REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DOMAIN FREE DEEP LEARNING BASED SECURITY MODELS FOR CYBERPHYSICAL SYSTEMS

Dilara GÜMÜŞBAŞ

DOCTOR OF PHILOSOPHY THESIS
Department of Electronics and Communications Engineering
Electronics Program

Advisor
Prof. Dr. Tülay YILDIRIM

July, 2020

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DOMAIN FREE DEEP LEARNING BASED SECURITY MODELS FOR CYBERPHYSICAL SYSTEMS

A thesis submitted by Dilara GÜMÜŞBAŞ in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 29.07.2020 in Department of Electronics and Communications Engineering, Electronics Program.

Prof. Dr. Tülay YILDIRIM
Yıldız Technical University
Advisor

Approved By the Examining Committee

Prof. Dr. Tülay YILDIRIM, Advisor
Yıldız Technical University _____

Prof. Dr. Zümray DOKUR ÖLMEZ, Member
Istanbul Technical University _____

Assist. Prof. Dr. Nihan KAHRAMAN, Member
Yıldız Technical University _____

Prof. Dr. Mehmet Timur AYDEMİR, Member
Gazi University _____

Assist. Prof. Dr. Sadiye Nergis TURAL POLAT, Member
Yıldız Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Domain Free Deep Learning Based Security Models for Cyberphysical Systems supervised by my supervisor, Prof. Dr. Tülay YILDIRIM. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Dilara GÜMÜŞBAŞ

Signature

Dedicated to my family
and two music bands:
Camel and Pink Floyd

ACKNOWLEDGEMENTS

There are many people I would like to thank for their contribution to my four-years-adventure as a Ph.D. candidate. First, I would like to thank my supervisor Prof. Dr. Tülay YILDIRIM who answered my endless last-minute emails/calls and guided me all the time with passion and patience. Her knowledge and foresight made my endless ambitious raw goals form better. I am very thankful to her not only for motivating me to find a way to work on the next steps when I am stuck on/confused about my ideas but also for showing me the ways how to think and look at a problem from different perspectives.

I also would like to thank members of my thesis monitoring committee, Prof. Dr. Zümray DOKUR ÖLMEZ and Assist. Prof. Dr. Nihan KAHRAMAN, for making my Ph.D. journey better with their valuable recommendations that gave me huge help to improve my work.

During my Ph.D. adventure, I got a YÖK-YUDAB scholarship and was fortunate enough to do a wonderful internship with great researchers in the IEBI Lab at the University of Milan. I learned a lot from their unique perspectives to enrich my ideas. From this internship, I would like to express my gratefulness especially to Prof. Dr. Vincenzo PIURI, Prof. Dr. Fabio SCOTTI, Assist. Prof. Dr. Ruggero DONIDA LABATI and Assist. Prof. Dr. Angelo GENOVESE for their insightful comments and supervision.

When I look back to my bachelor's degree years, I was extremely lucky to have a great lecturer who motivates and encourages all his students with a positive attitude. Thanks to him, I could challenge myself and start my MSc journey at the University of Southampton. Therefore, I would like to thank Prof. Dr. Mehmet Timur AYDEMİR for making my academic journey started.

Last but not the least, I would like to extend my gratitude towards my beloved family. Without their love and support, I could not be courageous enough to take the challenge and move forward all the time.

Dilara GÜMÜŞBAŞ

TABLE OF CONTENTS

LIST OF SYMBOLS

T_c	Adaptable constant: 1 when the class is existing
λ	Down-weighting constant for missing classes
s_j	Input of a Capsule
L	Margin Loss for each class in DigitCaps
v_j	Output of a Capsule

LIST OF ABBREVIATIONS

AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
CFS	Correlation Feature Selection
CNN	Convolutional Neural Networks
CPS	Cyberphysical Systems
DBN	Deep Belief Network
DL	Deep Learning
DNN	Deep Neural Network
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
IG	Information Gain
IoT	Internet of Things
k-NN	k-nearest neighbors
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NIDS	Network Intrusion Detection System
NN	Neural Network
PCA	Principal Component Analysis

PSO	Particle Swarm Optimization
RBM	Restricted Boltzmann Machine
RF	Random Forest
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SMOTE	Synthetic Minority Over-sampling Technique
SOM	Self Organizing Map
SVM	Support Vector Machine
TL	Transfer Learning
VAE	Variational AutoEncoder
WD	Writer-dependent
WI	Writer-independent

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Domain Free Deep Learning Based Security Models for Cyberphysical Systems

Dilara GÜMÜŞBAŞ

Department of Electronics and Communications Engineering
Doctor of Philosophy Thesis

Advisor: Prof. Dr. Tülay YILDIRIM

With the developments in digital age and growing interest in IoT, a variety of institutions and organizations have started to digitalize their systems. As a consequence of these digitalizations, security of collecting, accessing and transferring great amounts of private data via internet connection have became an important issue. In particular, protection of data collected, transmitted and stored on cyberphysical systems (CPS) such as security systems have gained great importance.

Recently, many studies have been conducted using state-of-the-art Deep Learning (DL) algorithms for security systems. However, despite their groundbreaking results, most of these studies either are biased to some particular datasets or too complex and computationally-expensive to be used in real time. Moreover, DL algorithms require a lot of input data to extract the most informative feature representations and become disadvantageous in real situations, where imbalances among classes and unlabelled samples in input data are quite common. Therefore, first goal of this dissertation is to conduct a comprehensive research and to study AI-based new approaches for two different domains of security-themed systems: biometric systems and cybersecurity. In particular, new Capsule-based feature representations for these domains are investigated in detail and these representations are compared with their equivalent state-of-the-art algorithm-based models for the first time.

Second goal is to conduct an experiment on Transfer Learning (TL) for cybersecurity, where features are in time-domain and benchmark datasets do not share sufficient common feature space with each other like image-domain counterparts such as

biometric systems to use pre-trained network in 1D. In addition, possible scenarios are examined to adapt security systems into different domains and generalize by using available benchmark datasets with different traffic collection as well as feature spaces.

Keywords: Deep learning, capsule networks, network intrusion detection, biometric identification and verification, cyberphysical systems

ÖZET

Siberfiziksel Sistemler için Alan Bağımsız Derin Öğrenme Tabanlı Güvenlik Modelleri

Dilara GÜMÜŞBAŞ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Prof. Dr. Tülay YILDIRIM

Dijital çağdaki gelişmeler ve IoT'a yönelik artan ilgiyle, çeşitli kurum ve kuruluşlar kendi sistemlerini dijitalleştirmeye başlamışlardır. Bu dijitalleştirmelerin sonucunda, büyük miktarlardaki kişisel verilerin internet yoluyla toplanması, ulaşılması ve iletilmesi önemli bir konu haline gelmiştir. Özellikle de, güvenlik sistemleri gibi siberfiziksel sistemler (SFS) üzerinde toplanan, iletilen ve kaydedilen verilerin güvenliği çok büyük önem kazanmıştır.

