Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT for Internet & Multimedia

Traffic anomaly detection based on its 2D representation

Master Candidate

Nicolò Scialpi

Supervisor

Prof.ssa Federica Battisti

Co-supervisor

Dr.ssa Sara Baldoni, Michael Neri

Academic Year
2022/2023

To the most passionate professor Federica,
to Valentina,
my parents
and friends.
To the future.

Abstract

Cyber Physical Systems (CPS) have become increasingly prevalent in various sectors such as healthcare, transport, energy, and industrial systems. However, the communication capability of these systems exposes them to numerous network-level threats, which calls for ensuring their security. In this study, we propose a deep learning-based anomaly detection system that uses a 2D representation of the network traffic to address this issue. Specifically, we suggest utilizing a time-sensitive representation of the traffic in a 2D matrix and an autoencoder to model the nominal system behavior and identify anomalies. The hypothesis underlying this approach is that anomalous samples cannot be accurately reconstructed by the model trained on normal data. The results obtained from the proposed method are encouraging and demonstrate its effectiveness. Moreover, this work provides a basis for further research in this direction.

ITALIANO:
I sistemi cibernetico-fisici (CPS) sono diventati sempre più diffusi in vari settori come la sanità, i trasporti, l'energia e i sistemi industriali. Tuttavia, la capacità di comunicazione di questi sistemi li espone a numerose minacce a livello di rete, il che richiede di garantirne la sicurezza. In questo studio, proponiamo un sistema di rilevamento delle anomalie basato sull'apprendimento profondo che utilizza una rappresentazione 2D del traffico di rete per affrontare questo problema. In particolare, suggeriamo di utilizzare una rappresentazione sensibile al tempo del traffico in una matrice 2D e un autoencoder per modellare il comportamento normale del sistema e identificare anomalie. L'ipotesi alla base di questo approccio è che i campioni anomali non possano essere ricostruiti con precisione dal modello addestrato su dati normali. I risultati ottenuti dal metodo proposto sono incoraggianti e dimostrano la sua efficacia. Inoltre, questo lavoro fornisce una base per ulteriori ricerche in questa direzione.

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

CPS	Cyber Physical System(s)
AD	Anomaly Detection
DL	Deep Learning
AE	Auto Encoder
DoS	Denial of Service
PFA	Probability of False Attack

1

Introduction

1.1 Frame of reference

1.1.1 Cyber Physical Systems Security

What is a CPS

Cyber Physical System(s) (CPS) are complex systems composed of digital, analog, physical, and human components that work in conjunction with each other to perform a specific function or service. CPS-based frameworks are utilized in various domains, including emergency management, power systems, traffic management, and personalized healthcare, with successful results. As the use of CPSs continues to grow in our society, ensuring the availability, reliability, and security of their services is becoming increasingly crucial. The criticality of these services necessitates the adoption of robust security measures to protect against cyber threats, data breaches, and other malicious activities. The importance of CPS security has made it a critical research area with ongoing efforts to develop novel and effective security solutions to address the ever-evolving threats to these systems.

The importance of security

The security threats posed to Cyber Physical Systems (CPSs) are a significant concern, given that a successful attack could result in the disruption of operations or the denial of critical services to society. Furthermore, CPSs are often deployed in public environments, making them vulnerable to physical level attacks. They are also susceptible to network-level threats due to their communication functionality and application-level attacks.

Machine learning over traditional methods

To detect any discrepancies in CPSs from expected behavior, whether caused by system faults or cyber-attacks, numerous anomaly detection methods have been proposed in the literature. However, traditional approaches often require expert knowledge, making them challenging to implement. As a result, machine learning-based approaches have emerged as a promising alternative to traditional methods. These techniques leverage the power of deep learning algorithms to automatically detect anomalous traffic patterns and identify potential security threats. The adoption of these methods has significantly reduced the need for expert knowledge, making them easier to deploy and manage.

1.1.2 Anomaly Detection Methods

The literature offers two main categories of anomaly detection methods: signature-based and profile-based. The former relies on prior knowledge of the features of known anomalies, while the latter uses historical data to create a model of the system's normal behavior. Anomalies are then identified as significant deviations from the modeled behavior, which could be caused by malicious actions or genuine but unusual activities. Both approaches have their strengths and weaknesses. Profile-based techniques can detect new and unforeseen anomalies without relying on a specific model, while signature-based methods can only detect known anomalies. However, since only known anomalies are detected, the false alarm rate may be lower.

In summary, anomaly detection systems can be categorized based on the approach used for setting the threshold. Manual setting of alert thresholds by experts and automatic threshold setting using artificial intelligence are the two classes. Automatic systems have proven to be more effective, adaptive, and require less human intervention. In this work, the selection of the approach was based on the characteristics of anomalies, such as unknownness, heterogeneity, and rarity. Thus, a profile-based context-aware security framework using deep learning approaches is proposed to detect anomalies.

1.1.3 Our Proposal

It is worth emphasizing that while the proposed method is applied to the CPS case study in this work, it is not limited to this specific scenario. Rather, it can be employed in any general network monitoring scenario, where the goal is to identify anomalies in network traffic data.

In the realm of Anomaly Detection (AD), there are three predominant strategies that are commonly used to tackle this problem. These three approaches are as follows:

- Supervised Deep Learning (DL)
- DL methods for feature extraction
- Generic normality feature learning

The supervised DL approach necessitates the availability of labeled data, which may not always be feasible. Moreover, it may not be an ideal method for addressing AD directly since the distribution of normal and anomalous data is often heavily skewed. In addition, a supervised approach may fall short when it comes to identifying anomalous instances that were not encountered during training.

Generic normality feature learning involves learning representations of data instances by optimizing a generic feature learning objective function that is not specifically designed for anomaly detection. However, the benefit of this approach is that the learned representations are compelled to capture some underlying data regularities, thus enhancing the performance of AD algorithms. The most popular methods for generic feature learning include data reconstruction, generative models, predictability modeling, and self-supervised learning. These methods leverage the intrinsic properties of data and exploit them to learn effective representations.

In this paper, we aim to leverage the generalization ability of unsupervised deep learning for AD applied to real traffic data, represented as images. To achieve this goal, we perform data reconstruction using an Auto Encoder in a semi-supervised framework that exploits a time-sensitive 2D representation of network traffic. Only a few approaches that rely on a 2D representation to detect network traffic anomalies leverage DL. By employing unsupervised DL in a semi-supervised framework, we can take advantage of the high-level abstractions learned by the model to identify subtle differences between normal and anomalous network traffic. This approach is especially effective when dealing with complex and high-dimensional data such as network traffic data.

In summary, the proposed approach in this paper provides an effective way to leverage unsupervised DL for AD applied to real traffic data, represented as images. By exploiting the generalization ability of DL and the semi-supervised learning framework, our method can capture the underlying data regularities and enhance the performance of AD algorithms. Furthermore, our approach is unique in its use of a 2D representation of network traffic data, which allows for a more comprehensive and intuitive analysis of network traffic.

1.1.4 The time component

To implement the proposed anomaly detection (AD) algorithm, we started by constructing a 2D representation of network traffic data. Our approach was inspired by the method proposed in the [2] and aimed to develop an image that also incorporates the time component. We put special emphasis on designing an image that captures the dynamic behavior of the traffic over time while still preserving its spatial properties. This 2D representation was then used as input to an autoencoder, a type of neural network that can learn a compressed representation of the input data. The autoencoder was trained on the nominal background traffic data to generate a low-dimensional representation of the data that captures its essential features. This compressed representation serves as a reference for the AD algorithm.

1.1.5 How to detect anomalies

In the testing phase, we use the 2D representation of the test traffic data and pass it through the trained autoencoder. The autoencoder generates a reconstructed image, which is compared to the original test image. The difference between the two images is measured using a reconstruction error metric. If the reconstruction error exceeds a predefined threshold, we consider the test image to be anomalous. This approach allows us to detect anomalies in the traffic data based on their deviation from the nominal traffic behavior.

