores all traffic and using industry standard Wireshark tool. In the mentioned network, attacks are performed by 105 devices. Data set contains attacks executed by malicious IoT devices that target other IoT devices. Each attack examined in this data set has different characteristics. For example, traffic generated by DDoS attack is larger than the traffic generated by a spoofing attack [83].

Table 5.1. Attack Classes with their Subclasses of CIC IoT 2023 Data Set.

Attack Class	Subclasses
Brute force attacks	Dictionary brute force
DDoS attacks	HTTP Flood, PSHACK Flood, Slowloris, SYN Flood, TCP Flood, UDP Flood, ACK Fragmentation
DoS attacks	HTTP Flood, SYN Flood, TCP Flood, UDP Flood
Reconnaissance attacks	OS Scan, Ping Sweep, Port Scan, Vulnerability Scan, Host Discovery
Web-based attacks	Browser Hijacking, Command Injection, SQL Injection, Uploading Attack, XSS, Backdoor Malware
Spoofing attacks	DNS Spoofing, ARP Spoofing

CIC IoT 2023 dataset may be utilized in Packet Capture (PCAP) and Comma Separated Values (CSV) file formats. PCAP files contain original data collected in the mentioned network for different attack scenarios. All packets are included and may be used to extract or engineer other features to examine. Note that, CSV files present easier way of using mentioned data. CSV files contain features extracted from original PCAP files and they are presented by fixed packet size window. Features are extracted from packets between two different hosts. They are determined by assessing several literature studies regarding security. Therefore, major aim of dataset is to implement an approach to evaluate machine learning models with several features. Also, many features may be extracted with scripts using raw network traffic. CIC IoT

2023 data set extends existing security concept by using an advanced topology with several devices. Therefore, the data set clearly allows to examine how novel machine learning methodologies perform in various classification tasks.

5.1.3. Encrypted Network Traffic Classification Data Set

The encrypted network traffic classification data set includes network traces from real world applications collected in Boğaziçi University. This data set includes different network traffic categories with different applications. Some applications have network traces in multiple categories. For example, Discord application has network traces in all categories which are Audio Chat, File Transfer, Text Chat and Video Chat. The applications for each category are given in Table 5.2. In the simulations, this data set is used for category detection and application identification. The sample size of each category within each application is shared in the simulation results. The category classes and related applications in this category are given in Table 5.2.

Table 5.2. Category and Applications of Encrypted Network Traffic Classification Data Set.

Category	Applications
Audio Chat	Discord, GoToMeeting, Instagram, Line, Microsoft Teams, Snapchat, Tango
File Transfer	Discord, Line, Microsoft Teams, Slack
Text Chat	Discord, Instagram, Snapchat, Tango, Tiktok, Microsoft Teams, Slack, Line
Video chat	Discord, GoToMeeting, Instagram, Line, Microsoft Teams, Snapchat, Tango

5.2. Performance Metrics

Evaluation of the performance of the model is achieved with widely used performance metrics. These metrics provide a thorough understanding of how proposed model successfully categorize input sequence. Different metrics including accuracy, precision, recall etc. are calculated for evaluation of detection success of the method. These metrics are calculated using a confusion matrix of size 2×2 for binary classification which is given in Table 5.3.

Table 5.3. Confusion matrix for 2x2 binary classification case.

	Actual: Positive	Actual: Negative
Prediction: Positive	True Positive	False Negative
Prediction: Negative	False Positive	True Negative

True Positive corresponds to attack samples categorized as attack. True Negative samples are normal samples categorized as normal. Also, False Positive corresponds to normal samples falsely categorized as attack. False Negative are attack samples falsely categorized as normal. Mentioned performance metrics are calculated as follows:

- False Positive Rate (FPR) = $\frac{FP}{TN+FP}$,
- False Negative Rate (FNR) = $\frac{FN}{TP+FN}$,
- True Positive Rate (TPR) = $\frac{TP}{TP+FN}$,
- True Negative Rate (TNR) = $\frac{TN}{TN+FP}$.

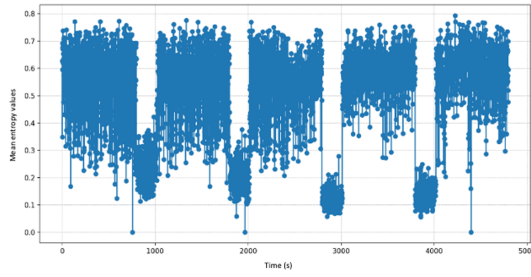
A robust detection system is expected to achieve high True Positive Rate (TPR) and True Negative Rate (TNR) values, demonstrating successful classification of both attack and normal instances. At the same time, it should have low False Positive Rate (FPR) and False Negative Rate (FNR) since it indicates the model makes fewer misclassifications. However, relying solely on these four metrics may not fully capture the performance characteristics of the model. Therefore, additional evaluation criteria such as accuracy, precision, recall, specificity and F1-score are also considered.

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$,
- Precision = $\frac{TP}{TP+FP}$,
- Recall = $\frac{TP}{TP+FN}$
- Specificity = $\frac{TN}{TN+FP}$
- F_1 -score = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

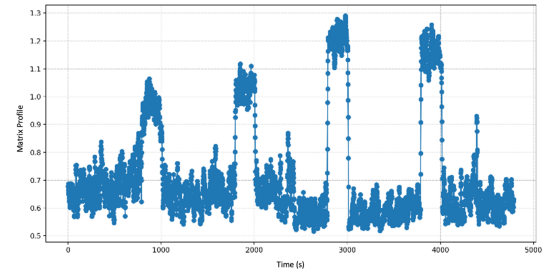
Accuracy shows the overall rate of correct predictions made by the model. Precision indicates how many of the predicted positive samples are actually correct. Recall measures how well the model detects all actual positive cases. F1-score is the harmonic mean of precision and recall, and it becomes especially useful when the dataset is imbalanced. These metrics provide a clear understanding of how well the model performs in different situations.

5.3. Matrix Profile Based Random Forest Model Results

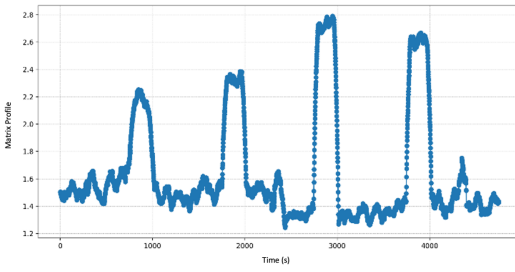
Matrix profile algorithm is implemented using Boğaziçi University DDoS TCP attack data set given in [82]. A time-series from this dataset can be seen in Figure 5.1a. Mean entropy, skewness and kurtosis features are extracted from the dataset. Four attacks may clearly be observed in the time-series. The Euclidean distance of this time-series with itself is calculated using sliding windows of varying sizes. Sliding window sizes are determined as 10, 50 and 100 which may be seen in Figure 5.1b, 5.1c and 5.1d respectively. Distance matrix and matrix profile are calculated. As clearly seen in Figure 5.1d, the matrix profile values are high in time intervals that are considered as abnormal in time-series and as a result 4 attacks are recognized with matrix profile. Receiver Operating Characteristic (ROC) curve is used to evaluate the performance of this method. The ROC curve plots the true positive rate against the false positive rate, and area under the curve (AUC) indicates the discrimination ability of the model. Note that, the higher the AUC value, the better the performance of the model. The ROC curve is given in Figure 5.2. As an important point, the optimal threshold value on the ROC curve is usually the point that maximizes the TPR and minimizes FPR.



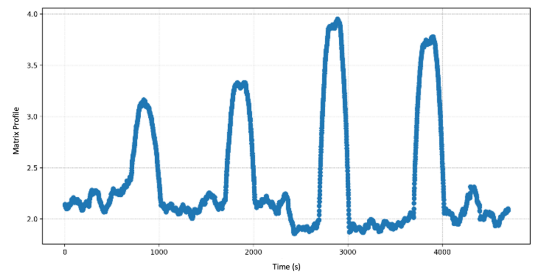
(a) Time-series of mean entropy values.



(b) MP with sliding window size 10.



(c) MP with sliding window size 50.



(d) MP with sliding window size 100.

Figure 5.1. Matrix profile results of mean entropy values with different sliding window sizes.

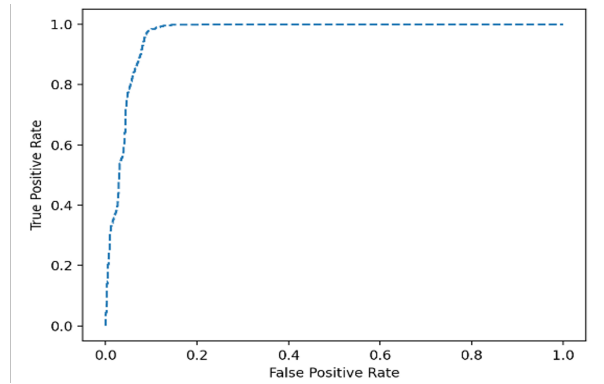


Figure 5.2. ROC Curve on mean entropy values of Boğaziçi University DDoS attack data set.

The optimal threshold value, precision and recall values can be seen in Table 5.4. It can be seen that the optimal threshold value is 2.44, the precision is calculated as %91.2 and the recall value is %97.9.

Table 5.4. Performance Metrics of Matrix Profile Method using Boğaziçi University DDoS Attack Data Set.