Son zamanlarda, güvenlik sistemleri için yapılan çoğu çalışma en gelişkin Derin Öğrenme (DÖ) metodları kullanılarak gerçeklenmiştir. Lakin, çığır açıcı sonuçlara rağmen, bu çalışmaların çoğu ya belirli veri setlerine meyilli olacak şekilde öğrenmiştir ya da gerçek zamanlı kullanmak için çok karışık ve hesapsal yükü fazladır. Dahası, DÖ algoritmaları giriş verisinden en anlamlı özellikleri çıkarabilmek için çok sayıda giriş veri örneğine ihtiyaç duyar bunun sonucunda sınıflar arası verinin dengesiz olduğu yada sınıflanmamış veri örneklerinin olduğu gerçek durumlar için dezavantaj haline gelir. Bu sebeple, bu tezin ilk amacı kapsamlı bir araştırma yürütmek ve AI-tabanlı yeni yaklaşımları güvenlik temalı sistemlerdeki iki farklı çalışma alanı -biyometrik sistemler ve siber güvenlik sistemleri- için analiz etmektir. Özellikle de, yeni kapsül-tabanlı elde edilmiş özellikler ayrıntılı bir şekilde incelenmiş ve bu özellikler ilk kez bu iki çalışma alanı için eşdeğer en-gelişkin algoritma tabanlı özellikler ile kıyaslanmıştır.

İkinci amaç ise özelliklerin görüntü uzayında değil, zaman uzayında olduğu ve referans veri setlerinin görüntü veri setleri gibi birbiriyle yeterli miktarda ortak özellik uzayının olmadığı siber güvenlik sistemleri için Transfer Öğrenme (TÖ) üzerine

çalışma yapmaktır. Buna ek olarak, güvenlik sistemlerini farklı özellik uzaylarına uyarlayabilmek ve varolan farklı şekilde toplanmış ve ayrı özellik uzaylarına sahip referans veri setlerini kullanarak genelleştirebilmek için muhtemel senaryolar incelenmiştir.

Anahtar Kelimeler: Derin öğrenme, kapsül ağları, ağ saldırı tespiti, biyometrik kimlik tanımlama ve doğrulama, siber fiziksel sistemler

1
INTRODUCTION

Cyberphysical system (CPS) is an interdisciplinary system that combines computer-based systems with physical systems to operate together. Some of the most known examples of CPS are biometric monitoring, smart grid systems, cybernetics and autonomous cars. With the increasing interest of Internet of Things (IoT), a type of CPS with less complexity regarding coordination, data collection, transmission and storage has gained importance.

In this study, we consider security-themed approaches for IoT and employ cybersecurity and biometric systems to conduct realistic research on domain-free Artificial Intelligence (AI) methods for CPS. Here, cybersecurity is used to secure the process of data transmission and storage of IoT. Similarly, biometric systems are chosen to secure the process of data collection for IoT since it is done via physical components such as keyboards, sensors for security systems and passwords or several biometric traits are frequently used to secure the process of data collection.

Throughout this chapter, AI methods for CPS are discussed with a specific focus on two different types of security-themed approaches, namely biometric systems and cybersecurity. These two systems are chosen for two reasons. Firstly, AI-based algorithms vary from time-series models to computer vision models. Thus, using such security systems that are in different domains such as image and time domains is important to obtain a comprehensive study. Besides, we employ two different types of biometric systems, behavioral-based and physical-based systems, to widen the scope of the study. Secondly, domain-free AI security models and Transfer Learning (TL) can be experimented on these security-themed approaches since both show different characteristics and experiments could give an idea about how dataset bias affects those models.

1.1 Literature Review

This section presents AI methods for cybersecurity and biometric systems. These AI methods are chosen according to two criteria: their contribution as a pioneer to the literature and/or being recently-published with high citation statistics. Furthermore, advantages and disadvantages of each AI algorithm type are only explained in detail.

1.1.1 AI Methods for Cybersecurity

With the increasing pace of developments in the digital age, accessing and transferring great amounts of data via internet connection and evolving cyber threats, cybersecurity-related issues have been increased. Therefore, these issues created the need to deploy more trustable cybersecurity systems, which are composed of a variety of preventive methods.

As one of the widely-studied branches of cybersecurity systems, Intrusion Detection System (IDS) is developed to detect cyber threats and to ensure safe user access and privacy protection. IDS primarily gathers data and makes a detection system into work to catch and identify possible threats for the use of security analysts. Besides, it can be categorized into two systems: Network Intrusion Detection System (NIDS) and Host-based Intrusion Detection System (HIDS). While NIDS is based on network traffic data that consists of whole interaction among devices on a network, HIDS is based on HIDS agent data collected from only host devices such as operating system logs.

A variety of algorithms are used for IDS which can be observed under three categories: rule-based, statistics-based and Machine Learning (ML) based algorithms. While rule-based algorithms use data distributions to construct a rule and execute it, statistics-based algorithms benefit from previous attack patterns to estimate a statistical distribution and employ this distribution to detect attacks. The last category falls under ML-based algorithms as a sub-field of AI, which refers to machines that mimic human cognitive abilities which varies from perception to problem-solving. ML-based algorithms concentrate on the learning part of these cognitive abilities. After learning, they perform classifier training to detect anomalies including known attacks. Since each algorithm has its advantages and disadvantages thereby, choosing the best one depends on the problem and the trade-offs. For example, rule-based algorithms can be performed simply and quickly. However, it cannot perform well under missing and/or imprecise data. Moreover, updating this approach is cumbersome. Similarly, statistics-based algorithms solve these problems but as a trade-off, they demand high computational power and they are not suitable for large amounts of data. Unlike

rule-based and statistics-based algorithms, ML-based algorithms are proposed to solve these problems using inference models which can capture the complexity and can be trained on big data.