1.1.6 Summary

In summary, our proposed AD algorithm is based on a 2D representation of network traffic data that captures both its spatial and temporal properties. We use an autoencoder to learn a compressed representation of the nominal traffic behavior, and then use this representation to detect anomalous traffic based on the deviation from the learned nominal traffic behavior. Our approach is effective in identifying subtle anomalies that may not be apparent with traditional anomaly detection methods. Additionally, our approach can be applied to any general network monitoring scenario and is not limited to the CPS case study.

In short terms, the main contributions of this work are:

- The design of a different 2D traffic representation
- Improving the performance of the existing Auto Encoder (AE) approach through the new design
- The evaluation of the proposed approach and the comparison of its performances with respect to the previous work that uses a similar approach.

The rest of this document is structured as follows: Chapter 2 dives in the literature concerning context-based security and network anomaly detection systems based on multi-dimensional representations of data. Chapter 3 provides a description of the dataset used in this study and the time-sensitive matrix representation. Chapter 4 presents the method implementation, while Chapter 5 details the experimental evaluation. Lastly, Chapter 6 draws conclusions based on the findings of this study.

2

Related Work

This chapter provides an in-depth exploration into interconnected paths of research: Security using ML and autoencoders, Contextual Security Strategies, Signature Based Anomaly Detection Systems and Multi-Dimensional Traffic Data Representation Techniques for Enhancing Security.

2.1 Related security with ML and autoencoders

In their paper published in 2020 [4], Ganesh, Kumar, and Pattabiraman address the importance of anomaly detection in various fields, particularly in network security. Anomalies, which are significantly different data points, are often rare and result in a large class imbalance. Detecting anomalies has significant applications in fields such as cyber security and credit card fraud detection. In network security, it is crucial to identify rogue packets that could lead to network attacks and potentially impact system availability, resulting in financial losses due to website downtimes. The authors propose the use of autoencoders, a type of neural network, for anomaly detection.

Oligeri (2023) [10] present a comprehensive study on the effectiveness of AI-based solutions for the physical-layer authentication of Low-Earth Orbit (LEO) satellites in the context of wireless communications. The authors address the significant challenge of satellite radio transducers, which possess non-standard electronics designed for harsh conditions, as well as the unique attenuation and fading characteristics experienced by receivers due to the low bit-rate and high speed of LEO satellites. Through extensive measurements on the IRID-

IUM LEO satellites constellation, encompassing over 100 million I-Q samples collected during a 589-hour campaign, the authors demonstrate that Convolutional Neural Networks (CNN) and autoencoders, when properly calibrated, can successfully authenticate satellite transducers with accuracy ranging from 0.8 to 1, depending on prior assumptions.

Yu [18] address the critical security problem of state estimation in cyber-physical systems (CPS) and propose a novel attack detection method specifically targeting replay attacks. The authors introduce a model-free reinforcement learning-based framework for replay attack detection, which autonomously learns and recognizes evolving attacks with enhanced effectiveness. Recognizing that attackers can adapt their strategies based on defenders' actions, the authors further propose a defense strategy against the interaction between attackers and defenders, solved through optimization learning. The reinforcement learning technology employed in their analytical procedure can also be extended to other control applications.

Zebin et al. (2023) [19] address the security concerns surrounding Domain Name Service (DNS) traffic and the challenges posed by the DNS over HTTPS (DoH) protocol. They highlight that while DoH provides privacy and security benefits to internet users, it also hinders network administrators' ability to detect suspicious network traffic associated with malware and malicious tools. To address this issue, the authors propose an explainable AI solution using a novel machine learning framework for accurate detection and classification of DNS over HTTPS attacks.

2.2 Context and signature basis

2.2.1 Context-based Anomaly Detection Systems

Context-based anomaly detection, also known as behavioral or heuristic-based anomaly detection, is a type of anomaly detection that focuses on understanding the normal behavior of a system and detecting deviations from this behavior. This type of detection system monitors the behavior of users, systems, networks, or applications over time and establishes a baseline or profile of normal behavior. Any deviation from this normal profile is considered an anomaly and can trigger an alert. The advantage of context-based anomaly detection is that it can detect unknown or zero-day attacks since it doesn't rely on predefined signatures. However, it requires extensive training data to establish a reliable normal behavior baseline and can sometimes lead to a high number of false positives.

2.2.2 Signature-Based Anomaly Detection Systems

Signature-based anomaly detection, also known as misuse or knowledge-based detection, involves detecting anomalies based on predefined patterns or signatures of known attacks or malicious behavior. These systems use a database of known attack signatures and compare them with observed events or behavior to detect potential threats. The strength of signature-based detection lies in its ability to accurately detect known attacks with a low rate of false positives. However, it's limited in its ability to detect new or unknown attacks that do not match any existing signature in its database.

2.3 Contextual Security Works

The profound influence of contextual data in amplifying the security capabilities of Cyber-Physical Systems (CPSs) has been well-established in past research. For instance, Sylla et al. (2019) [12] introduced a context-aware security blueprint specifically designed for the Internet of Things (IoT) applications, wherein the selection of security protocols was made contingent on specific user contexts, such as their mobility patterns.

In a similar vein, Dsouza et al. (2019) [14] proposed an innovative context-aware biometric security framework that merges real-time data with context-driven information, such as geographical location, prevailing light conditions, and temporal data.

Alagar et al. (2018) [1] introduced an advanced context-sensitive role-based access control mechanism, tailored for healthcare IoT applications. Furthermore, Ehsani-Besheli and Zarandi (2018) [3] demonstrated the practical utility of context information in the detection of anomalies within communication patterns of embedded systems.

2.4 Multi-Dimensional Traffic Data Representation for Security

While the adoption of 2D or 3D data representation strategies for network anomaly detection remains relatively novel, a handful of pioneering solutions have emerged. One of the earliest instances of such an approach is the work of Kim et al. (2004) [5], who proposed a unique image-based visualization model for network traffic, utilizing source and destination IP addresses along with the destination port number to render traffic in a 3D space.

Continuing on this trajectory, Kim and Reddy (2005a, 2005b) [6] applied classical image processing techniques to the packet count data within the address domain to facilitate the analysis of traffic patterns.

Shifting gears towards a 2D representation, Nataraj et al. (2011) [9] proposed a 2D representation strategy for malware binaries, which was followed by Wang et al. (2017) [17] who employed an image-based representation of network traffic in combination with a Convolutional Neural Network (CNN) for classifying malware traffic. In a similar context, Taheri et al. (2018) [13] utilized a comparable representation strategy for detecting botnet activity within IoT environments.

In more recent developments, deep learning techniques have been employed by Vasan et al. (2020) [15] who introduced an ensemble method for malware classification, and Venkatraman et al. (2019) [16] who proposed a hybrid model for malware detection and classification. An innovative method to represent general time-series data as images was proposed by Zhang et al. (2019) [20]. Meanwhile, Mohammadpour et al. (2018) [8] leveraged a 2D structure and a CNN for network intrusion detection.

Lastly, the insightful study by Salehi and Rashidi "A Survey on Anomaly Detection in Evolving Data" [11] provided an extensive exploration of the world of adaptive models. They examined the potential of these models to adapt to the dynamic alterations in environmental characteristics and thereby enhance the detection of anomalies in 'evolving' data streams - an aspect of paramount significance given the common scarcity of label information in numerous real-world applications.

3

Dataset & 2D Representation

3.1 Dataset

3.1.1 Datset Description

We selected the UGR'16 dataset [7] to meet their requirements. The dataset consists of two subsets: a calibration subset, which contains real background traffic for training, and a test subset, which includes a combination of real background and controlled attack traffic for testing. The calibration subset was recorded over a period of 100 days, with two short gaps, while the test subset was recorded for approximately a month. The network infrastructure used in the dataset is illustrated in Figure 3 of the paper. Netflow probes were installed on the outgoing network interfaces of the border routers to capture incoming and outgoing traffic data.

The dataset provides information on flows, which are essentially records of network traffic between a source and destination IP address. Each flow includes various features, such as the timestamp of the end of the flow, its duration, the source and destination ports, the protocol used, and information on the packets and bytes exchanged. These features can be used as input to machine learning models for network intrusion detection.