Optimal threshold	2.44
Precision	91.2%
Recall	97.9%

In matrix profile based approach, the incoming network traffic is analyzed using a window size of 1 second. From all packets gathered in this 1 second period, 35 features are extracted. The features are structured in a time-series format. Each time-series that represents a feature is divided into a window of size 100 and matrix profile is calculated. Sliding window size is set as 20. Therefore, matrix profile of length 81 is calculated for each window. After applying matrix profile to each window in the time-series, matrix profile values are concatenated. This process is applied to every feature. As a result, a time-series including matrix profile results are obtained. This approach is applied for both intrusion detection and application identification results.

5.3.1. Intrusion Detection

The intrusion detection method is tested with CIC IoT 2023 data set given in [83]. Attack class and attack type classification is performed using matrix profile and random forest algorithm. Classification of attack classes means classifying 7 classes which are DDoS, DoS, Reconnaissance, Spoofing, Web-based, Brute Force and Benign. Whereas attack type classification means, classifying DDoS classes based on whether they belong to ACK Fragmentation, HTTP Flood, PSHACK Flood, SlowLoris, SYN Flood, TCP Flood or UDP Flood and similarly for DoS, Reconnaissance, Spoofing and Web-based classes. Random forest is applied with 30 estimators and maximum depth of 20. %70 of the data is used in training and %30 of the data is used in testing. The confusion matrix for the classification of attack classes is shown in Figure 5.3. In the confusion matrix horizontal axis indicates the predicted class and vertical axis indicates the true class. Diagonal entries in the confusion matrix represent true predictions. It can be seen that all attacks are classified with an accuracy above %90. However, %7.8 of web-

based attacks are misclassified as Reconnaissance which is the most misclassified attack class among them. Performance table is given in Table 5.5. The overall classification performance across all seven attack types and the benign class is remarkably high, with a macro average F1-score of %98.6 and weighted average F1-score of %99.4, indicating that the model performs consistently well even across imbalanced class distributions. The Brute Force and Web-based attack classes exhibit relatively lower recall scores suggesting that a small portion of these instances are misclassified. On the other hand, DoS, DDoS, and Spoofing attacks are classified with near-perfect scores. It can be seen that there are less instances in spoofing, web-based and brute force. The macro-average F1 score is %98.6 meaning that model performs well among all class instances, even for imbalanced data.

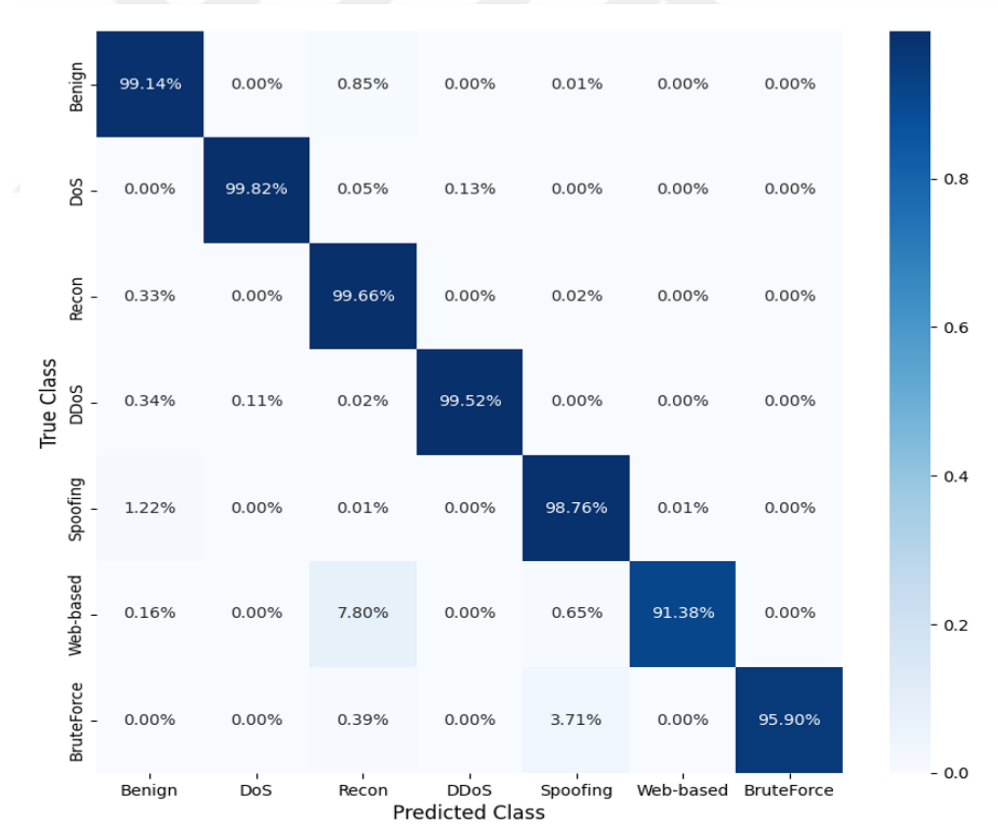


Figure 5.3. Confusion matrix of attack classes using matrix profile based random forest method with CIC IoT 2023 data set.

Table 5.5. Performance Table of Attack Classes using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

Class	Precision	Recall	F1-Score	Support
Benign	0.988	0.991	0.990	24442
DoS	0.998	0.998	0.998	22284
Recon	0.986	0.997	0.991	23070
DDoS	0.999	0.995	0.997	36211
Spoofing	0.996	0.988	0.992	7646
Web-based	0.999	0.914	0.955	1230
Brute Force	1.000	0.959	0.979	512
Macro Avg	0.995	0.977	0.986	115395
Weighted Avg	0.994	0.994	0.994	115395

Attack type classification is performed for 5 classes which are DDoS, DoS, Reconnaissance, Web-based and Spoofing. For every attack type classification, 30 estimators and maximum depth of 10 is set as the random forest parameters. Performance table for DDoS class is given in Table 5.6. The model is highly accurate, overall accuracy of the model is %96.99. In particular, the classifier demonstrates exceptional performance in detecting ACK Fragmentation, HTTP Flood, and UDP Flood attacks, all achieving near-perfect or perfect scores in precision, recall, and F1-score. Moreover, the macro and weighted averages of precision, recall, and F1-score are consistently above 0.97, confirming that the model generalizes well across different DDoS attacks.

Table 5.6. Performance Table of DDoS Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
SYN Flood	0.935	0.934	0.934	6756
PSHACK Flood	0.986	0.943	0.964	5725
ACK Fragmentation	1.000	1.000	1.000	5264
TCP Flood	0.925	0.957	0.941	6880
UDP Flood	0.999	0.998	0.999	8236
HTTP Flood	1.000	0.999	1.000	1362
SlowLoris	0.983	1.000	0.991	1975
Macro Avg	0.975	0.976	0.976	36198
Weighted Avg	0.970	0.970	0.970	36198

Performance table for DoS class is given in Table 5.7. The model correctly classifies %98.60 of all test samples which is a good performance. Macro average of the F1-score is %98.7. The model maintains high consistency across all DoS types.

Table 5.7. Performance Table of DoS Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
SYN Flood	0.972	0.977	0.975	6121
TCP Flood	0.978	0.977	0.978	6214
UDP Flood	1.000	0.996	0.998	8202
HTTP Flood	0.999	1.000	1.000	1781
Macro Avg	0.987	0.988	0.987	22318
Weighted Avg	0.986	0.986	0.986	22318

Performance table for Reconnaissance class can be seen in Table 5.8. The model correctly classifies %99.51 of all Reconnaissance instances. Macro average of the F1-score is %98.6.

Table 5.8. Performance Table of Reconnaissance Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
OSScan	0.985	0.969	0.977	1946
VulnerabilityScan	0.998	1.000	0.999	14236
PortScan	0.970	0.975	0.972	1791
HostDiscovery	1.000	1.000	1.000	4927
PingSweep	1.000	0.963	0.981	54
Macro Avg	0.991	0.981	0.986	22954
Weighted Avg	0.995	0.995	0.995	22954

Performance table for Spoofing class can be seen in Table 5.9. The overall accuracy of the model is %98.73. The classifier maintains high consistency across both attack types considering the macro average of the F1-score.

Table 5.9. Performance Table of Spoofing Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
DNS Spoofing	0.993	0.983	0.988	4093
ARP Spoofing	0.981	0.992	0.987	3577
Macro Avg	0.987	0.988	0.987	7670
Weighted Avg	0.987	0.987	0.987	7670

Performance table for web-based class can be seen in Table 5.10. The model correctly classifies %97.65 of all instances, demonstrating strong reliability. Also, high macro average score which is %97.9 indicates that the model performs consistently across all web-based attack categories.

Table 5.10. Performance Table of Web-based Class using Matrix Profile based Random Forest method with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
Browser Hijacking	1.000	0.996	0.998	226
SQL Injection	1.000	1.000	1.000	233
Command Injection	0.923	0.969	0.946	260
XSS	0.963	0.955	0.959	243
Backdoor Malware	1.000	0.961	0.980	231
Uploading Attack	1.000	0.988	0.994	86
Macro Avg	0.981	0.978	0.979	1279
Weighted Avg	0.977	0.977	0.977	1279

The impact of the matrix profile method on classification performance was evaluated by also performing attack class and type detection by using features directly. The comparative results obtained for attack class detection are presented in Figure 5.4. These results demonstrate that the matrix profile improves performance in every class, with the most notable improvements observed in the Web-Based, Spoofing, Brute Force, and Benign classes. This method has enabled the selected feature set to distinguish different attack types more effectively.