As many organizations have started to employ interconnected systems, the amount of data collected and transferred over a network has been growing gradually. Therefore, the protection of the data coming from these systems has become even more vulnerable to not only unauthorized access but also authorized access by the insider attackers. Moreover, there may be a lack of human force to protect these systems in real-time. To solve these issues, ML-based approaches, in particular Deep Learning (DL) methods, are frequently used in cybersecurity for three main reasons. Firstly, these approaches are successful to find underlying patterns of data not only for known but also for novel attacks to automate threat and anomaly-based security monitoring and detection. Secondly, ML-based approaches are good at reducing false positives and lessen the number of false alarms to be analyzed by security analysts. Therefore, they facilitate the process for security analysts and increase their productivity regarding intrusion detection and response time. As a result, reducing the amount of data to be investigated makes a huge contribution to real-time detection and avoids data losses. Thirdly, they make the monitoring and detection systems computationally inexpensive and adaptable to update towards evolving attack types. Furthermore, ML-based approaches are able to predict anomalies.

This section not only presents an all-inclusive overview of ML approaches for cybersecurity by analyzing concerning evaluation results and limitations but also a further investigation on factors that affect reliability and scalability of these approaches are provided for potential future directions. Besides, taxonomy for ML-based approaches can be found in Figure 1.1.

1.1.1.1 k-means Clustering

K-means Clustering is a method where each input data is grouped to a randomly chosen clusters (k) according to their distance to these cluster centers and cluster centers are updated until certain criterion such as minimizing distances among clusters is met [1]. Although this algorithm is easy to implement, fast and computationally inexpensive for big data, there are several issues related to the k-means clustering. Firstly, choosing the optimal number of clusters is difficult. Secondly, noise in input data has a strong effect on performance results. Lastly, k-means clustering is negatively affected when different classes create the same cluster due to their same mean values or data is non-convex.

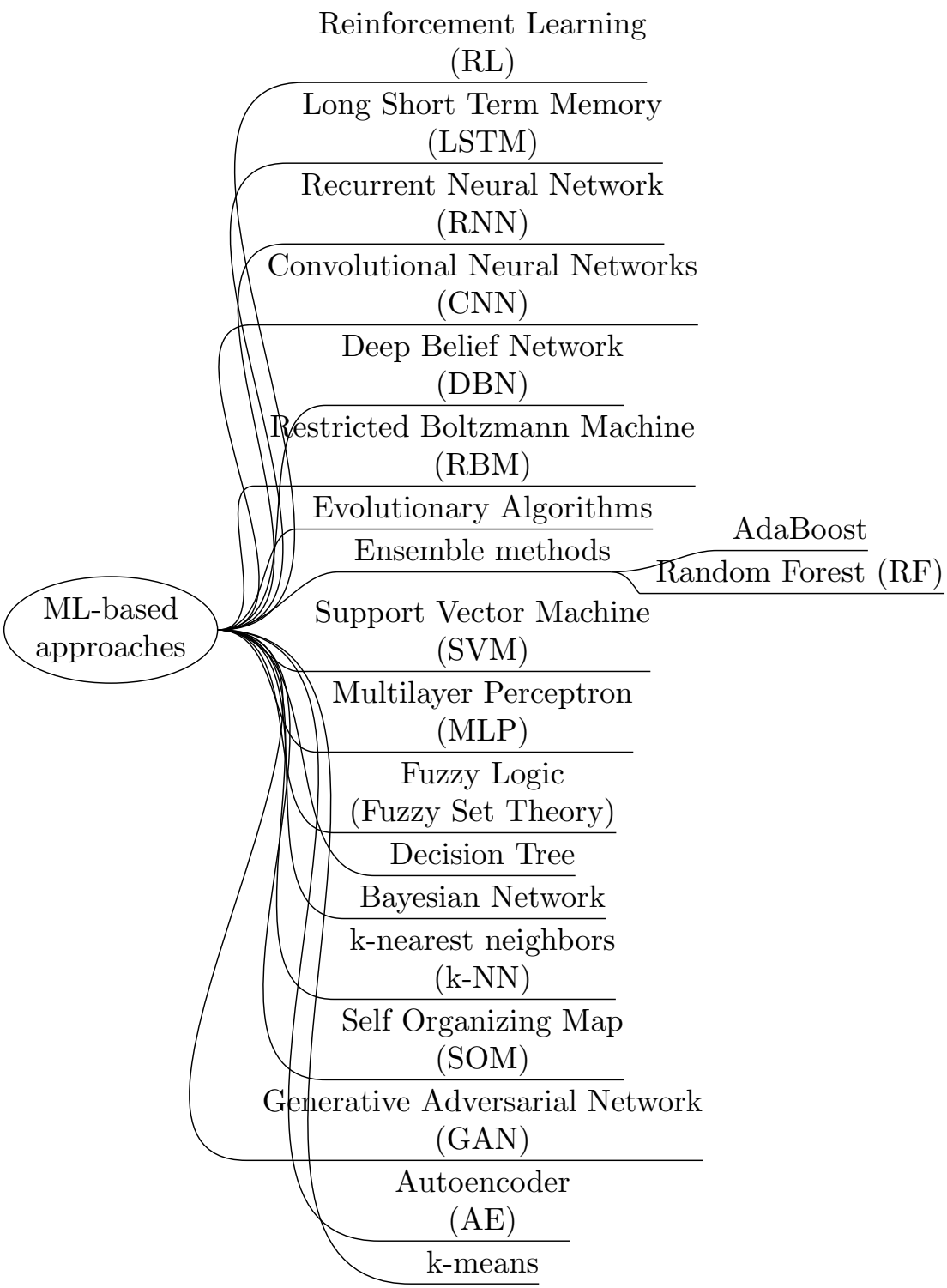

Figure 1.1 Taxonomy of ML-based algorithms for cybersecurity

Several methods have been proposed using different distance metrics. For instance, the method introduced in [2] uses System Call Frequency Distribution (SCFD) to calculate similarity metrics with k-means, where the cut-off distance of clusters is calculated using cumulative distribution function (CDF) with Mahalanobis distance to differentiate normal from attacks. The method achieves better detection than Euclidean distance on outliers on the private dataset while it may not detect local variations in system call sequences. Similarly, the method introduced in [3] employs k-means with Gaussian Similarity Measure on DARPA98 dataset.

To achieve high detection rates, several approaches have employed hybrid methods. For example, the model proposed in [4] first employs k-means to obtain different training subsets then uses five Fuzzy Neural Networks (NN). As a final step of the model, it classifies with SVM. The model achieves high detection results on KDD99 for each attack type. Similar to [4], the model proposed in [5] selects the most distinctive data samples with k-means then classifies these samples with NN. However, the model achieves low detection rates for minority classes: R2L and U2R in KDD99. The method introduced in [6] uses k-means as a first step of separating data into clusters then learns subgroups in clusters with C4.5 decision tree while the method proposed in [7] does same with Naive Bayes. Both methods achieves a high true positive rate with low false positives on KDD99 dataset.