Concerning the attacks, the following classes have been simulated:

- Denial of Service (DoS):
  - DoS11: one-to-one DoS where attacker A_1 attacks the victim V_{21};

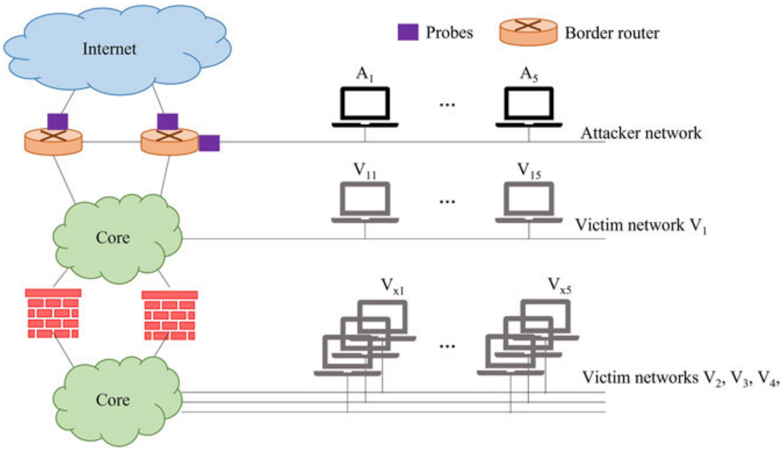


Figure 3.1: Dataset network structure

  - DoS53: the five attackers (A_1 - A_5) attack three victims. More specifically, attackers A_1 and A_2 attack the victim V21, attackers A_3 and A_4 attack the victim V_{31}, and attacker A_5 attacks the victim V_{41};

- Port Scanning:

  - Scan11: one-to-one scan attack where attacker A_1 scans the victim V_{41};
  - Scan44: four-to-four scan attack where the attackers A_1, A_2, A_3 and A_4 scan the victims V_{21}, V_{11}, V_{31} and V_{41}, respectively.

- Botnet: an attack involving all the twenty victim machines is simulated.

The attacks were scheduled in two different ways: a planned scheduling where there was no overlap between the attacks, and a random scheduling where overlap was possible. The calibration set consisted of real traffic, and even though no attack data was injected, real anomalies could be present. To address this, in [7], further classification was performed to distinguish between normal and anomalous background traffic in the calibration set.

3.1.2 Further Transformation Motivations

The sparsity of the images generated in the work from [2] poses a significant challenge for the effectiveness of an Auto Encoder approach. Although the images could differentiate between background traffic and under-attack traffic, the majority of the pixels in the images had a value of 0, making the images highly sparse. To address this issue, we aim to create a less sparse representation of the traffic data, while still preserving the key features highlighted by the previous transformation.

3.2 Time Sensitive Representation

Our approach to 2D representation builds on the method proposed in the [2], but with an emphasis on capturing a larger time window. While [2] generated images from only one second of data, our work combines multiple images to represent N seconds of past data. This approach ensures that the temporal information is preserved, allowing us to distinguish between data from the more distant past and the more recent past. By leveraging this temporal information, our 2D representation provides a more comprehensive view of the data, enabling us to gain deeper insights into the underlying patterns and trends.

To generate the results presented in the following section, we followed the same approach as that of [2] and used a fixed time window size of one second to generate our images. Our dataset comprises 377171 clean images and 9541 anomalous images, each with a size of 1x256x256.

Following image generation, we utilized a sliding window of N single-channel images to capture information on the source IP and the maximum amount of data exchanged from that IP in one second. Specifically, we identified the row index (from top to bottom) where the last non-zero value is present for each image and column. This information was then used to construct a one-dimensional vector for each image, containing information on the source IP and the maximum amount of data exchanged from that IP in a one-second time window.

By stacking N one-dimensional arrays, we constructed a 2D matrix that represents the traffic magnitude over a broader time window of N seconds of past data. This approach allows us to capture the temporal dependencies in the data and enables us to more accurately identify anomalies that may be present in the traffic patterns. With this approach, we are able to gain a more comprehensive understanding of the underlying traffic patterns, which is essential for effective anomaly detection.

We proceeded to generate images with a dimensional structure of 1x64x256. This size was found to be the most fitting for our analytical operations, striking a balance between detailed resolution and manageable computational complexity.

The technique we employed in our study was a subtle adaptation of the JET color mapping scheme. This highly effective approach allowed us to enhance the visibility of our images while maintaining the integrity of the darker pixels. In the context of the JET color mapping, we made sure that the black pixels, representative of particular network behaviors, retained their original state without any alteration. However, all other pixel values underwent a transformation to become more distinguishable, effectively enabling a more comprehensive analysis.

As a result of this application, our initial images transitioned into a more detailed configuration of 3x64x256, representing the Red, Green, and Blue (RGB) color spaces. This alteration did not only increase the depth of our images but also enriched the data available for subsequent analysis. Through this strategic use of color transformation, we were able to bring more clarity and detail to our investigation, enhancing our chances of accurately detecting and analyzing network anomalies.

3.3 Data Overview

Utilizing this methodology, we can illustrate various types of network traffic. For every images, we will keep two versions:

1. A greyscale version, immediately derived from the transformation described in detail in the following chapters.
2. A colored image, obtained by applying the JET color mapping to the greyscale images. The color map is slightly modified to retain the black pixels.

Figure 3.2 provides a clear representation of background traffic, while an example of a Denial of Service (DoS53) attack, rendered as an image utilizing our method, can be observed in Figure 3.3. Two key observations emerge from this comparative illustration:

- The first observation is that our proposed representation exhibits a greater density compared to the representation offered in [2]. This densification provides an advantage for the subsequent unsupervised learning stage, as a richer set of features can potentially lead to more accurate classification results.
- Our second observation stems from an anticipated variation between the two representations. Specifically, the background traffic image predominantly exhibits denser pixel regions towards the right side. In contrast, the anomaly representations - in this case, the DoS53 attack - appear denser

towards the left side. This distinction provides a clear visual demarcation between normal and anomalous network traffic, aiding in the detection process.

In-depth summaries of the pixel density across different image types can be viewed in Figure 3.4 (background) and Figure 3.5 (anomalies). These graphical representations offer a lucid visual understanding of the distribution patterns within each image type, thereby enabling a more comprehensive examination and interpretation of the network traffic. This detailed visual inspection can aid in building a more robust anomaly detection system, as it might lead to the discovery of unanticipated patterns or trends within the traffic data.

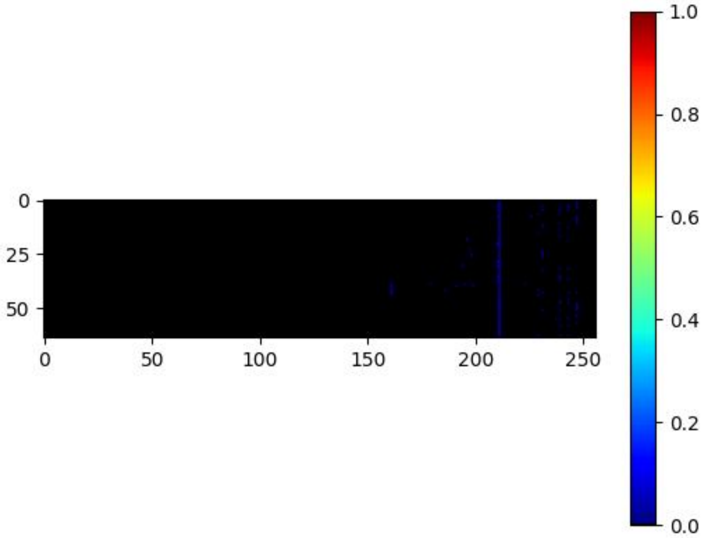


Figure 3.2: Example of background

Our objective in this research is to conduct an in-depth analysis of the images generated from background network traffic. This detection is primarily centered around confirming two key hypotheses:

- The first hypothesis asserts that the implementation of our proposed architectural design to the two-dimensional depiction, which is an extension derived from the representation presented in the prior work, could potentially yield a high true positive (TP) detection rate.

- Our second hypothesis also highlights its proficiency in correctly modeling the nominal traffic. This characteristic, in turn, contributes significantly towards reducing the false positives (FPs).