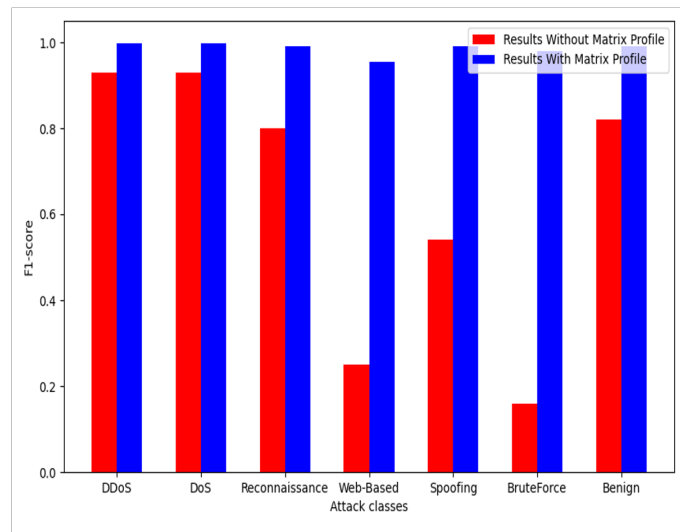


Figure 5.4. Effect of matrix profile method on F1-score for attack class.

A comparison was also made between the results obtained with and without matrix profile method for attack type detection. The weighted average F1-score was chosen as the comparison metric. The results are shown in Figure 5.5. It was observed that the matrix profile method also improved performance in attack type detection. The most significant improvement was observed in the Web-Based and Spoofing class.

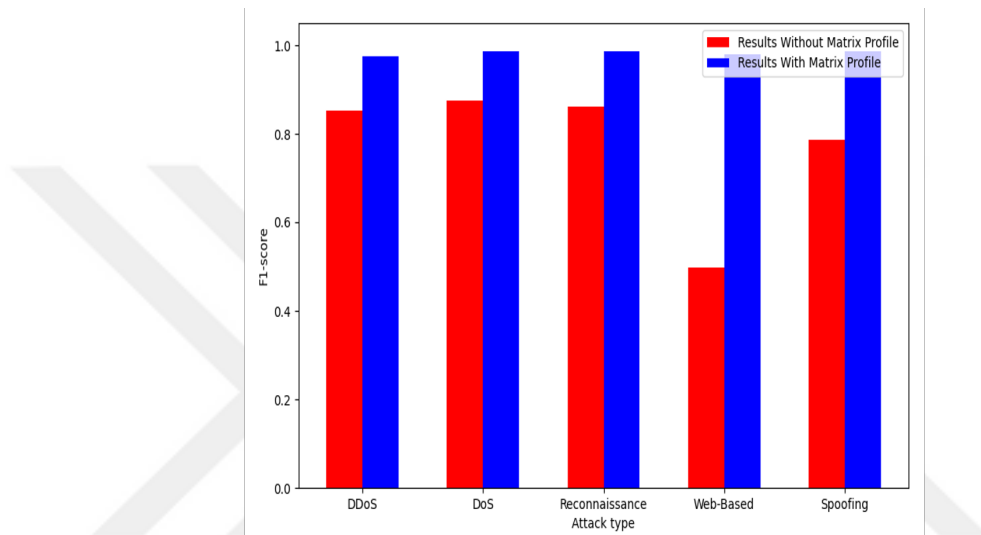


Figure 5.5. Effect of matrix profile method on F1-score for attack type.

A comparison between different machine learning algorithms is given in Figure 5.6. Classification of attack class is performed using matrix profile and various machine learning algorithms. The random forest algorithm achieves the highest F1-score. In contrast SVM shows the lowest performance among them. These results indicate the effectiveness of machine learning algorithms when used in combination with matrix profile for intrusion detection tasks.

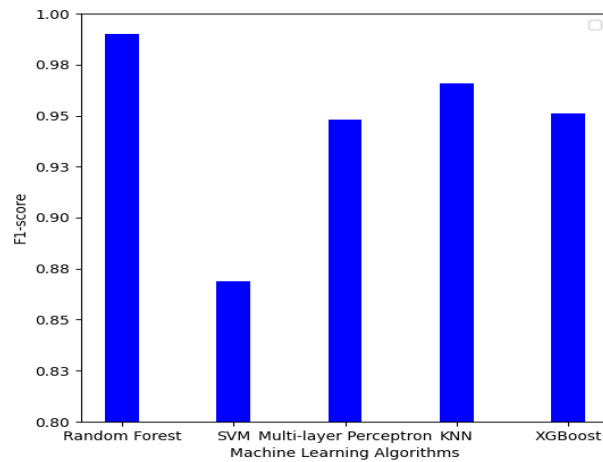


Figure 5.6. Performance comparison of machine learning algorithms with matrix profile method.

5.3.2. Application Identification

The application identification results are obtained using Encrypted Network Traffic Classification data set. 2 types of classification is performed. Matrix profile and random forest algorithm is applied to classify application category and application type. Applications include Discord, Go To Meeting, Instagram, Line, Microsoft Teams, Snapchat, Tango, Slack and TikTok. There exist 4 main categories. Category classification classifies each of these applications whether they belong to audio chat, file transfer, text chat or video chat. Application type classification means, knowing that it belongs to audio chat, classifying which application it is. This means that the classification process first determines the general category of an application and then identifies the specific application within that category. Random forest is applied with 20 estimators and maximum depth of 5. %70 of the data is used in training and %30 of the data is used in testing. Confusion matrix for classification of application categories is given in Figure 5.7. This confusion matrix represents the performance of a classification model in categorizing applications into four main categories: Audio Chat, Text Chat, Video Chat, and File Transfer. The diagonal values indicate high accuracy across all categories, with Audio Chat and Video Chat achieving nearly perfect classification. Text Chat and File Transfer also show strong performance but exhibit slightly

higher misclassification rates. It is also seen that, Text Chat is confused with Video Chat (%2.33), and File Transfer is misclassified as Video Chat (%3.02), suggesting that these categories share overlapping features. Overall, the model demonstrates excellent classification performance with minimal misclassification errors. Performance table is given in Table 5.11. This performance table presents the classification results for application categories using metrics such as Precision, Recall, F1-Score, and Support. The overall accuracy is %98.68 which indicates the good performance of the model. It can be seen that there are less instances in Text Chat and File Transfer. The F1 score of the classifier is %98.68.

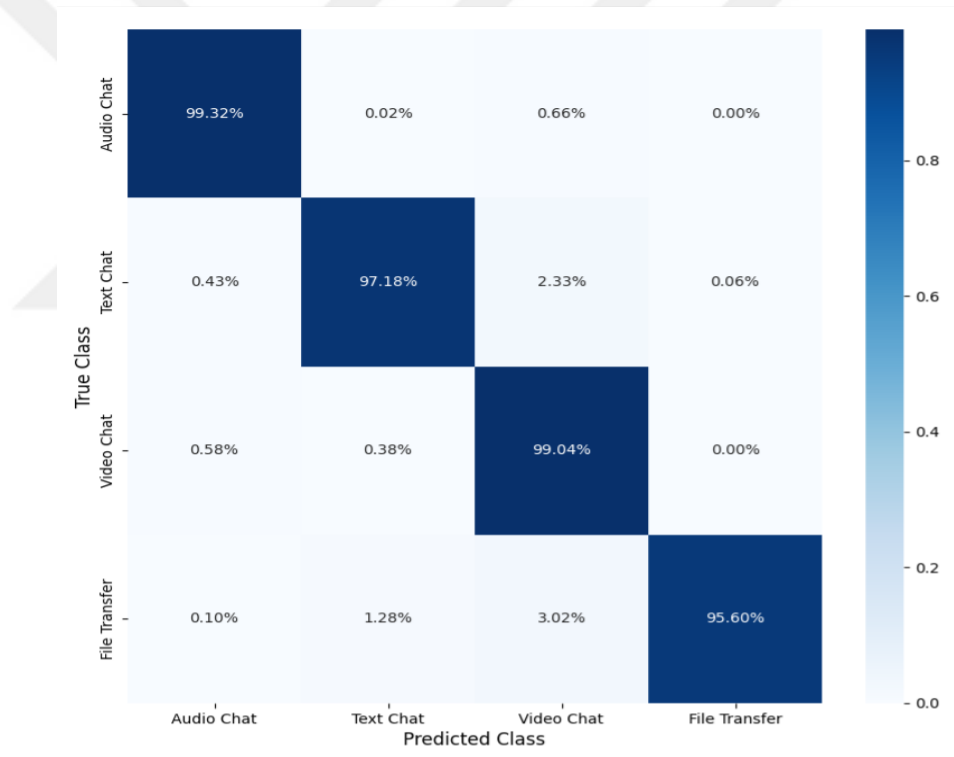


Figure 5.7. Confusion matrix of application categories using matrix profile based random forest method with Encrypted Network Traffic Classification data set.

Table 5.11. Performance Table of Application Categories using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.

Class	Precision	Recall	F1-Score	Support
Audio Chat	0.992	0.993	0.993	16382
Text Chat	0.978	0.972	0.975	5103
Video Chat	0.983	0.990	0.986	18068
File Transfer	0.999	0.956	0.977	2977
Macro Avg	0.988	0.978	0.983	42530
Weighted Avg	0.987	0.987	0.987	42530

Identification of specific application within audio chat category is performed. 10 estimators are used and maximum depth of 4 is set as the random forest parameters. Performance table for Audio Chat is given in Table 5.12. Overall accuracy of the model is %97.38, demonstrating strong classification capability across different applications. Discord and Go To Meeting are classified with almost perfect accuracy whereas Tango has the weakest performance.

Table 5.12. Performance Table of Audio Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.