1.1.1.2 Autoencoder (AE)

Autoencoder (AE) is a type of unsupervised DL method that first encodes then decodes the original data to build novel representations of the data, which can be seen in Figure 1.2. While the encoding layers make data representations into lower dimensions to find the most informative feature space, decoding layers samples from this space to original feature space under an unsupervised fashion [8]. All weights are optimized by minimizing reconstruction error. AE has generally been used for dimension reduction in cybersecurity thanks to its capability of extracting informative feature representations. However, choosing the optimal structure of encoding and decoding layers is difficult.

Several works are published for two different combinations of AE: AE with shallow/deep ML algorithms and AE with statistical algorithms or statistics-driven AE models such as Variational AE (VAE) with shallow ML algorithms. The studies proposed for the first combination in [9, 10] use AE for dimension reduction/nonlinear feature extraction and combine it with several shallow classifiers such as SVM on NSL-KDD dataset. It is reported that combinations with AE achieve higher accuracy compared to the combinations with other dimension reduction methods. Furthermore,

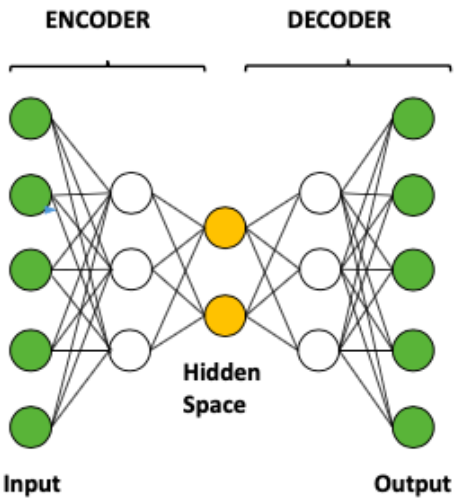

Figure 1.2 An example of Autoencoder

the study presented in [11] employs AE with a softmax regression classifier on the same dataset and reports higher accuracy than the previous ones. In addition to 1-hidden-layered AE, multi-layered versions, also known as stacked AEs, are used in the works introduced in [12–14]. These works combine stacked AEs with shallow classifiers. While the first two uses random forest on NSL-KDD and KDD99 datasets, the latter uses a radial basis function to achieve high overall accuracy on AWID2018 dataset. Similarly, the model introduced in [15] uses stacked AEs to extract valuable information from raw traffic data and automate the intrusion detection process. Besides, the works proposed in [16, 17] first use AE to extract meaningful information from raw network traffic then detects anomalies with CNN.

The methods proposed for the second combination of AE in [18, 19] use VAE to reduce dimension for raw network traffic and several featured datasets named NSL-KDD and UNSW-NB15, respectively. In the second work, several shallow algorithms such as random forest are also used to detect anomalies using the output of VAE. In a similar manner, the work introduced in [20] employs VAE with gradient-based linear SVM to detect some particular attacks on AWID2019 dataset, where SVM first reduces feature dimension then VAE selects the most relevant features. It is reported that the detection rate is higher than state-of-the-art models. In addition to models in [18–20], the model introduced in [21] combines VAE with GAN and DNN. Basically, it uses VAE to obtain new input representations formed in a statistical and nonlinear way then GAN to augment less-represented intrusions. Finally, the model uses DNN to classify unknown intrusions as well as known ones.

Furthermore, several AE combinations with statistical algorithms in [22, 23] adopt AE to extract nonlinear representations, then use density estimation on NSL-KDD and

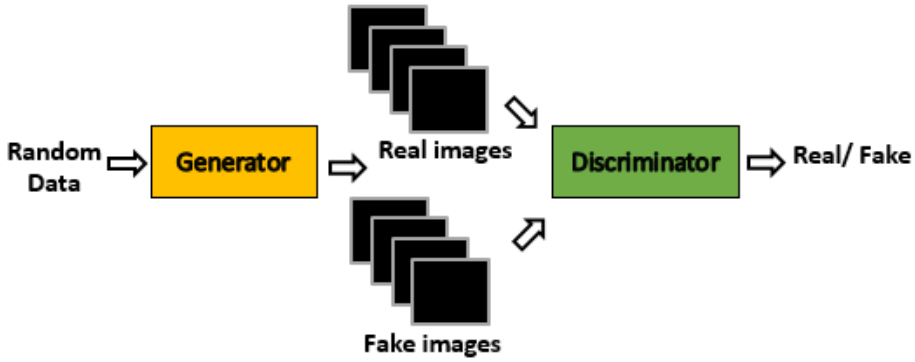

Figure 1.3 An example of Generative Adversarial Network

Gaussian Mixture Model (GMM) on KDD99, respectively. The results indicate that AE with statistical algorithms improves detection rates, especially for frequency-related intrusions. Similarly, the work presented in [24] combines AE with statistical models and achieves higher accuracy than state-of-the-art deep and shallow ML on NSL-KDD dataset.

1.1.1.3 Generative Adversarial Network (GAN)

GANs are one of the DL algorithms that consists of an encoder, a generator and a discriminator. As can be seen in Figure 1.3, the encoder first extracts statistical information from the input, then the generator creates new samples using the information and discriminator tries to differentiate original input from created ones [25]. The training of generator and discriminator is frequently done to minimize loss of the generator while maximizing loss of the discriminator. GANs have a great advantage of not only classifying but also augmenting new data samples only using statistical characteristics of input, in particular for minority classes in a dataset. Therefore, it has gained great interest in cybersecurity applications.

Several studies are done on data augmentation for cybersecurity datasets which have frequently imbalanced data samples among classes. The work proposed in [26] uses sequence GANs to augment ADFA-LD dataset. Similarly, the studies introduced in [21, 27] employ GANs to augment raw network traffic data. Both report an improvement in detection results. Similarly, another method proposed in [28] uses Flow Wasserstein GANs to generate adversarial data samples then employs them to detect and model anomalies better for cybersecurity. The evaluation is conducted on ISCX-2012 and ISCX-2017 datasets and an outperforming detection rate is reported.

In addition to the use of GANs for data augmentation, GANs are used for classification of anomalies. The study proposed in [29] uses GANs for anomaly detection. Similarly,

the work proposed in [30] employs GANs with some modifications to achieve an improvement in time. Both reported high and similar detection rates on KDD99 dataset.

1.1.1.4 Self Organizing Map (SOM)

Self Organizing Map (SOM) is an ML method where input data is organized and reduced into a lower dimension in an unsupervised manner with a competitive learning algorithm [31]. Even though SOM is one of the easiest methods to use, it is unstable to distribution shifts in the input data as well as the initialization of neuron weights.