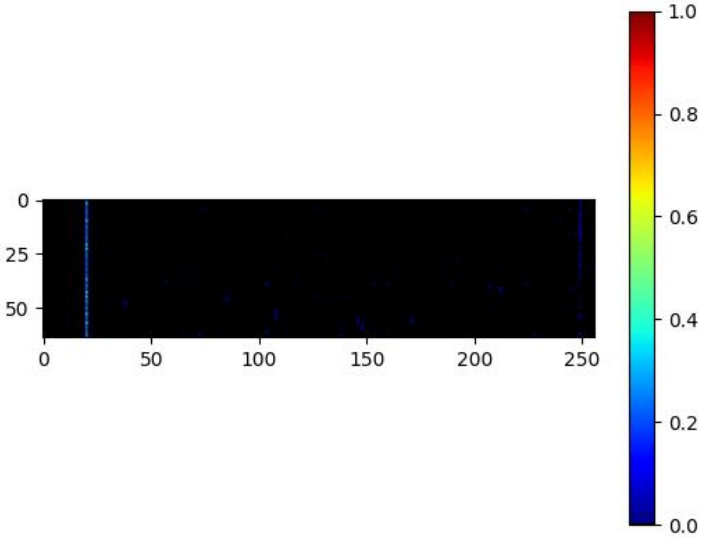


Figure 3.3: Example of DoS53

These hypotheses, if confirmed, could not only reinforce the robustness of our proposed method but also its versatility and efficiency in network traffic modeling and anomaly detection. With the promising potential of attaining a competitive TP detection rate and effectively limiting the FPs, our method might present a proficient AD system. This dual capability, balancing both high detection and low false alarms, underscores its potential impact and practical applications in improving network security measures.

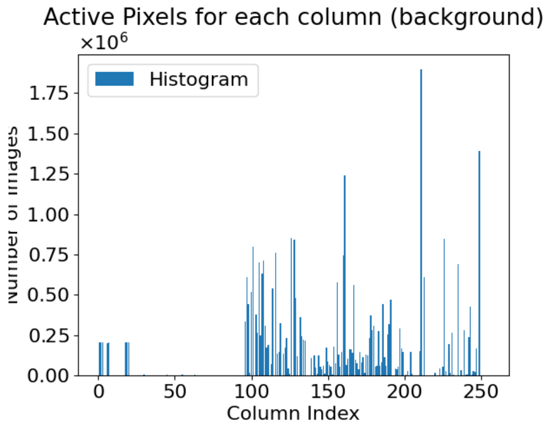
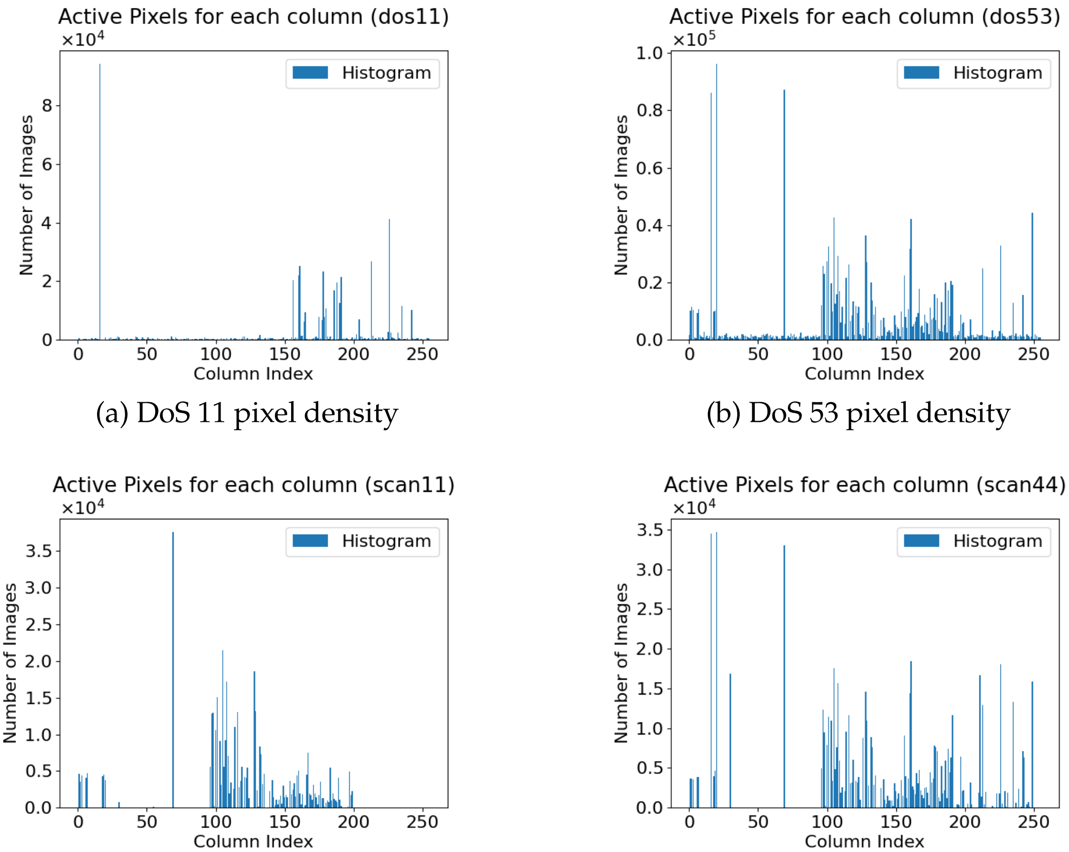


Figure 3.4: Background traffic pixel density



(a) DoS 11 pixel density

(b) DoS 53 pixel density

(c) Scan 11 pixel density

(d) Scan 44 pixel density

Figure 3.5: Anomalies histograms

4 Implementation & Strategy

4.1 Data transformation details

4.1.1 How the images are built

To describe the transformation of the data more formally, we assume the existence of matrices $\boldsymbol{M}_t$, which are derived from the process described in [2]. We aim to describe the process in a causal fashion, where $t = 0$ represents the present time (i.e., the most recent second).

Consider the last N seconds of data, comprising exactly N matrices with row index i and column index j. For each column c, we extract the highest row index where a non-zero value is present (returning 0 if none is found). We then create a vector $\boldsymbol{v}_t$ with 256 entries, denoted as $\boldsymbol{v}_{t_{0...255}}$. It is straightforward to calculate each value $\boldsymbol{v}_{t_j}$ using the algorithm shown in Algorithm 1. Here, $\boldsymbol{M}_{t_{i,j}}$ represents a single entry in a matrix. This process enables us to transform our original data into a format that captures temporal dependencies and is amenable to analysis using a range of techniques.

Repeating the process (alg. 1) in a matrix $\boldsymbol{M}_t$ for every coumn $j = 0 \ldots 255$ gives us the vector $\boldsymbol{v}_t$ associated with that matrix. Now we need N vector like $\boldsymbol{v}_t$, derived by varying the value of t from $t = 0$ to $t = -(N-1)$.

We can now define our last matrix $\boldsymbol{N}_t$, obtained by combining N vectors like $\boldsymbol{v}_t$ with $t \in (-(N-1), 0)$ as rows of $\boldsymbol{N}_t$, thus obtaining a 1xNx256 matrix where every row is a representation of 1 second of data and the whole matrix $\boldsymbol{N}_t$ an overview of the traffic in a time window of N seconds.

Algorithm 1 Values of v_t

```
Ensure: $v_{t_j} \leftarrow$ Highest row index where a non-zero value is present
  $t$ {Current matrix index}
  $j \leftarrow 0 \ldots 255$ {Column index}
  $v_{t_j} \leftarrow 0$
  $i \leftarrow 255$ {Highest possible value}
  while $i >= 0$ and $v_{t_j} = 0$ do
    if $M_{t_{i,j}} \neq 0$ then
      $v_{t_j} \leftarrow i$
    end if
    $i \leftarrow i - 1$
  end while
```

4.2 Deep learning model

In the research undertaken, the key model employed was an Auto Encoder. This particular model is composed of two primary modules: the encoder module and the decoder module. Each of these modules play a distinct and essential role within the overall framework of the model.