	Precision	Recall	F1-Score	Support
Snapchat	0.975	0.978	0.976	2999
Microsoft Teams	0.977	0.992	0.984	2888
Go To Meeting	0.967	1.00	0.983	3035
Discord	0.994	0.989	0.991	1658
Instagram	0.957	0.994	0.975	2801
Tango	0.965	0.846	0.902	1376
Line	0.999	0.942	0.970	1518
Macro Avg	0.976	0.963	0.969	16275
Weighted Avg	0.974	0.974	0.973	16275

The recognition of specific applications within the file transfer category is performed. Performance table for File Transfer is given in Table 5.13. The overall accuracy of the model is %95.98. The results indicate that Slack achieved the highest performance with an F1-Score of %99, showing strong classification accuracy. Discord has the lowest recall %87, suggesting that some actual Discord instances were misclassified as other applications. It is worth noting that Discord has the lowest number of instances with support number of 379 which may have contributed to its lower recall value. This suggests a potential class imbalance issue that could affect the model's ability to generalize across underrepresented classes.

Table 5.13. Performance Table of File Transfer using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.

	Precision	Recall	F1-Score	Support
Microsoft Teams	0.944	0.962	0.953	818
Slack	0.982	0.988	0.985	774
Line	0.948	0.970	0.959	1017
Discord	0.985	0.871	0.924	379
Macro Avg	0.965	0.948	0.955	2988
Weighted Avg	0.960	0.960	0.960	2988

The identification of individual applications within the text chat category has been performed. Performance table for Text Chat can be seen in Table 5.14. Overall accuracy is %98.67 indicating that the classifier performs exceptionally well in distinguishing between different text chat applications. The model maintains robustness even under class imbalance, as reflected by the varying support values across applications. TikTok, Slack, and Instagram exhibit the strongest performance, while Line has the most misclassifications due to lower recall.

Table 5.14. Performance Table of Text Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.

	Precision	Recall	F1-Score	Support
Snapchat	0.970	0.985	0.978	866
TikTok	0.989	0.997	0.993	1254
Tango	0.994	0.979	0.987	720
Discord	0.997	0.971	0.983	307
Slack	0.984	0.999	0.991	745
Microsoft Teams	0.994	0.975	0.985	522
Instagram	0.986	0.996	0.991	569
Line	1.000	0.917	0.957	132
Macro Avg	0.989	0.977	0.983	5115
Weighted Avg	0.987	0.987	0.987	5115

Identification of specific application within video chat category is performed. Performance table for Video Chat can be seen in Table 5.15. The accuracy of the model is %96.5, which suggests that it performs well in classifying video chat applications. Macro average F1-Score (%96.1) and weighted average F1-Score (%96.5) indicate strong and balanced performance across classes. Microsoft Teams and Go To Meeting is the most accurately classified applications whereas Line has the highest misclassification rate suggesting difficulty in identifying all Line-related video traffic instances. Applications such as Microsoft Teams, GoToMeeting, and Tango are classified with high confidence, achieving F1-scores above 0.95, which reflects the model's ability to capture distinctive temporal patterns.

Despite the impressive classification performance overall, the results reveal that applications with smaller support values, such as Line and Tango, tend to have more variability in recall. This may point to underlying challenges in distinguishing apps with similar traffic patterns or limited training samples.

Table 5.15. Performance Table of Video Chat using Matrix Profile based Random Forest method with Encrypted Network Traffic Classification Data Set.

Class	Precision	Recall	F1-Score	Support
Microsoft Teams	0.993	0.997	0.995	2934
Go To Meeting	0.985	0.997	0.991	3096
Snapchat	0.979	0.976	0.978	2931
Line	0.968	0.876	0.920	1516
Instagram	0.916	0.960	0.937	3163
Discord	0.942	0.961	0.951	3067
Tango	0.998	0.921	0.958	1445
Macro Avg	0.969	0.956	0.961	18152
Weighted Avg	0.966	0.965	0.965	18152

The comparison between results with and without matrix profile across different application categories demonstrates a consistent improvement in F1-score when the matrix profile is applied. The results can be seen in Figure 5.8. The most significant improvements are seen in Text Chat and File Transfer categories. The results show that the matrix profile method enhances the effectiveness of the selected feature set, enabling better classification between application categories. The improvements across all categories confirm that applying matrix profile to the feature set contributes to a more robust classification performance and makes it a valuable technique.

A comparison was also made between the results obtained with and without matrix profile method for application type detection which can be seen in Figure 5.9. Matrix Profile method consistently enhances F1-score across all application types. The most significant improvement is observed in Text Chat where the F1-score increased from 0.43 to 0.98 . Similarly File Transfer category showed a significant improvement with the application of matrix profile.

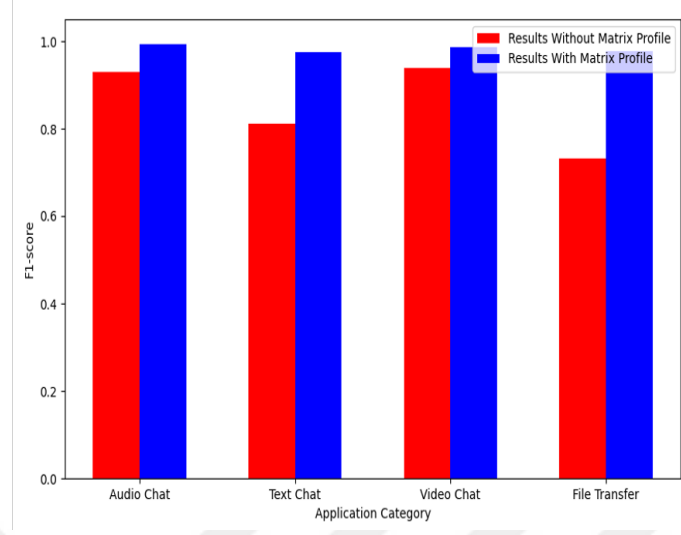


Figure 5.8. Effect of matrix profile method on F1-score for application category.

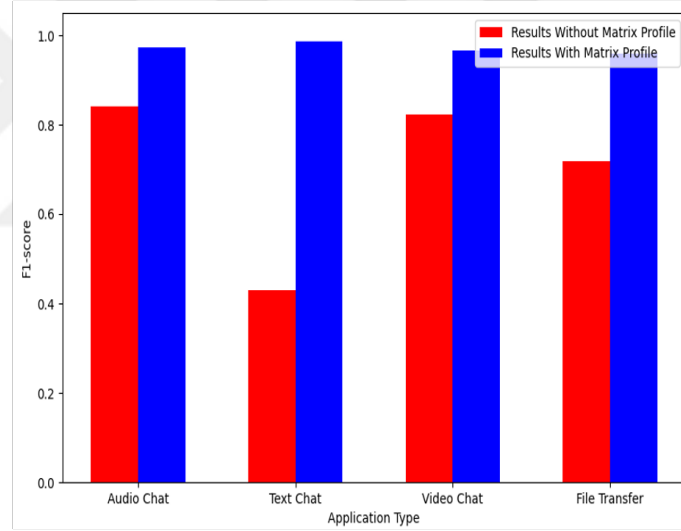


Figure 5.9. Effect of matrix profile method on F1-score for attack type.

5.4. Gramian Angular Summation Field Based CNN Model Results

In this study, CIC IoT 2023 data set is used. The window size is determined as 1 second. From every packet in this 1 second period, 35 features are extracted. The features extracted from the first window can be shown as $X_i = [X_1, X_2, \dots, X_N]$, where X_i represents the feature vector and N represents the number of features. In order to prepare the data for CNN, overlapping segments are constructed to extract meaningful features from the data. Firstly, all of the features are normalized between $[-1,1]$. The

segment size is set as 20 and overlap is determined as 5. GASF images are constructed from each segment. Since the segment size is determined as 20, the GASF matrix will be of size (20×20) . After considering all of the features, the resulting GASF matrix for one segment will be of size $(20 \times 20 \times 35)$. The total number of segments for the time-series data can be calculated as

$$\text{Total segment number} = \frac{\text{Total size of time series} - \text{segment size}}{\text{overlap}} + 1. \quad (5.1)$$

In total 297 GASF images of size $(20 \times 20 \times 35)$ are constructed. Each sub-class is classified using GASF images as an input to CNN. In every simulation, 70% of the GASF images is used in training of the CNN model and 30% is used in test. Number of epoch is defined as 5 and batch size is set as 32.

5.4.1. Intrusion Detection

The intrusion detection method is tested with CIC IoT 2023 data set. Attack class and attack type classification is performed using GASF and CNN. 7 attack classes are classified. The confusion matrix is given in Figure 5.10. In confusion matrix, diagonal entries represent instances which are classified correctly. Spoofing, Brute Force and Benign classes are classified with 100% accuracy. There are small confusions between DDoS and DoS classes. The highest misclassification percentage is in web-based attacks of which 98.73% is correctly classified but 2.50% is misclassified as Brute Force.

The classification results are given in Table 5.16. The overall accuracy is 99.93%. Precision, recall, F1-score values are high in each category. Macro and weighted average values are 99% which indicates minimal misclassification.