The model introduced in [32] employs hierarchical SOM for a variety of different design structures. The model achieves unsatisfactory detection rates except for DoS attacks on KDD99 dataset. Several approaches also use hybrid models based on SOM. For example, the model proposed in [33] first reduces feature space with Principal Component Analysis (PCA) by selecting eight eigenvectors with less noise with Fisher Discriminant Ratio then classifies with SOM. The model achieves high sensitivity and specificity on NSL-KDD dataset. Another approach proposed in [34] employs J.48 decision tree for misuse detection and a SOM for anomaly detection. This approach first models normal data for TCP, UDP and ICMP protocols, then analyses anomaly with SOM. It obtains a high detection rate with low false positive on KDD99 dataset.

1.1.1.5 k-nearest neighbors (k-NN)

K-nearest neighbor (k-NN) is an ML method where each input data is assigned to a class of its randomly chosen neighborhood (k) according to their distance similarity [1]. Despite using fewer parameters, simple calculations, scalability, robustness to noise and uncovering natural patterns of data, there are a couple of problems related to this method. Firstly, choosing the right k parameter is not simple. Because too small k value models noise while too big k models other classes. Secondly, clustering algorithms like k-NN makes algorithm stuck on a local minimum point. Thirdly, using the Euclidean distance metric might not separate tangled data, in fact it may contribute misleading results. Lastly, this method becomes slow and memory inefficient for high dimensional data.

To accelerate detection time as well as to obtain a high detection rate, several approaches combine k-NN with methods such as feature selection and new feature representations. For example, the model introduced in [35] uses GMM to model statistical regularities in features. After GMM parameters are modeled in Gaussian

form, these parameters are fine-tuned with the EM algorithm. Then, the model classifies with k-NN using these parameters. The model achieves satisfactory detection results in particular on R2L and U2R attack types in KDD99 dataset. Similarly, the model proposed in [36] first uses a new feature selection approach that assumes the variance of a feature as a quality indicator and reduces all low-quality features. After the selection of ten features, the model uses k-NN and achieves faster detection than the one without feature selection on KDD99.

Several methods in the literature have used k-NN within the cascade system to achieve higher detection rates. For instance, the method proposed in [37] firstly ranks multi-resolution network traffic flow according to the level of anomaly then uses a threshold to classify a high-level anomaly labeled flow as an intrusion. The method achieves sufficient detection accuracy on KDD99 dataset. Similarly, the method described in [38] uses three level k-NN based cascade system. The method first extracts cluster centers and nearest neighbors then forms training data by summing the calculated distances between data and its cluster center and data and its nearest neighbor. The method obtains a significantly high accuracy and detection rate on KDD99 dataset. Similar to the methods proposed in [37, 38], the method introduced in [39] uses a two-tier system based on k-NN with the knowledge-based system. The method first uses a knowledge-based system to generate alarms then filters these alarms with k-NN. The method, however, achieves average results on DARPA-1999 dataset.

1.1.1.6 Bayesian Network

Bayesian Network is an ML based model that learns from the intrinsic behavior of input data by using statistical dependencies without requiring prior knowledge. Although this network detects small deviations in data and can be applied for continuous, discrete as well as binary input data types, there are some negative aspects related to it. Firstly, it may be vulnerable towards distributed/low-frequency attacks that create normal-like traffic. Secondly, it may be ineffective towards correlated features since it assumes that every feature is independent of one another while calculating statistical dependencies. Thirdly, it is slow for larger-scale input data due to computational load.

Bayesian Network is applied for several scenarios introduced in [40–43]. The approach proposed in [40] uses Naive Bayes on NSL KDD and achieves a high true positive rate for DoS, R2L, Probe attacks. Similarly, the model introduced in [43] uses Naive Bayes after discretization of the data. However, it only improves DoS attack detection on KDD99 dataset. Besides, the work introduced in [42] modifies Naive Bayes with Discretization Filter, where a set of predefined intervals are used to change

feature values into interval values, and obtains higher detection with small alarm rate than Naive Bayes itself on NSL KDD.

Several works in the literature combine Bayesian Network with other shallow algorithms to achieve higher detection results. For example, the model proposed in [44] employs Correlation Feature Selection (CFS) and Information Gain (IG) for feature selection then combines Adaptive Boosting and Naive Bayes on NSL KDD dataset to detect anomalies. Similarly, the method introduced in [45] combines Naive Bayes with ADAM based system on DARPA98 and DARPA99 datasets.

1.1.1.7 Decision Tree

Decision Tree is an ML model where all features are scanned and separated into groups. The model is composed of three main elements which are leaf, root and decision node. If-else command path obtains output from roots of decision tree to leaves while leaving less important features behind [46]. In particular, it is effective for classes with insufficient data and able to work with categorical as well as numerical input data. Moreover, it automates feature selections for trees and can be easily interpreted thanks to the tree structure. However, it ignores the mutual relationships among features. Mostly-known decision tree models are C4.5, CART and J48, respectively.

The model proposed in [47] employs a suffix tree using a sequence covering. This model calculates similarities between system calls on UNM and ADFA-LD datasets. Although it does not use the length of symbolic sequences to achieve faster-convergence than rival methods, the percentage of normal data samples in the training dataset plays a crucial role.

Several models in the literature combine Decision Tree with other algorithms. For example, the model proposed in [48] employs Decision Tree with IG to investigate features and their relevance to each sub-attack type. The method reports that source bytes and destination bytes are two of the most relevant features for all attack types on KDD99 dataset. Similarly, another hybrid model introduced in [49] uses Decision Tree with Bayesian clustering. The model first splits data into three classes: DoS, Probe and others then classifies others into attack and normal. As a final step, the model separates U2R and R2L attacks. The model achieves high detection rates except for U2R and R2L attacks on KDD99 dataset.

1.1.1.8 Fuzzy Logic (Fuzzy Set Theory)

Fuzzy logic is a method where classification boundary is treated as soft boundary among the range of 0 to 1 rather than firm boundary according to fuzzy rules [50]. These rules are defined to classify classes by the experts. Despite having uncertainty flexibility towards input data with fuzzy rules, these rules cannot be scaled to other systems easily.

The model introduced in [51] uses fuzzy association rules based IDS approach. However, the model achieves above average detection rates on KDD99 dataset. To achieve higher detection rates, several hybrid methods extend their Fuzzy Logic model by combining other algorithms. For instance, the models proposed in [52, 53] combine Fuzzy Logic with Genetic Algorithm (GA). Both methods achieve high accuracy and detection rates with low false-positive rates on private, KDD99, NSL KDD and Gure-KddCup datasets. Similarly, several hybrid methods proposed in [54, 55] first employ a fuzzy rough set for feature selection/reduction. Then, [54] uses k-NN and achieves state-of-the-art detection with a small error rate on KDD99 while [55] creates GMM based attack and normal pattern library and obtains high detection with low error rate on NSL KDD dataset. Another hybrid approach proposed in [56] uses Fuzzy Logic to create different training subsets then employs NN to classify attacks. The approach achieves improved precision and recall particularly on R2L and U2R attack types in KDD99 dataset.