To delve a little deeper, the encoder module is assembled with a structure consisting of six convolutional layers: Conv1 by 16 filters, Conv2 by 32 filters and Conv3 by 64 filters, all having 2x2 stride and 5x5 kernels. The Adam optimizer was used.

Mirroring the encoder is the decoder module, exhibiting a symmetric structure. The functionality of the decoder module is inverse to that of the encoder. While the encoder is responsible for condensing input data into a compressed representation, the decoder's duty is to rebuild or regenerate the original data from this condensed form. The structure of the model is summerized in figure 4.1.

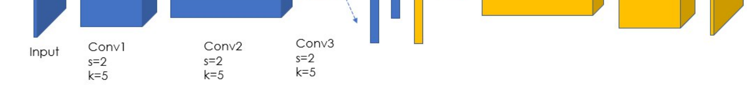


Figure 4.1: Dataset network structure

The primary objective in training the autoencoder (AE) model was to reduce as much as possible a critical measure known as the Mean Square Error (MSE).

By focusing on minimizing the MSE, the model aims to achieve the most accurate predictions possible. The goal is to make the output, the reconstructed data from the decoder module, as close as possible to the original input data fed into the encoder module. This reflects the fundamental purpose of the Autoencoder: to effectively compress and reconstruct data with minimal loss of information.

4.3 Strategy for anomaly detection

To put this into action, an examination is conducted on the reconstruction error. If this error surpasses a certain predetermined threshold, an anomaly is flagged. This means that if the difference between the original input and the reconstructed output is greater than what is deemed "acceptable" or "normal" (the threshold), the system identifies this as an unusual event or an anomaly.

Establishing this threshold is a key part of the process and involves a carefully designed approach. Initially, a validation dataset - a subset of the available data that the model hasn't been trained on - is passed through the model. During this process, the reconstruction errors for each piece of data in the validation set are computed and collected.

Once all the errors have been gathered, a suitable threshold is calculated. This calculation considers an acceptable Probability of false alarm (PFA), which essentially signifies the acceptable rate of false positives, or the instances where the system falsely identifies an event as an anomaly.

With the PFA in mind, a corresponding MSE threshold is chosen. This threshold will serve as the line of demarcation between normal and anomalous events. If a new input data generates a reconstruction error that exceeds this MSE threshold, it will be classified as an anomaly. This process underscores a balance between detection sensitivity and the false alarm rate, guiding the model in its task of effective anomaly detection.

4.4 Performance Evaluation

Finally, the evaluation of the Autoencoder (AE) model's performance was carried out using some metrics: Precision, Recall, F1 Score, and Accuracy. Each of these metrics provides a different perspective on the model's performance, offering a view of its effectiveness in identifying anomalies.

Among these, Precision and Recall stood out as particularly significant in this specific casedue to the unbalanced nature of the dataset used for training and testing the model. Unbalanced datasets are those where the instances of one class greatly outnumber the instances of another class. In such scenarios, Precision and Recall offer a more nuanced and insightful evaluation than Accuracy alone.

Precision measures the proportion of correctly identified positive cases from all cases that the model predicted as positive. In other words, it shows how precise the model is in predicting positive instances. High Precision indicates that the model correctly identified a high percentage of anomalies, while minimizing false positives.

Recall, on the other hand, measures the proportion of correctly identified positive cases from all actual positive cases. This metric tells us how well the model can find all the positive instances, or in this case, the anomalies. High Recall indicates that the model was able to detect most of the actual anomalies.

4.5 Technical setup

The computational tasks related to training the AE model were performed using an NVIDIA RTX 3070 GPU. Graphics Processing Units (GPUs) such as this one are highly advantageous for these tasks due to their ability to handle large amounts of data and perform numerous calculations simultaneously, accelerating the training process significantly.

When it comes to training a machine learning model, a key aspect involves setting the hyperparameters. These are the settings or the variables that govern the overall training process and directly impact the performance of the model. They are usually set before the training process begins and are not adjusted during the training itself.

```
config = {
        "batch_size": 16,
        "Nfc": 128,
        "latent_dim": 20,
        "max_pooling_kernel": 2,
        "lr": 0.0001,
        "wBCE": 15
    }
```

Code 4.1: Hyperparameters configuration

5 Experimental Results

The most noteworthy findings of this research are summarized in the following table. Remarkably, even with just a 0.10% Probability of false alarm (PFA), we are able to achieve impressive precision rates across different types of attacks.

	DOS11	**DOS53**	**SCAN11**	**SCAN44**	**Avg**
Precision	0.929	0.973	0.842	0.942	**0.922**
Recall	0.979	1.000	1.000	1.000	**0.994**

Table 5.1: Results with PFA 0.10%

One of the critical aspects that needs thorough analysis is the variation in the time window and the interplay between the time window and the Probability of false alarm (PFA). Moreover, evaluating the overall performance of the model, particularly in its capability to reconstruct images, is essential. The time window is an important factor, and it refers to the specific duration of data that is fed into the model at once. For instance, in time-series data or sequences, the time window determines how many data points or time steps the model considers simultaneously. Varying the size of this window can have substantial impacts on the model's ability to learn patterns and make predictions, especially when dealing with temporal data. It's necessary to find a balance where the time window is large enough to capture relevant data but not so large that it leads to an unacceptable number of false anomalies. Now, considering the Probability of false alarm (PFA), which we previously discussed as the acceptable rate of false positives, there's an intricate relationship between PFA and the time window.

Altering the time window might affect the rate at which false anomalies are detected.

By rigorously analyzing the impact of the time window, understanding its interplay with PFA, and evaluating the models performance in image reconstruction, valuable insights can be gained. These insights can be instrumental in fine-tuning the model for optimal performance and in understanding its strengths and limitations.

5.1 Training results

Let's start with the training overview.

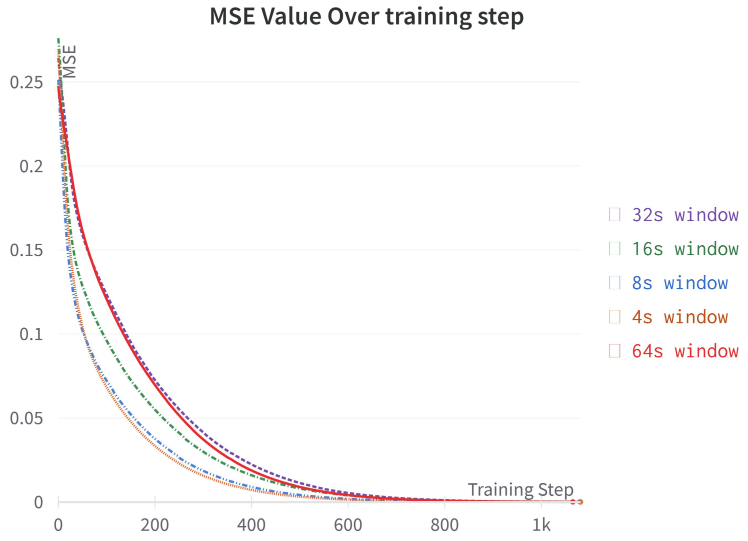
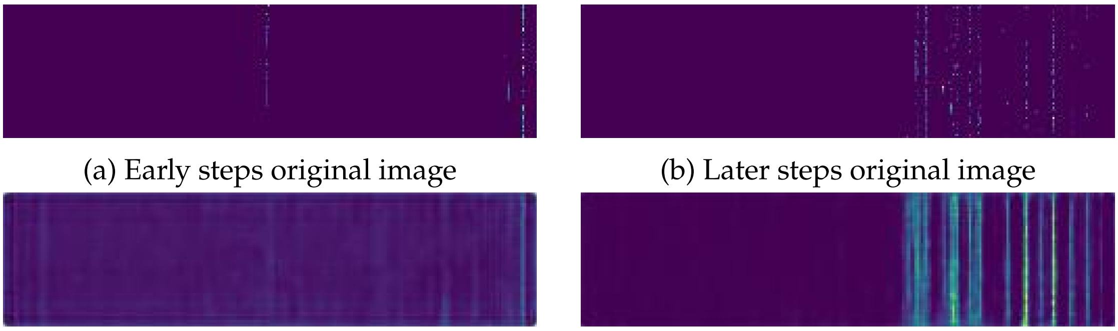


Figure 5.1: Traning: MSE/training step

As indicated in Figure referenced as 5.1, the training process exhibits a successful decrease in the Mean Square Error (MSE) between the input data and the reconstructed output across all the cases studied. The variation in the time window, ranging as integer powers of 2 from 4 seconds to 64 seconds, didn't hinder the training's effectiveness. This achievement reflects a successful completion of the model's training and lends confidence in assuming that the forthcoming test results would align with the MSE reduction goal.