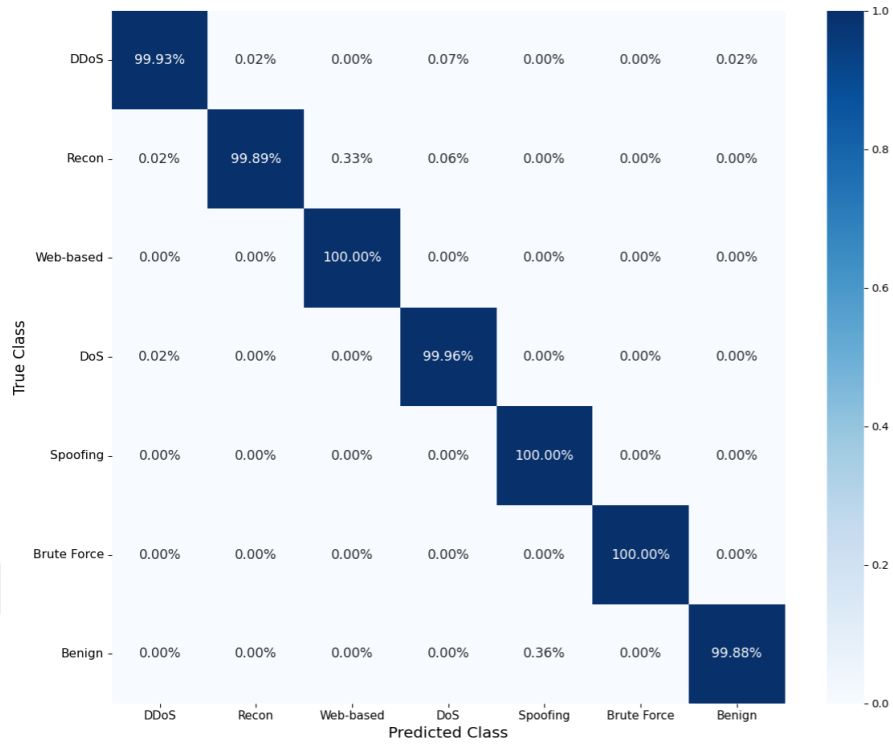


Figure 5.10. Confusion matrix of attack classes using GASF based CNN model with CIC IoT 2023 data set.

Table 5.16. Performance Table of Attack Classes using GASF based CNN model with CIC IoT 2023 Data Set.

Class	Precision	Recall	F1-Score	Support
DDoS	0.999	0.999	0.999	8943
Recon	0.999	0.998	0.999	5699
Web-based	0.996	1.000	0.998	299
DoS	0.998	0.999	0.999	5448
Spoofing	0.996	1.000	0.998	1961
Brute force	1.000	1.000	1.000	119
Benign	0.999	0.998	0.999	6013
Macro Avg	0.998	0.999	0.999	28482
Weighted Avg	0.999	0.999	0.999	28482

DDoS attacks include 7 sub-classes. Multiclass classification considering these 7 sub-classes are performed. Performance table is given in Table 5.17. Overall, 99.7% accuracy is obtained. Precision, recall and F1-score values are consistently high across all DDoS attack types. Slow Loris, UDP Flood and ACK Fragmentation attacks have perfect recall meaning that all instances are correctly detected.

Table 5.17. Performance Table of DDoS Class using GASF based CNN model with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
HTTP Flood	0.998	0.994	0.997	342
PSHACK Flood	0.988	0.997	0.992	1352
Slow Loris	0.998	1.000	0.999	514
SYN Flood	0.997	0.999	0.998	1722
TCP Flood	0.999	0.990	0.995	1734
UDP Flood	0.997	1.000	0.997	2009
ACK Fragmentation	0.996	1.000	0.996	1263
Macro Avg	0.997	0.997	0.997	8936
Weighted Avg	0.997	0.997	0.997	8936

DoS attacks include 4 sub-classes. Multiclass classification considering these 4 sub-classes are performed. The performance table for DoS attacks can be seen in Table 5.18. The macro and weighted averages which are %99.8 and %99.9 respectively indicate that the model generalizes well across all DoS classes. Overall, %99.87 accuracy is obtained. Among all subtypes, UDP Flood reaches perfect scores, while HTTP Flood, SYN Flood and TCP Flood each score an F1-score of 99.8%, demonstrating highly consistent and accurate detection performance across different DoS types.

Table 5.18. Performance Table of DoS Class using GASF based CNN model with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
HTTP Flood	0.996	1.000	0.998	464
SYN Flood	0.999	0.997	0.998	1558
TCP Flood	0.997	0.999	0.998	1508
UDP Flood	1.000	1.000	1.000	1980
Macro Avg	0.998	0.999	0.998	5510
Weighted Avg	0.999	0.999	0.999	5510

Reconnaissance attacks include 5 sub-classes. Multiclass classification considering these 5 sub-classes are performed. The performance table can be seen in Table 5.19. The highest recall is observed in Ping Sweep which means that almost all instances of this attack were correctly identified. The support values represent the number of instances present in that class. Looking at the support values, it is seen that vulnerability scan has the highest number of samples with 3467 instances, whereas ping sweep has the least number of samples with 18 instances which is an impact on the model performance. The overall accuracy of the model is 99.96%.

Table 5.19. Performance Table of Reconnaissance Class using GASF based CNN model with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
OS Scan	0.997	0.998	0.997	481
Ping Sweep	0.997	0.999	0.998	18
Port Scan	0.996	0.998	0.997	452
Vulnerability Scan	0.998	0.997	0.998	3467
Host Discovery	0.997	0.998	0.997	1248
Macro Avg	0.998	0.999	0.998	5666
Weighted Avg	0.999	0.999	0.999	5666

Spoofing attacks include 2 sub-classes. Multi class classification considering these 2 sub-classes are performed. The performance table is given in Table 5.20. The GASF-based CNN model exhibits exceptionally high performance in detecting spoofing attacks, achieving perfect or near-perfect metrics across both DNS and ARP Spoofing classes. The model's ability to distinguish between spoofing attacks is confirmed also from the macro and weighted averages. For spoofing attacks overall accuracy is calculated as %99.84. Such consistent performance, even with relatively limited support sizes (991 and 903), validates the model's applicability to real-world intrusion detection systems where spoofing attacks are typically rare but highly impactful.

Table 5.20. Performance Table of Spoofing Class using GASF based CNN model with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
DNS Spoofing	1.000	0.997	0.998	991
ARP Spoofing	0.997	1.000	0.998	903
Macro Avg	0.998	0.998	0.998	1894
Weighted Avg	0.998	0.998	0.998	1894

Web-based attacks include 6 sub-classes. Multiclass classification considering these 6 sub-classes are performed. The performance table is given in Table 5.21. SQL Injection and Backdoor Malware are identified with perfect or near-perfect accuracy indicating the model's capability to learn distinct temporal features. However, Uploading Attack exhibits the lowest recall %77.3 and consequently the lowest F1-score %87.2, likely due to its very limited support which is only 22 instances. For web-based attacks overall accuracy is calculated as %96.15.

Table 5.21. Performance Table of Web-based Class using GASF based CNN model with CIC IoT 2023 Data Set.

	Precision	Recall	F1-Score	Support
Browser Hijacking	1.000	0.893	0.943	56
Command Injection	0.887	1.000	0.940	71
SQL Injection	1.000	1.000	1.000	54
Uploading Attack	1.000	0.773	0.872	22
XSS	0.968	0.984	0.976	61
Backdoor Malware	0.980	1.000	0.990	48
Macro Avg	0.972	0.942	0.953	312
Weighted Avg	0.965	0.962	0.961	312

5.4.2. Application Identification

In this study, Encrypted Network Traffic Classification data set is used. Audio chat, file transfer, text chat and video chat samples are combined. Classification of these 4 categories are performed using GASF and CNN. In the CNN model, epoch number is set as 5 and batch size as 32. Performance table is given in Table 5.22. The confusion matrix is given in Figure 5.11. Overall accuracy is %99.86.

Table 5.22. Performance Table of Application Categories using GASF based CNN model with Encrypted Network Traffic Classification Data Set.

Class	Precision	Recall	F1-Score	Support
Audio Chat	1.000	0.999	0.999	3958
File Transfer	1.000	0.992	0.996	750
Text Chat	0.998	0.999	0.998	1245
Video Chat	0.998	0.999	0.998	4539
Macro Avg	0.999	0.997	0.998	10492
Weighted Avg	0.999	0.999	0.999	10492

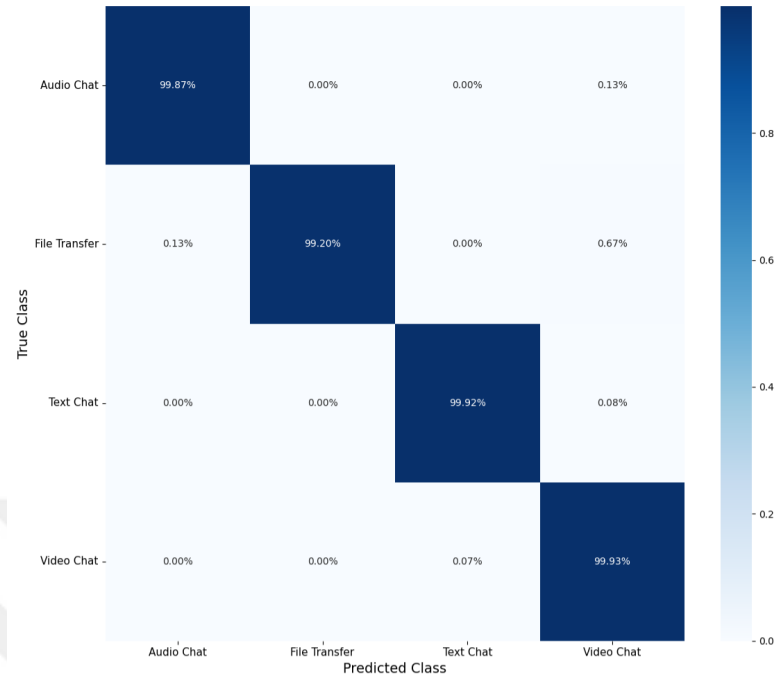


Figure 5.11. Confusion matrix of application categories using GASF based CNN model with Encrypted Network Traffic Classification data set.

Audio Chat category includes 7 applications. The aim is to classify these applications using GASF and CNN. The performance table is given in Table 5.23. Support numbers represent the testing data which consist 30% of the overall data. Overall, 99.93% accuracy is obtained. The accuracy result indicates that our proposed approach has a good performance on classifying applications under audio chat category.