1.1.1.9 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN) that is composed of neurons with associated scalar weights to interconnect other neurons, activation functions and layers. This network uses a backpropagation algorithm such as Gradient Descent to tune/update weights by minimizing classification error [57]. Despite their robustness to noise and compatibility with linear and non-linear inputs, choosing the optimal number of layers and neurons is difficult. Moreover, it may be stuck at local minima resulting from Gradient Descent.

The model introduced in [58] first employs Particle Swarm Optimization (PSO) to optimize parameters of MLP then conducts classification via MLP. The model obtains slightly better error rates than the one without PSO. Another anomaly model proposed in [59] first converts symbolic data to numerical by using Ghosh prototype and the canberra metric then employs MLP with the chaotic neuron. However, the model obtains average results on DARPA 1998 dataset.

A misuse based method is proposed in [60]. The method employs three different

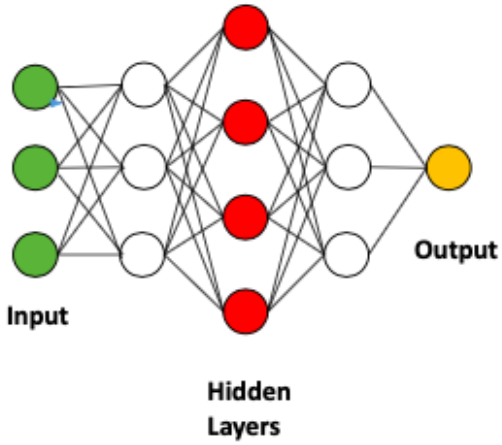

Figure 1.4 An example of Multilayer Perceptron

3-layer MLP structures and trains each by using ICMP, TCP and UDP based features, separately. Then, the method uses rules by thresholding each output from MLPs. Despite obtaining high detection rates on the private dataset with known DDoS and unknown DDoS attacks, the model is limited to a few types of attacks and may not handle DoS attacks with encrypted packet headers. Additionally, choosing the right threshold value is not easy.

1.1.1.10 Support Vector Machine (SVM)

Support Vector Machine (SVM) is an ML method that defines a hyper-plane by maximizing the margin among data samples from different classes [61]. To do that, the method uses the closest data samples to the hyper-plane from different classes and takes advantage of kernel space to map data into higher dimensional space. Despite having the advantage of separating non-linear data using kernel-trick, the choice of kernel type and the volume of feature space due to support vector size have a great impact on performance results.

Several methods proposed in the literature frequently employ SVM with other algorithms or cascade SVM. For instance, the model proposed in [62] uses two different SVMs, where one is for misuse, another is for anomaly detection. Also, the model presented in [63] first employs one of the Manifold methods, k variable locally linear embedding(kv-LLE), and Isomap for feature reduction. Then, the model uses SVM for anomaly detection. The model achieves high detection rates for anomaly detection on KDD CUP 99 and UNM datasets. However, kv-LLE and kv-Isomap combined with SVM achieves better detection rate than SVM itself on KDD99 dataset in terms of reducing false-positive rates. In addition to [62, 63], the model proposed in [64] first employs memory-efficient kernel tricked PCA for online feature extraction

then uses SVM to classify. The model achieves a high overall detection rate on KDD99. Also, the main contribution of the model is fast real-time/online detection.

Another hybrid detection model with SVM is proposed in [65]. The model uses agent for anomaly detection and SVM for misuse detection, where four different SVMs are trained for each attack type in KDD99 dataset. The model achieves a fast and high detection rate. Additionally, the method presented in [66] first reduces the feature space from 41 to 19 using GRF method then classifies with SVM. The method achieves high accuracy by improving training time.

1.1.1.11 Ensemble Methods

Ensemble classifiers are a combination of two or more shallow classifiers. Random Forest (RF) is one of the frequently used ensemble classifiers that consists of a bunch of decision trees. Although it is a shallow classifier, training many shallow decision trees contributes to optimizing/generalizing the model, add randomness and prevent from overfitting [67].

Several methods in the literature employ RF. For example, the method introduced in [68] first separates data using known patterns for specific intrusions then decides whether data belong to anomaly by using RF. The model achieves a high overall detection rate with low false alarms/positives on KDD99 dataset. Similarly, the model proposed in [69] uses RF-based model named as Hybrid Isolation Forest (HIF). The model first assumes unoccupied areas in feature space as normal then models potential-anomaly-spots using few anomaly samples. The model achieves high detection rates with a small improvement compared to other rival algorithms such as SVM on ISCX IDS 2012 dataset. Another RF-based model proposed in [70] first preprocesses by using Synthetic Minority Oversampling Technique (SMOTE) to grow training sample size of U2R from 52 to 468 in NSL KDD dataset. Then, the model employs IG to reduce features from 41 to 19. After preprocessing is completed, data is given to RF for multiple classifications. The model achieves state-of-the-art detection rates without false positives by improving detection for minority attack types. Likewise, the model introduced in [71] first employs RF for misuse detection then uses k-means for anomaly detection. The model obtains high overall detection with low false alarms on KDD99 dataset.

Besides RF-focused models, there are a variety of shallow classifier combinations with Adaboost algorithm in the literature. For instance, the model introduced in [72] first uses RF for feature selection then employs k-means++ to separate data into three clusters that represent normal, R2L and U2R attacks and remaining attack

types due to similarity among normal, R2L and U2R. After these steps, the model uses Adaboost to separate attack cluster into four sub-attack classes and achieves state-of-the-art accuracy on balanced KDD99. Similar to [72], the method described in [73] is composed of boosting algorithm, Adaboost and RF. The method achieves identical accuracy with the one using only RF. Moreover, the model proposed in [74] employs an ensemble of J48, Naive Bayes, Random Tree, AdaBoost, Meta Papping, DecisionStump and REPTree on NSL KDD dataset while another model introduced in [75] uses AdaBoost on KDD99 dataset. Both models achieve high accuracy.