A vital component of our evaluation involves a qualitative analysis that compares the original input image to the image reconstructed by the Autoencoder (AE) model. This side-by-side comparison in figure 5.2 essentially underlines



(a) Early steps original image

(b) Later steps original image

(c) Early steps reconstructed image

(d) Later steps reconstructed image

Figure 5.2: Later steps reconstructed image

how the model "perceives" and processes the data it's fed with:

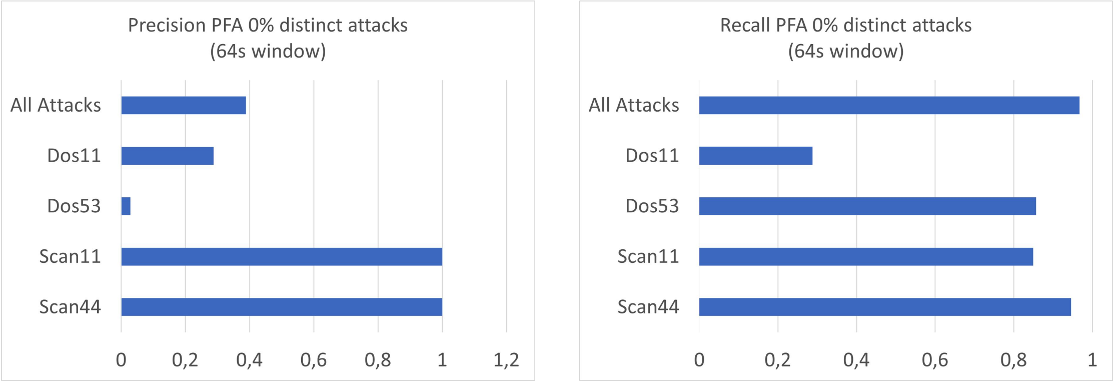
- Figure 5.2a reperesents an input image at the first steps of the training,
- Figure 5.2b reperesents an input image at the first steps of the training,
- Figure 5.2c reperesents an input image towards the end of the training,
- Figure 5.2d reperesents an input image towards the end of the training.

As the model becomes more adept at learning from the data through the training process, the reconstructed images it produces should increasingly resemble the original ones. This change is a significant indication of the model's improvement and its ability to effectively learn and capture the key characteristics of the data.

5.2 Tests results

5.2.1 General analysis of Precision & Recall

Now, with the successful completion of the training process and an acceptable decrease in MSE, it's an opportune time to delve deeper into the performance analysis, particularly focusing on the model's ability to recognize attacks. It's crucial to examine this aspect for each type of attack separately to gain a detailed understanding of how well the model performs in various scenarios. In this phase of the evaluation, the Probability of false alarm (PFA) is set to 0%, signifying a scenario where no errors are tolerated. This setting offers the most stringent evaluation conditions possible, wherein every anomaly flagged by the system is considered as an actual anomaly, and there is no allowance for false



(a) Precision distinct attacks PFA 0%

(b) Recall distinct attacks PFA 0%

Figure 5.3: Precision and recall on distinct attacks

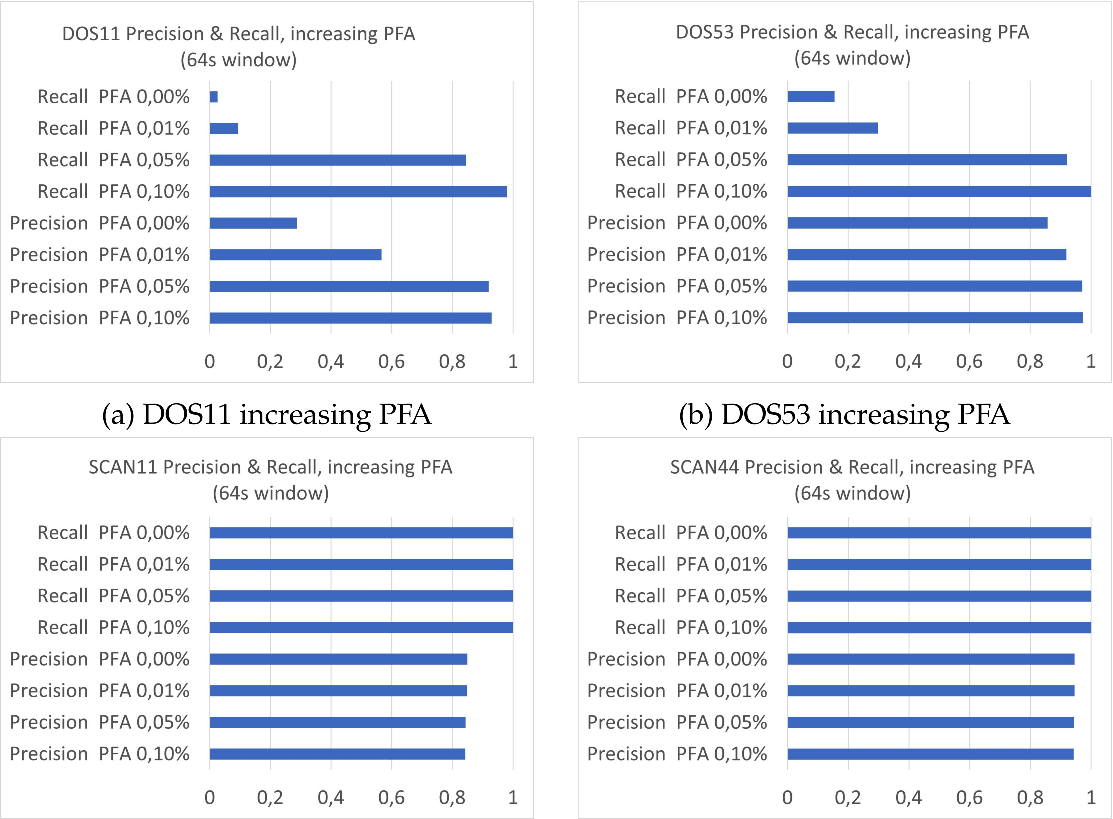
positives. While such an approach might seem extreme, it can be a valuable assessment tool. It essentially sets the bar high for the model, challenging it to be as precise as possible in its predictions. This strict evaluation will highlight the model's strengths, expose potential weaknesses, and provide clear direction for further tuning and improvement. Its in this rigorous testing that the model's true capability in recognizing attacks will be revealed.

Analyzing the data from Figure 5.3, it becomes evident that the model displays strong Precision yet relatively weak Recall, especially when dealing with Denial-of-Service (DOS) attacks. This means that while the model has a high success rate in accurately identifying attacks when it does flag one (as indicated by the high Precision in Figure 5.3a), it doesn't catch all the attacks that actually occur (as denoted by the lower Recall in Figure 5.3b).

5.2.2 PFA analysis

Moving forward, let's delve into how the Probability of false alarm (PFA) impacts the performance of our model. It's reasonable to anticipate that as we increase the PFA, effectively allowing for a certain degree of error, both Precision and Recall metrics will likely rise. This is because, by increasing the PFA, we're loosening the strict conditions under which an event is identified as an anomaly, which would typically result in identifying more true positives (hence increasing Recall) and reducing false negatives. However, our objective here isn't merely to maximize these metrics. We must also ensure that we maintain a high level of reliability in the model's performance. Balancing these aspects is key to creating a robust and effective anomaly detection system. The impacts of varying PFA levels can be observed in the series of figures referenced as 5.4.