File Transfer category includes 4 applications. Performance table for file transfer is given in Table 5.24. Overall, 98.89% accuracy is obtained. The performance table demonstrates that the GASF-based CNN model achieves exceptionally high classification accuracy for file transfer applications, with F1-scores above 0.97 for all classes and perfect scores for Microsoft Teams.

Table 5.23. Performance Table of Audio Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.

Application	Precision	Recall	F1-Score	Support
GoToMeeting	1.000	0.998	0.999	752
Discord	1.000	1.000	1.000	396
Microsoft Teams	0.998	1.000	0.999	728
Tango	1.000	1.000	1.000	347
Line	1.000	1.000	1.0000	386
Instagram	0.997	1.000	0.998	686
Snapchat	1.000	0.997	0.998	722
Macro Avg	0.999	0.999	0.999	4017
Weighted Avg	0.999	0.999	0.999	4017

Table 5.24. Performance Table of File Transfer using GASF based CNN model with Encrypted Network Traffic Classification Data Set.

Application	Precision	Recall	F1-Score	Support
Line	0.987	0.992	0.990	232
Discord	0.990	0.997	0.994	109
Microsoft Teams	1.000	1.000	1.000	206
Slack	0.965	0.978	0.970	190
Macro Avg	0.980	0.985	0.983	737
Weighted Avg	0.980	0.985	0.983	737

Text Chat category includes 8 applications. Performance table for text chat is given in Table 5.25. Overall accuracy of the model is 99.68%. The model demonstrates both accuracy and consistency, effectively handling class imbalance and ensuring reliable classification. These results highlight the effectiveness of the CNN architecture combined with GASF image transformation in handling encrypted traffic classification tasks. The ability to achieve near-perfect or perfect classification across diverse

platforms emphasizes the representational power of GASF in preserving meaningful temporal features in the time-series data. Moreover, even with relatively low support values in some classes such as Line and Discord, the model generalizes well, demonstrating resilience against class imbalance.

Table 5.25. Performance Table of Text Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.

Application	Precision	Recall	F1-Score	Support
Slack	0.994	1.000	0.997	186
Discord	0.974	1.000	0.987	76
Instagram	1.000	1.000	1.000	131
Line	1.000	0.971	0.985	35
Microsoft Teams	1.000	1.000	1.000	140
Tango	1.000	0.988	0.994	179
Snapchat	0.995	1.000	0.997	229
Tiktok	1.000	0.996	0.998	284
Macro Avg	0.995	0.995	0.995	1260
Weighted Avg	0.995	0.995	0.995	1260

Video Chat category includes 7 applications. Performance table for video chat is given in Table 5.26. The performance table for video chat classification using the GASF-based CNN model shows consistently high precision, recall, and F1-scores across all applications, with most classes exceeding F1-score of 99% . Particularly, Instagram and Snapchat achieved perfect scores, while the macro and weighted averages of 0.995 and 0.994, respectively, confirm the model's strong generalization and effectiveness in handling encrypted video traffic. Overall, 99.48% accuracy is obtained.

Table 5.26. Performance Table of Video Chat using GASF based CNN model with Encrypted Network Traffic Classification Data Set.

Application	Precision	Recall	F1-Score	Support
GoToMeeting	1.000	0.987	0.999	773
Discord	0.991	0.995	0.994	723
Microsoft Teams	0.998	0.995	0.996	745
Tango	0.970	0.980	0.974	354
Line	0.985	0.989	0.986	385
Instagram	1.000	1.000	1.000	749
Snapchat	1.000	1.000	1.000	750
Macro Avg	0.995	0.995	0.995	4479
Weighted Avg	0.994	0.994	0.994	4479

6. CONCLUSION & FUTURE WORK

Although several academic research is proceeding for the subject area, novel approaches are really required that avoid difficulties experienced in trying to determine exact network intrusion type. With this thesis, a novel representation approach is presented to examine the applicability of machine learning methodologies and convolutional neural networks to resolve network intrusion and application recognition problem. Network application classification recognizes applications in the network traffic. This information is critical for network management and maintaining quality of service. This thesis evaluates Random Forest and CNN models with novel mathematical algorithms such as Matrix Profile and GASF to reach accurate classification performance. This thesis leverages advances in machine learning studies with application of GASF time-series to image conversion and Matrix Profile algorithms to build a novel solution for network intrusion and application classification problem.

Issues of traditional systems may be cleared with appropriate examination of such time-series analysis methods. These algorithms advance success characteristics of machine learning and CNN classifiers. This work proves that machine learning models may benefit image-based representation techniques. Therefore, a clear perspective for network packet data representation is presented that may be utilized for further studies. As shown with this thesis, Matrix Profile and GASF algorithms improve success of machine learning and deep learning models if they are utilized for network intrusion and application classification. It is noted that, Matrix Profile and GASF mathematics provide obvious differences in represented time-series that may be used for classification. This thesis proves the ability of GASF mathematics, when applied as a visualization mechanism for time-series data, in a way that layered convolution structure of neural network achieves superior performance for feature extraction and classification.

It should be stated that, utilizing appropriate feature vectors that may successfully differentiate network attacks is complicated. Relevant feature selection is definitely a major achievement of this study. This study successfully characterizes time-series data based on the premise that, if feature representation reveals valuable data with not readily uncovered patterns, best classification results may be achieved. Also, feature representation method used in this work does not compromise information present in original time-series. Multi-stage model presented in this thesis is based on Matrix Profile and GASF processing of time-series feature data and machine learning application for classification. Image transformation methods process time-series data in various patterns and emphasize novel characteristics. GASF algorithms redefines classification as an image recognition effort. Therefore, classification problem is studied as image processing work. New characteristics are uncovered with expression of features as images. Also, time-series feature data is examined with Matrix Profile algorithm. Matrix Profile processes time-series for anomalies. Method gathers similarities in time-series. Therefore, outliers are found that indicates intrusions. In this thesis, Matrix Profile algorithm is utilized to improve outcome of Random Forest machine learning model. Results of validation show that accurate and reliable classifier models are implemented with this study. This may be expressed as one of major benefits of this study. It should be stated that, Matrix Profile is shown to be significantly effective for classification. Algorithms is robust and requires low configuration. It is used with Random Forest machine learning model to improve classification success. Matrix Profile algorithm processes features before Random Forest machine learning classifier to make them more distinctive.

Evaluation of the model is mainly focused on certain datasets which may indicate dependence. If presented method is validated for different datasets, usage may be generalized. Therefore, a methodology toward general intrusion detection may be implemented that covers most types of attacks. Recommendation for subsequent academic research is to spread development effort to consider and include multiple datasets to reach validation of general applicability. Maintaining scalability of approach for varying size of datasets is also an important aspect to consider for further studies. It

should be stated that, processor systems that manage networks run on several resource constraints. Therefore, processing power limitations need to be addressed. Lowering processor occupancy for the platform with the help of improved algorithms may surely be a way for future studies. Also, deploying platform for real-time usage may be subject of further studies. Performance metrics of such implementation may be determined to advance model structure and classification. It should be mentioned that, neural network parameter tuning is important aspect of performance improvement. Classical experimental process may not cover all cases that may be encountered and generalization of the model may be disturbed. Therefore, neural network tuning procedure development is open to further research and may be subject of future studies to enhance the findings of this thesis. Carefully designed parameters of machine learning classifiers are surely expected to improve overall performance metrics. Improved time-series to image transformation may reduce computational burden on servers which is necessary for real-time applications. Studies to improve execution time of used algorithms may have major effect especially when constraints such as processing power requirements are considered. Such requirements are major design concern and needs to be resolved for whole platform to be functional. However, it may be subject of another study. This work suggests acquisition of a generalized method to apply to real-time network security context. Promising results point to such possibility with novel representation algorithms. Further studies may achieve significant results.

REFERENCES

1. David, J. and C. Thomas, “Efficient DDoS Flood Attack Detection Using Dynamic Thresholding on Flow-Based Network Traffic”, *Computers & Security*, Vol. 82, No. 3, pp. 284–295, 2019.
2. Kumari, P. and A. K. Jain, “A Comprehensive Study of DDoS Attacks over IoT Network and Their Countermeasures”, *Computers & Security*, Vol. 127, p. 103096, 2023.
3. Zhou, Z., Q. Ai, P. Lou, J. Hu and J. Yan, “A Novel Method for Rolling Bearing Fault Diagnosis Based on Gramian Angular Field and CNN-ViT”, *Sensors*, Vol. 24, No. 12, p. 3967, 2024.
4. Li, X., J. Zhang, B. Xiao, Y. Zeng, S. Lv, J. Qian and Z. Du, “Fault Diagnosis of Hydropower Units Based on Gramian Angular Summation Field and Parallel CNN”, *Energies*, Vol. 17, No. 13, p. 3084, 2024.
5. Wu, Q., Y. Li, H. Ouyang and Y. Xuan, “A Multimodal Fusion Human Activity Recognition Method Based on CNN and Gramian Angular Field”, *Third International Conference on Advanced Algorithms and Neural Networks (AANN 2023)*, Vol. 12791 of *Proceedings of SPIE*, pp. 368–373, Xi’an, China, 2023.
6. Yang, C.-L., Z.-X. Chen and C.-Y. Yang, “Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images”, *Sensors*, Vol. 20, No. 1, p. 168, 2019.
7. Ateş, Ç., S. Özdel and E. Anarım, “Clustering-Based DDoS Attack Detection Using the Relationship Between Packet Headers”, *Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–6, Izmir, Turkey, 2019.