In addition to discussed Adaboost combined models, other combinations are proposed to create ensemble models. For example, the model proposed in [12] first uses Non-symmetric Deep Autoencoder (NDAE) for dimension reduction then employs RF for classification. It achieves high accuracy on DoS and Probe attacks while obtaining below-average accuracy for minority attacks on both KDD99 and NSL KDD datasets due to the need for more samples to train/tune Deep Learning models well. Moreover, the main contribution of the model is shorter process time than standard DBN techniques. Similarly, the model proposed in [76] combines CART with Bayesian Network. The model obtains high detection rates especially for DoS, Probe and R2L attacks on KDD99 dataset. Moreover, a misuse based detection model proposed in [77] employs ensemble boosted decision trees. The model achieves high detection rates only for DoS, R2L and probe attacks on KDD99 dataset.

1.1.1.12 Evolutionary Algorithms

Evolutionary algorithms are models that are inspired by the natural process of evolution to solve optimization problems. There are a variety of evolutionary classifiers such as Genetic Algorithms (GA), Ant Colony Optimization, Particle Swarm Optimization (PSO). Among them, GA is one of most frequently used type of evolutionary classifiers that generates chromosomes randomly and make stochastic searches until the best combinations of chromosomes are found. During these searches, chromosomes evolve through mutation, crossover and selection. GA are advantageous at detecting global minima without requiring prior information about feature space. However, the decision of fitness function and hyperparameters is difficult.

The model proposed in [78] first employs PSO to reduce features in KDD99 dataset to eighteen then classifies with SVM. The model achieves high accuracy with low false positives. Another model proposed in [79] first uses rule mining process then optimizes with graph-based genetic network programming to model parameters. The model obtains high overall accuracy with low false positives on KDD99 dataset.

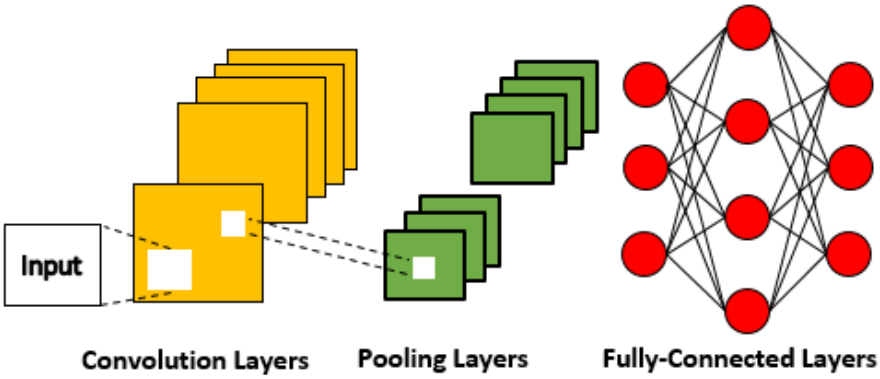

Figure 1.5 An example of Convolutional Neural Networks

Besides, the study introduced in [80] combines two GA with fuzzy sets to evolve new fuzzy rules. it evaluates new rules on several benchmark datasets.

1.1.1.13 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) is an ANN that is composed of different variations of convolutional, pooling and fully connected layers [81]. As can be seen in Figure 1.5, the input is first processed by convolutional and pooling layers that create a variety of feature maps to find informative representations of the input. Then, it is given to a fully-connected layer to classify. Besides, all weight parameters for convolutional and fully-connected layers are optimized by gradient descent during training.

CNN has a great advantage of automated feature extraction and is frequently used in many recent works. However, using CNN and its powerful backbones in two-dimensional space may require an additional step for the preprocessing of one-dimensional input to be compatible with two-dimensional input. For example, several approaches proposed in [82–84] use different preprocessing methods with CNN. The first one converts symbolic features into numeric values using binarization while converting continuous features into intervals to make them numeric features. Then, one-hot encoding and reshaping are applied to all converted features to form them pixel-like, respectively. Similar to the first method, the second one takes raw input composed of numbers as 8-bit binary numbers and converts these binary numbers into their analogous decimal counterparts. Afterwards, reshaping is applied to form them image-like. Likewise, the third converts input into grayscale image format after the process for the first method is done. Similarly, several methods employ these preprocessing steps after feature selection is done [85].

In addition to the new preprocessing steps, some works proposed in [86–88] focus on more state-of-the-art CNN backbones to obtain higher detection rates. In particular, most of the recent researches are conducted using LeNet backbone. For example, the model proposed in [17] uses this backbone with an additional batch normalization layer. The maximum performance result is reported as 94% accuracy for multi-class classification on NSL-KDD dataset while the detection rate on minority classes such as U2R, R2L is low. Similarly, the model introduced in [89] modifies this backbone with Inception modules. The performance result is reported as 94.11% accuracy on KDD99 dataset. Furthermore, the method proposed in [90] adds preprocessing steps to RGB scaling. After preprocessing, LeNet backbone is used for multi-class classification and average accuracy is reported as above 99.8%.

Several CNN-based approaches are also proposed without using backbone models. The models described in [16, 91, 92] focus on designing a novel CNN structure for DoS/DDoS detection on KDD99, private and CICDDoS2019 datasets, respectively. Similarly, the method introduced in [93] employs CNN on NSL-KDD dataset while the method proposed in [84] employs CNN on NSL-KDD, UNSW-NB15 and CICIDS2017 with a new encoding method that is designed to give equal weight for each feature and it provides twenty-four bits for each pixel as RGB-like encoding. Both reports high accuracy.

Other CNN methods combine CNN with other methods such as LSTM and AE. For example, the model proposed in [94] uses CNN with LSTM on the raw dataset. This model extracts not only temporal features but also the spatial ones. Although deciding hyperparameters such as the flow sizes is difficult, this model achieves high accuracy over 95% with flow size as 100. Similar to [94], the model introduced in [16] encodes raw input with one-hot encoding then uses CNN with AE. The performance result is reported as having 98,95% accuracy. Another combined model, a Character Level CNN (CLCNN), is proposed in [95]. This model first converts input, particularly each character in input to an eight-bit numerical string. Then, the model gives an encoded version of input into CNN. Reported results exhibit 98.8% accuracy.

1.1.1.14 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a type of ANN in which the hidden neurons are connected by following a temporal sequence. Thanks to such arrangement of their nodes, RNNs are principally used to process data in the form of time series [96]. Even though RNN poses some problems such as vanishing gradients, it is frequently used for time-series modeling for cybersecurity.