- In figure 5.4a we can see that at 0.10% tolerance we already have almost perfect anomaly detection,
- In figure 5.4c and 5.4d we can see that varying the PFA doesn't affect the results of anomaly detection performance for the SCAN11 and SCAN44 attacks, which is already quite high.



(a) DOS11 increasing PFA

(b) DOS53 increasing PFA

(c) SCAN11 increasing PFA

(d) SCAN44 increasing PFA

Figure 5.4: Precision and recall on distinct attacks (lower bar means lower PFA)

5.2.3 Time window analysis

Now, let's examine how the time window and the Probability of false alarm (PFA) interact to influence the performance metrics.

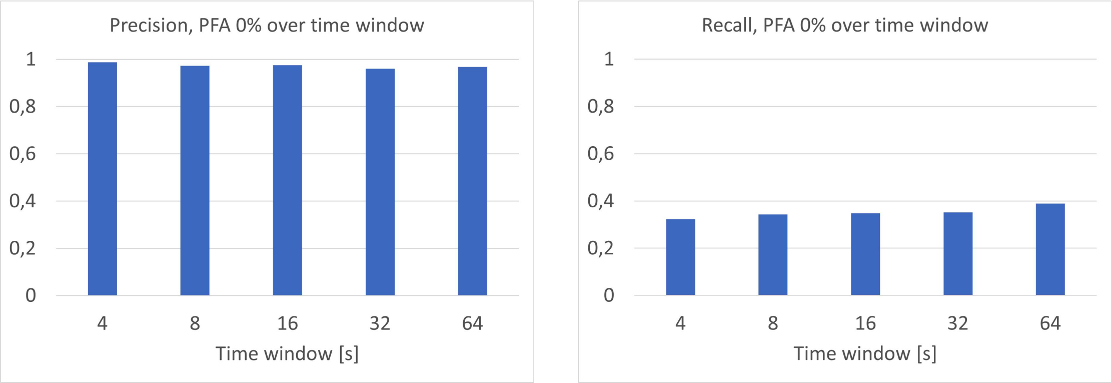
The findings from this examination present some fascinating insights into the workings of our system and how it reacts to changes in PFA and the time window.

When the PFA is set at 0% (as shown in figures 5.5), irrespective of the time window, the system retains high precision. This indicates that it continues to accurately identify threats. However, the recall is fairly low, around 0.4, suggesting that numerous threats manage to evade detection.

In scenarios where the PFA is set at 0.10% (figures 5.6), the system's precision remains nearly perfect in almost all cases, reaching peak performance with a 64-second time window. On the other hand, the recall shows a noticeable variation depending on the time window, increasing as the time window expands.

As the PFA rises to 0.30% (figures 5.7), the system maintains near-perfect precision and recall, with the notable exception of the 4-second time window. Despite the higher tolerance for errors, the system is less reliable in this shortest time window scenario. This is, however, an expected behavior considering that detecting an anomaly within such a brief timeframe can be quite challenging.

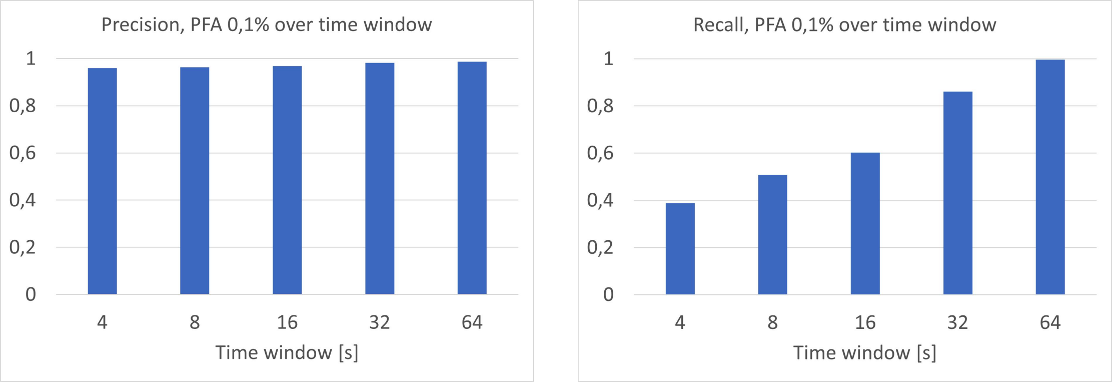
Finally, in the case of a 0.50% PFA (figures 5.8), included more for curiosity's sake and to confirm that the PFA indeed impacts the system as expected, both precision and recall are very high across all time windows. However, it's worth noting that an anomaly detection system that tolerates a 0.50% failure rate may not be considered practical or desirable for most applications.



(a) Precision over time window

(b) Recall over time window

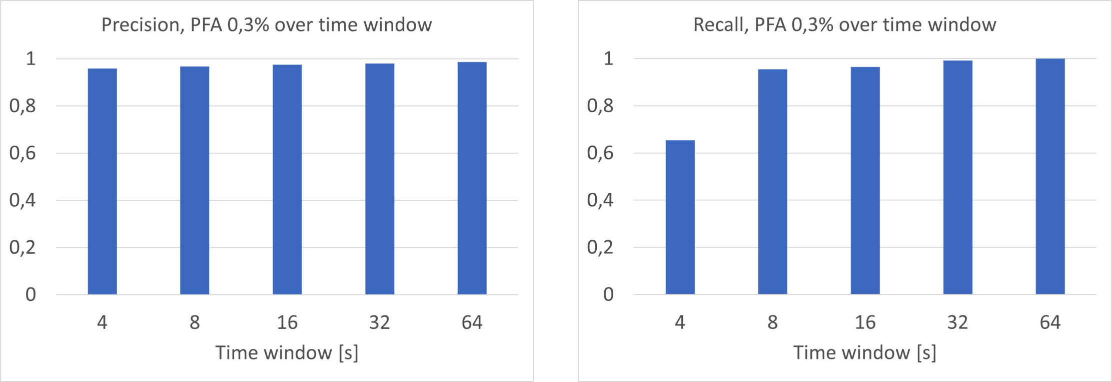
Figure 5.5: Precision and recall over time window at PFA 0%



(a) Precision over time window

(b) Recall over time window

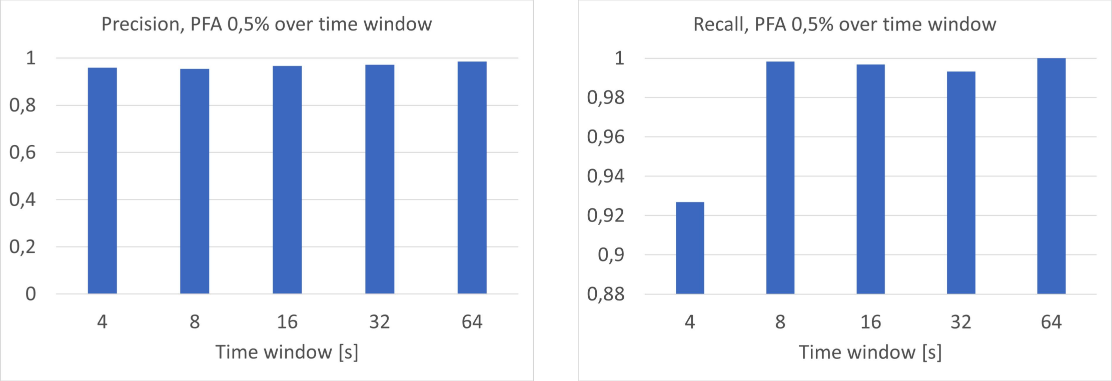
Figure 5.6: Precision and recall over time window at PFA 0.10%



(a) Precision over time window

(b) Recall over time window

Figure 5.7: Precision and recall over time window at PFA 0.30%



(a) Precision over time window

(b) Recall over time window

Figure 5.8: Precision and recall over time window at PFA 0.50%

6 Conclusions and Future Works

6.1 Conclusive thoughts

This research journey has provided numerous valuable insights into the workings of autoencoders in the context of anomaly detection. The designed model has exhibited the ability to learn and adapt to different scenarios effectively, allowing for insightful observations and findings to be garnered.