8. Erhan, D. and E. Anarim, “Hybrid DDoS Detection Framework Using Matching Pursuit Algorithm”, *IEEE Access*, Vol. 8, pp. 118912–118923, 2020.
9. Salmi, S. and L. Oughdir, “Performance Evaluation of Deep Learning Techniques for DoS Attacks Detection in Wireless Sensor Network”, *Journal of Big Data*, Vol. 10, No. 1, p. 17, 2023.
10. Sarhan, M., S. Layeghy, N. Moustafa, M. Gallagher and M. Portmann, “Feature Extraction for Machine Learning-Based Intrusion Detection in IoT Networks”, *Digital Communications and Networks*, Vol. 10, No. 1, pp. 205–216, 2024.
11. Kadri, M. R., A. Abdelli, J. B. Othman and L. Mokdad, “Survey and Classification of DoS and DDoS Attack Detection and Validation Approaches for IoT Environments”, *Internet of Things*, Vol. 25, p. 101021, 2024.
12. De Neira, A. B., B. Kantarci and M. Nogueira, “Distributed Denial of Service Attack Prediction: Challenges, Open Issues and Opportunities”, *Computer Networks*, Vol. 222, p. 109553, 2023.
13. Dong, S., K. Abbas and R. Jain, “A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments”, *IEEE Access*, Vol. 7, pp. 80813–80828, 2019.
14. Kumar, A., K. Abhishek, M. R. Ghalib, A. Shankar and X. Cheng, “Intrusion Detection and Prevention System for an IoT Environment”, *Digital Communications and Networks*, Vol. 8, No. 4, pp. 540–551, 2022.
15. Ma, X., Q. Li, X. Wang, Q. Wen and Y. Zeng, “A Brute-Force Attack Detection Method Based on the Combination of Artificial Neural Network and Autoencoder”, *Proceedings of the 2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE)*, pp. 447–450, IEEE, Chongqing, China, 2024.

16. Navruzov, E. and A. Kabulov, “Detection and Analysis Types of DDoS Attack”, *Proceedings of the 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–7, IEEE, Vancouver, BC, Canada, 2022.
17. Rios, V. D. M., P. R. Inácio, D. Magoni and M. M. Freire, “Detection and Mitigation of Low-Rate Denial-of-Service Attacks: A Survey”, *IEEE Access*, Vol. 10, pp. 76648–76668, 2022.
18. Bukhowah, R., A. Aljughaiman and M. H. Rahman, “Detection of DoS Attacks for IoT in Information-Centric Networks Using Machine Learning: Opportunities, Challenges, and Future Research Directions”, *Electronics*, Vol. 13, No. 6, p. 1031, 2024.
19. Alani, M. M. and E. Damiani, “XRecon: An Explainable IoT Reconnaissance Attack Detection System Based on Ensemble Learning”, *Sensors*, Vol. 23, No. 11, p. 5298, 2023.
20. Tekerek, A., “A Novel Architecture for Web-Based Attack Detection Using Convolutional Neural Network”, *Computers & Security*, Vol. 100, p. 102096, 2021.
21. Aldabbas, H. and R. Amin, “A Novel Mechanism to Handle Address Spoofing Attacks in SDN-Based IoT”, *Cluster Computing*, Vol. 24, No. 4, pp. 3011–3026, 2021.
22. Kang, J., M. Kim, J. Park and S. Park, “Time-Series to Image-Transformed Adversarial Autoencoder for Anomaly Detection”, *IEEE Access*, 2024.
23. Bayram, F. S., M. Dwedar, A. Melke, R. Schneider, R. Radtke and A. Jesser, “AnomalyBAGS: Bagging-Based Ensemble Learning with Time-Series-to-Image Transformations for Robust Anomaly Detection”, *2023 IEEE 8th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pp. 1–9, IEEE, Bahrain, Bahrain, 2023.

24. Altunkaya, D., F. Y. Okay and S. Ozdemir, “Image Transformation for IoT Time-Series Data: A Review”, *arXiv preprint arXiv:2311.12742*, 2023.
25. Lansky, J., S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh and A. M. Rahmani, “Deep Learning-Based Intrusion Detection Systems: A Systematic Review”, *IEEE Access*, Vol. 9, pp. 101574–101599, 2021.
26. Choi, K., J. Yi, C. Park and S. Yoon, “Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines”, *IEEE access*, Vol. 9, pp. 120043–120065, 2021.
27. Jang, Y., D.-W. Kim, G.-Y. Shin, S. Cho, K. Kim, J. Kang and M.-M. Han, “An Investigation of Learning Model Technologies for Network Traffic Classification Design in Cyber Security Exercises”, *IEEE Access*, Vol. 11, pp. 138712–138731, 2023.
28. Falowo, O. I., M. Ozer, C. Li and J. B. Abdo, “Evolving Malware and DDoS Attacks: Decadal Longitudinal Study”, *IEEE Access*, Vol. 12, pp. 28049–28060, 2024.
29. Uddin, R., S. A. Kumar and V. Chamola, “Denial of Service Attacks in Edge Computing Layers: Taxonomy, Vulnerabilities, Threats and Solutions”, *Ad Hoc Networks*, Vol. 152, p. 103322, 2024.
30. Zhang, C., D. Jia, L. Wang, W. Wang, F. Liu and A. Yang, “Comparative Research on Network Intrusion Detection Methods Based on Machine Learning”, *Computers & Security*, Vol. 121, p. 102861, 2022.
31. Wang, L. and Y. Liu, “A DDoS Attack Detection Method Based on Information Entropy and Deep Learning in SDN”, *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Vol. 1, pp. 1084–1088, Chongqing, China, 2020.

32. Tsobdjou, L. D., S. Pierre and A. Quintero, “An Online Entropy-Based DDoS Flooding Attack Detection System with Dynamic Threshold”, *IEEE Transactions on Network and Service Management*, Vol. 19, No. 2, pp. 1679–1689, 2022.
33. Halladay, J., D. Cullen, N. Briner, J. Warren, K. Fye, R. Basnet, J. Bergen and T. Doleck, “Detection and Characterization of DDoS Attacks Using Time-Based Features”, *IEEE Access*, Vol. 10, pp. 49794–49807, 2022.
34. Bakar, R. A., L. De Marinis, F. Cugini and F. Paolucci, “FTG-Net-E: A Hierarchical Ensemble Graph Neural Network for DDoS Attack Detection”, *Computer Networks*, Vol. 250, p. 110508, 2024.
35. Shang, Y., “Prevention and Detection of DDoS Attack in Virtual Cloud Computing Environment Using Naive Bayes Algorithm of Machine Learning”, *Measurement: Sensors*, Vol. 31, p. 100991, 2024.
36. Shamekhi, A., P. Shamsinejad Babaki and R. Javidan, “An Intelligent Behavioral-Based DDoS Attack Detection Method Using Adaptive Time Intervals”, *Peer-to-Peer Networking and Applications*, Vol. 17, No. 4, pp. 2185–2204, 2024.
37. Fan, C., N. M. Kaliyamurthy, S. Chen, H. Jiang, Y. Zhou and C. Campbell, “Detection of DDoS Attacks in Software-Defined Networking Using Entropy”, *Applied Sciences*, Vol. 12, No. 1, p. 370, 2021.
38. Fouladi, R. F., C. E. Kayatas and E. Anarim, “Statistical Measures: Promising Features for Time Series-Based DDoS Attack Detection”, *Proceedings of the 5th International Conference on Time Series and Forecasting (ITISE 2018)*, Vol. 2, p. 96, MDPI, Granada, Spain, 2018.
39. Raza, M. S., M. N. A. Sheikh, I.-S. Hwang and M. S. Ab-Rahman, “Feature-Selection-Based DDoS Attack Detection Using AI Algorithms”, *Telecom*, Vol. 5, No. 2, pp. 333–346, 2024.

40. Yue, M., H. Wang, L. Liu and Z. Wu, “Detecting DoS Attacks Based on Multi-Features in SDN”, *IEEE Access*, Vol. 8, pp. 104688–104700, 2020.
41. Al-zubidi, A. F., A. K. Farhan and S. M. Towfek, “Predicting DoS and DDoS Attacks in Network Security Scenarios Using a Hybrid Deep Learning Model”, *Journal of Intelligent Systems*, Vol. 33, No. 1, p. 20230195, 2024.
42. Altulaihan, E., M. A. Almaiah and A. Aljughaiman, “Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms”, *Sensors*, Vol. 24, No. 2, p. 713, 2024.
43. Ashiku, L. and C. Dagli, “Network Intrusion Detection System Using Deep Learning”, *Procedia Computer Science*, Vol. 185, pp. 239–247, 2021.
44. Idrissi, M. J., H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui and I. Berrada, “Fed-AnIDS: Federated Learning for Anomaly-Based Network Intrusion Detection Systems”, *Expert Systems with Applications*, Vol. 234, p. 121000, 2023.
45. Abbas, S., S. Alsubai, S. Ojo, G. A. Sampedro, A. Almadhor, A. A. Hejaili and I. Bouazzi, “An Efficient Deep Recurrent Neural Network for Detection of Cyberattacks in Realistic IoT Environment”, *The Journal of Supercomputing*, Vol. 80, No. 10, pp. 13557–13575, 2024.
46. Alzahrani, A. O. and M. J. Alenazi, “Designing a Network Intrusion Detection System Based on Machine Learning for Software-Defined Networks”, *Future Internet*, Vol. 13, No. 5, p. 111, 2021.
47. Wang, Y.-C., Y.-C. Houn, H.-X. Chen and S.-M. Tseng, “Network Anomaly Intrusion Detection Based on Deep Learning Approach”, *Sensors*, Vol. 23, No. 4, p. 2171, 2023.
48. Ahmad, I., Q. E. Ul Haq, M. Imran, M. O. Alassafi and R. A. AlGhamdi, “An Efficient Network Intrusion Detection and Classification System”, *Mathematics*,

Vol. 10, No. 3, p. 530, 2022.

49. Leon, M., T. Markovic and S. Punnekkat, “Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection and Attack Classification”, *Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Padua, Italy, 2022.
50. Thakkar, A. and R. Lohiya, “Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network”, *IEEE Internet of Things Journal*, Vol. 10, No. 13, pp. 11888–11895, 2023.
51. Mohammadpour, L., T. C. Ling, C. S. Liew and A. Aryanfar, “A Survey of CNN-Based Network Intrusion Detection”, *Applied Sciences*, Vol. 12, No. 16, p. 8162, 2022.
52. Lan, J., X. Liu, B. Li, Y. Li and T. Geng, “DarknetSec: A Novel Self-Attentive Deep Learning Method for Darknet Traffic Classification and Application Identification”, *Computers & Security*, Vol. 116, p. 102663, 2022.
53. Diao, Z., G. Xie, X. Wang, R. Ren, X. Meng, G. Zhang, K. Xie and M. Qiao, “EC-GCN: An Encrypted Traffic Classification Framework Based on Multi-Scale Graph Convolution Networks”, *Computer Networks*, Vol. 224, p. 109614, 2023.
54. Galeano-Brajones, J., M. I. Chidean, F. Luna and J. Carmona-Murillo, “A Novel Approach for Flow Analysis in Software-Based Networks Using L-Moments Theory”, *Computer Communications*, Vol. 201, pp. 116–122, 2023.
55. Azab, A., M. Khasawneh, S. Alrabaei, K.-K. R. Choo and M. Sarsour, “Network Traffic Classification: Techniques, Datasets, and Challenges”, *Digital Communications and Networks*, Vol. 10, No. 3, pp. 676–692, 2024.
56. Cheng, Q., C. Wu, H. Zhou, D. Kong, D. Zhang, J. Xing and W. Ruan, “Machine Learning-Based Malicious Payload Identification in Software-Defined Networking”,

Journal of Network and Computer Applications, Vol. 192, p. 103186, 2021.

57. Kunang, Y. N., S. Nurmaini, D. Stiawan and B. Y. Suprpto, “Attack Classification of an Intrusion Detection System Using Deep Learning and Hyperparameter Optimization”, *Journal of Information Security and Applications*, Vol. 58, No. 1, p. 102804, 2021.
58. Oliveira, N., I. Praça, E. Maia and O. Sousa, “Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems”, *Applied Sciences*, Vol. 11, No. 4, p. 1674, 2021.
59. Audibert, J., P. Michiardi, F. Guyard, S. Marti and M. A. Zuluaga, “Do Deep Neural Networks Contribute to Multivariate Time Series Anomaly Detection?”, *Pattern Recognition*, Vol. 132, No. 3, p. 108945, 2022.
60. Lan, S.-Y., R.-Q. Chen and W.-L. Zhao, “Anomaly Detection on IT Operation Series via Online Matrix Profile”, *arXiv preprint arXiv:2108.12093*, 2021.
61. Scott, B. A., M. N. Johnstone, P. Szewczyk and S. Richardson, “Matrix Profile Data Mining for BGP Anomaly Detection”, *Computer Networks*, Vol. 242, p. 110257, 2024.
62. Yeh, C.-C. M., A. Der, U. S. Saini, V. Lai, Y. Zheng, J. Wang, X. Dai, Z. Zhuang, Y. Fan, H. Chen, P. O. Aboagye and E. Keogh, “Matrix Profile for Anomaly Detection on Multidimensional Time Series”, *arXiv preprint arXiv:2409.09298*, 2024.
63. Alzahrani, M. A., A. M. Alzahrani and M. S. Siddiqui, “Detecting DDoS Attacks in IoT-Based Networks Using Matrix Profile”, *Applied Sciences*, Vol. 12, No. 16, p. 8294, 2022.
64. Mercer, R., S. Alaei, A. Abdoli, S. Singh, A. Murillo and E. Keogh, “Matrix Profile XXIII: Contrast Profile: A Novel Time Series Primitive That Allows Real World

- Classification”, *Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1240–1245, IEEE, Auckland, New Zealand, 2021.
65. Madrid, F., S. Imani, R. Mercer, Z. Zimmerman, N. Shakibay and E. Keogh, “Matrix Profile XX: Finding and Visualizing Time Series Motifs of All Lengths Using the Matrix Profile”, *Proceedings of the 2019 IEEE International Conference on Big Knowledge (ICBK)*, pp. 175–182, IEEE, Beijing, China, 2019.
 66. Bamus, I. and F. Y. Okay, “Image Transformation for IoT-Based Anomaly Detection”, *Proceedings of the 2024 32nd Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, Antalya, Turkey, 2024.
 67. Elmir, Y., Y. Himeur and A. Amira, “ECG Classification Using Deep CNN and Gramian Angular Field”, *Proceedings of the 2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 137–141, IEEE, San Francisco, CA, USA, 2023.
 68. Ma, J. and H. Wang, “Anomaly Detection in Sensor Data via Encoding Time Series into Images”, *Journal of King Saud University-Computer and Information Sciences*, Vol. 36, No. 10, p. 102232, 2024.
 69. Wang, Z. and T. Oates, “Imaging Time-Series to Improve Classification and Imputation”, arXiv preprint arXiv:1506.00327, 2015.
 70. Tang, D., S. Wang, B. Liu, W. Jin and J. Zhang, “GASF-IPP: Detection and Mitigation of LDoS Attack in SDN”, *IEEE Transactions on Services Computing*, Vol. 16, No. 5, pp. 3373–3384, 2023.
 71. Herath, N., M. Herath, T. Thilakanayake, C. Angammana and M. Liyanage, “An Image-Based Disaggregation Study of Time Series Energy Data Using Gramian Angular Field”, *Proceedings of the 2023 IEEE PES 15th Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1–6, IEEE, Chongqing, China,

2023.

72. Chen, H., J. Wang and D. Shi, "Spatial-Temporal Correlation-Concerned Measurement Manipulation Detection Based on Gramian Angular Summation Field and Convolutional Neural Networks", *Proceedings of the 2021 IEEE 4th International Electrical and Energy Conference (CIEEC)*, pp. 1–6, IEEE, Wuhan, China, 2021.
73. Ahmad, R. S., A. H. Ali, S. M. Kazim and Q. Niyaz, "A GAF and CNN-Based Wi-Fi Network Intrusion Detection System", *Proceedings of the IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, New York, NY, USA, 2023.
74. Bansal, M., A. Goyal and A. Choudhary, "A Comparative Analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory Algorithms in Machine Learning", *Decision Analytics Journal*, Vol. 3, No. 11, p. 100071, 2022.
75. Azam, Z., M. M. Islam and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree", *IEEE Access*, Vol. 11, pp. 80348–80391, 2023.
76. Sun, Z., G. Wang, P. Li, H. Wang, M. Zhang and X. Liang, "An Improved Random Forest Based on the Classification Accuracy and Correlation Measurement of Decision Trees", *Expert Systems with Applications*, Vol. 237, No. B, p. 121549, 2024.
77. Ioannou, C. and V. Vassiliou, "Network Attack Classification in IoT Using Support Vector Machines", *Journal of Sensor and Actuator Networks*, Vol. 10, No. 3, p. 58, 2021.
78. Agarwal, M., K. S. Gill, R. Chauhan, A. Kapruwan and D. Banerjee, "Classification of Network Security Attack Using KNN (K-Nearest Neighbour) and Comparison of

- Different Attacks Through Different Machine Learning Techniques”, *Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON)*, pp. 1–7, IEEE, Bangalore, India, 2024.
79. Akhtar, M. S. and T. Feng, “Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time”, *Symmetry*, Vol. 14, No. 11, p. 2308, 2022.
 80. Ahmed, S., Z. A. Khan, S. M. Mohsin, S. Latif, S. Aslam, H. Mujlid, M. Adil and Z. Najam, “Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron”, *Future Internet*, Vol. 15, No. 2, p. 76, 2023.
 81. Ateş, Ç., S. Özdel and E. Anarım, “A New Network Anomaly Detection Method Based on Header Information Using Greedy Algorithm”, *Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 657–662, IEEE, Paris, France, 2019.
 82. Erhan, D. and E. Anarım, “Boğaziçi University Distributed Denial of Service Dataset”, *Data in Brief*, Vol. 32, No. 32, p. 106187, 2020.
 83. Neto, E. C. P., S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu and A. A. Ghorbani, “CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment”, *Sensors*, Vol. 23, No. 13, p. 5941, 2023.
 84. Fotse, Y. S. N., V. K. Tchendji and M. Velempini, “Federated Learning-Based DDoS Attacks Detection in Large-Scale Software-Defined Network”, *IEEE Transactions on Computers*, Vol. 74, No. 1, pp. 101–115, 2025.
 85. Ren, X., H. Gu and W. Wei, “Tree-RNN: Tree Structural Recurrent Neural Network for Network Traffic Classification”, *Expert Systems with Applications*, Vol. 167, p. 114363, 2021.
 86. Naiem, S., A. E. Khedr, A. M. Idrees and M. I. Marie, “Enhancing the Efficiency

of Gaussian Naïve Bayes Machine Learning Classifier in the Detection of DDoS in Cloud Computing”, *IEEE Access*, Vol. 11, pp. 124597–124608, 2023.

87. Chan, K. Y., B. Abu-Salih, R. Qaddoura, A. Al-Zoubi, V. Palade, D.-S. Pham, J. Del Ser and K. Muhammad, “Deep Neural Networks in the Cloud: Review, Applications, Challenges and Research Directions”, *Neurocomputing*, Vol. 545, No. C, p. 126327, 2023.