The model, comprising of an encoder module with 6 convolutional layers and a symmetrical decoder module, was trained to minimize the Mean Square Error (MSE) between the input and the output. The results indicated that the training was successful in all the cases studied, with time windows varying from 4 to 64 seconds.

Anomaly detection was achieved by examining the reconstruction error and comparing it to a determined threshold. The threshold was computed using a validation set and an acceptable Probability of False Anomaly (PFA), facilitating the discrimination between normal and anomalous data.

Various performance metrics were employed, including Precision, Recall, F1 Score, and Accuracy. Given the unbalanced nature of the dataset, Precision and Recall stood out as the most significant indicators of the model's performance.

Analysis of the model's performance under varying PFA conditions and time windows yielded intriguing results. In the strictest scenario, with 0% PFA, the system showed high precision but relatively low recall. However, as the PFA increased, allowing for some error margin, both the precision and recall saw improvement. This effect was also connected with the size of the time window, with larger windows generally leading to improved recall.

Qualitative analysis further improved our understanding of the model's performance, offering a visual comparison between the original and reconstructed images. Over time, the model showed a significant improvement in its ability to reconstruct original images, indicating effective learning.

However, the results also highlight that while the model has its strengths, it isn't without limitations. Despite high precision, the lower recall in certain scenarios, especially in the 0% PFA case, shows there is room for improvement. Moreover, the model struggled to perform reliably in the shortest time window (4 seconds), even with higher PFA, underscoring the challenge of detecting anomalies in brief time frames.

6.2 Future works

However, the exploration doesn't end here. The future directions for this line of work are vast and filled with exciting possibilities. Based on the results and observations made, the following areas have been identified for future research:

1. **Exploring Different Architectures**: The present model employs a specific architecture with six convolutional layers in both the encoder and decoder modules. Investigating different architectures, possibly integrating recurrent layers or attention mechanisms, might lead to improved performance or offer different insights.

2. **Refining Threshold Calculation**: The current method for determining the anomaly detection threshold involves running the validation set through the model and gathering all the errors. Future research could explore more refined or adaptive methods for calculating this threshold, leading to improved anomaly detection.

3. **Adapting to Short Time Windows**: The model displayed some difficulties in effectively detecting anomalies within the shortest time window of 4 seconds, even when the PFA was increased. Techniques that allow for efficient detection in such brief time frames could be an interesting avenue for future exploration.

4. **Testing with Different Datasets**: The model was trained and tested on a specific dataset in this study. Applying the model to different datasets, particularly those with varying characteristics, could reveal more about its generalizability and adaptability.

5. **Integration with Other Models**: VAE could potentially be combined with other machine learning models to create a hybrid system that capitalizes

on the strengths of different approaches. This could also be a way to mitigate some of the weaknesses observed in the current model.

In conclusion, while the current research provides significant findings and contributes to the field of anomaly detection using VAEs, there remain numerous avenues to be explored. The adventure continues, with these potential future works serving as the roadmap for the next phase of this exciting journey in machine learning and anomaly detection.

References

[1] Vangalur Alagar et al. “Context-Based Security and Privacy for Healthcare IoT”. In: *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*. 2018, pp. 122–128. DOI: 10.1109/SmartIoT.2018.00-14.

[2] Sofia Casarin et al. “Unsupervised Network Anomaly Detection by Learning on 2D Data Representations”. In: *2022 9th Swiss Conference on Data Science (SDS)*. 2022, pp. 53–58. DOI: 10.1109/SDS54800.2022.00016.

[3] Fatemeh Ehsani-Besheli and Hamid R. Zarandi. “Context-Aware Anomaly Detection in Embedded Systems”. In: *Advances in Dependability Engineering of Complex Systems*. Ed. by Wojciech Zamojski et al. Cham: Springer International Publishing, 2018, pp. 151–165. ISBN: 978-3-319-59415-6.

[4] Mukkesh Ganesh, Akshay Kumar, and V Pattabiraman. “Autoencoder Based Network Anomaly Detection”. In: *2020 IEEE International Conference on Technology, Engineering, Management for Societal impact using Marketing, Entrepreneurship and Talent (TEMSMET)*. 2020, pp. 1–6. DOI: 10.1109/TEMSMET51618.2020.9557464.

[5] Hyogon Kim, Inhye Kang, and Saewoong Bahk. “Real-time visualization of network attacks on high-speed links”. In: *IEEE Network* 18.5 (2004), pp. 30–39. DOI: 10.1109/MNET.2004.1337733.

[6] S.S. Kim and A.L.N. Reddy. “Modeling network traffic as images”. In: *IEEE International Conference on Communications, 2005. ICC 2005. 2005*. Vol. 1. 2005, 168–172 Vol. 1. DOI: 10.1109/ICC.2005.1494341.

[7] Gabriel Maciá-Fernández et al. “UGR16: A new dataset for the evaluation of cyclostationarity-based network IDSs”. In: *Computers & Security* 73 (2018), pp. 411–424. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2017.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0167404817302353.

[8] Leila Mohammadpour et al. "A convolutional neural network for network intrusion detection system". In: *Proceedings of the Asia-Pacific Advanced Network* 46.0 (2018), pp. 50–55.

[9] L. Nataraj et al. "Malware Images: Visualization and Automatic Classification". In: *Proceedings of the 8th International Symposium on Visualization for Cyber Security*. VizSec '11. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2011. ISBN: 9781450306799. DOI: 10.1145/2016904.2016908. URL: https://doi.org/10.1145/2016904.2016908.

[10] Gabriele Oligeri et al. "PAST-AI: Physical-Layer Authentication of Satellite Transmitters via Deep Learning". In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 274–289. DOI: 10.1109/TIFS.2022.3219287.

[11] Mahsa Salehi and Lida Rashidi. "A Survey on Anomaly Detection in Evolving Data [With Application to Forest Fire Risk Prediction]". In: *SIGKDD Explor. Newsl.* 20.1 (May 2018), pp. 13–23. ISSN: 1931-0145. DOI: 10.1145/3229329.3229332. URL: https://doi.org/10.1145/3229329.3229332.

[12] Tidiane Sylla et al. "Towards a Context-Aware Security and Privacy as a Service in the Internet of Things". In: *Information Security Theory and Practice*. Ed. by Maryline Laurent and Thanassis Giannetsos. Cham: Springer International Publishing, 2020, pp. 240–252. ISBN: 978-3-030-41702-4.

[13] Shayan Taheri, Milad Salem, and Jiann-Shiun Yuan. "Leveraging Image Representation of Network Traffic Data and Transfer Learning in Botnet Detection". In: *Big Data and Cognitive Computing* 2.4 (2018). ISSN: 2504-2289. DOI: 10.3390/bdcc2040037. URL: https://www.mdpi.com/2504-2289/2/4/37.

[14] Dan Tang et al. "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost". In: *Future Generation Computer Systems* 106 (2020), pp. 347–359. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2019.12.034. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19310544.

[15] Danish Vasan et al. "Image-Based malware classification using ensemble of CNN architectures (IMCEC)". In: *Computers & Security* 92 (2020), p. 101748. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2020.

101748. URL: https://www.sciencedirect.com/science/article/pii/S016740482030033X.

[16] Sitalakshmi Venkatraman, Mamoun Alazab, and R. Vinayakumar. "A hybrid deep learning image-based analysis for effective malware detection". In: *Journal of Information Security and Applications* 47 (2019), pp. 377–389. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2019.06.006. URL: https://www.sciencedirect.com/science/article/pii/S2214212618304563.

[17] Wei Wang et al. "Malware traffic classification using convolutional neural network for representation learning". In: *2017 International Conference on Information Networking (ICOIN)*. 2017, pp. 712–717. DOI: 10.1109/ICOIN.2017.7899588.

[18] Yan Yu et al. "Reinforcement Learning Solution for Cyber-Physical Systems Security Against Replay Attacks". In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 2583–2595. DOI: 10.1109/TIFS.2023.3268532.

[19] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo. "An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks". In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 2339–2349. DOI: 10.1109/TIFS.2022.3183390.

[20] Chuxu Zhang et al. "A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 1409–1416. DOI: 10.1609/aaai.v33i01.33011409. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3942.