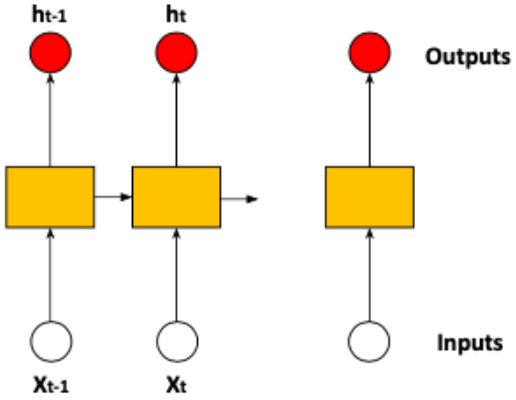

Figure 1.6 An example of Recurrent Neural Network

The model proposed in [97] uses RNNs and achieves high detection accuracy and fast real-time performance on the DARPA98 dataset. Similarly, the work introduced in [98] obtains higher accuracy than CNN, SVM, and RF classifiers on the ADFA-LD dataset. Another method proposed in [98] employs RNN with Gated Recurrent Unit (GRU) on ADFA-LD dataset. Since this dataset consists of system calls with various lengths, the semantic model uses different lengths between 10-30 of system-calls. The model achieves high detection rates. However, finding the optimal length of the system-call sequences may be problematic.

1.1.1.15 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) is designed as an improved version of RNN. An LSTM network consists of sequentially-connected neurons that are composed of input and output gate units, known as memory cells, to save the memory of previous inputs and forget these inputs for the subsequent interval of time [99]. As can be seen in Figure 1.7, the input is processed by sequential neurons to model as time-series.

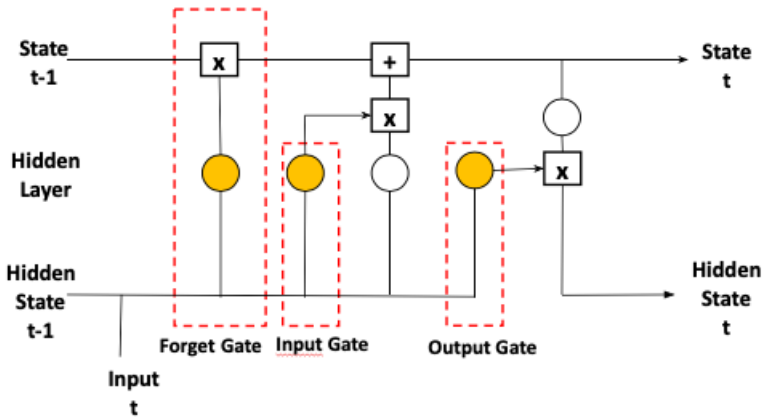

Figure 1.7 An example of Long Short Term Memory

LSTM has a great advantage of modeling time-series and it is employed in many recent works for cybersecurity despite the difficulty of choosing optimal hyperparameters. For instance, the models proposed in [100, 101] employ only LSTM as a classification algorithm on several benchmark datasets under different settings. While the first uses a 3-layer-structure on KDD99, UNM and ADFA-LD datasets, the second cascades LSTM combining their activations with voting in the end and both report high detection accuracy. Similarly, the work introduced in [102] only employs Bidirectional LSTM on UNSW-NB15 benchmark dataset and the study proposed in [103] adapts multivariate correlations analysis into LSTM on NSL-KDD dataset to separate feature subsets more efficient.

Besides, several works combine LSTM with DL algorithms, in particular with CNN. The works proposed in [94, 104] combine LSTM with CNN on frequently used benchmark datasets: KDD99 and CICIDS2017, respectively. Similarly, the approach introduced in [105] combines bi-directional LSTM with CNN to extract temporal and spatial features on NSL-KDD and UNSW-NB15 datasets after balancing the datasets with SMOTE.

1.1.1.16 Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine (RBM) is an energy-based neural network with two layers; hidden layer and visible layer, where the weights of the network are trained in an unsupervised fashion [106]. Since RBM can extract hidden patterns of input data modeling probability distributions of inputs, it is generally used for feature extraction in cybersecurity.

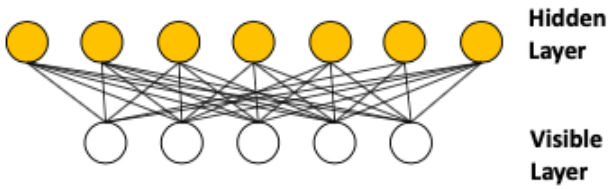

Figure 1.8 An example of Restricted Boltzmann Machine

The study introduced in [107] employs RBM for the FPGA-based intrusion detection system. Using RBM for this system increases computational efficiency by 30% on HTTP CSIC 2010 dataset. Similarly, the works proposed in [108, 109] use RBM for dimension reduction on KDD99 dataset to improve accuracy and memory efficiency. The work introduced in [110] uses RBM with AE on KDD99 to obtain powerful feature extraction and dimension reduction processes. The evaluation results show an improvement in detection.

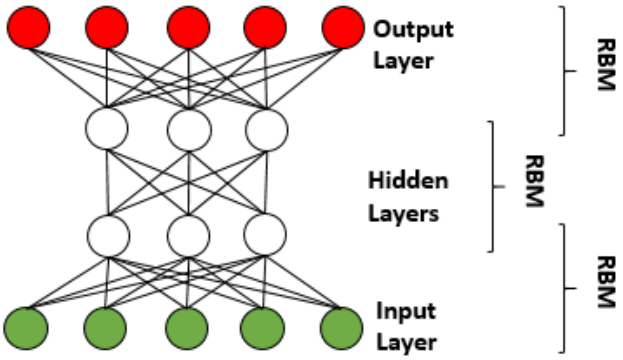

Figure 1.9 An example of Deep Belief Network

1.1.1.17 Deep Belief Network (DBN)

Deep Belief Network (DBN) is composed of several RBM blocks. As can be seen in Figure 1.9, all layers are fully-connected and only connections between successive layers are allowed while interconnections among non-subsequent layers and within each layer are prohibited [111]. DBN has a great advantage of being time-efficient thanks to training in a greedy-fashion as well as a good feature extractor/selector without requiring any supervision.

DBN is frequently employed for feature selection and combined with other ML algorithms. For example, the method presented in [112] uses DBN as a feature extractor with SVM to classify attack types on NSL-KDD dataset. Similarly, the approaches proposed in [113, 114] employ DBN to model and detect anomalies on the same benchmark dataset. A novel DBN-based model with extreme learning machine (ELM) on the same dataset is proposed in [115]. This model improves detection while reducing false positives.

In addition to the works mentioned, a variety of methods that use DBN are designed on KDD99 benchmark dataset. For instance; the [116] uses DBN with probabilistic ANN to detect intrusions. Similarly, the models introduced in [117, 118] employ DBN on the same benchmark dataset and report improved detection results compared to shallow ML methods such as ANN and SVM.

1.1.1.18 Reinforcement Learning (RL)

Reinforcement Learning (RL) is a DL method which uses agent interacting with the environment directly under three concepts: state, action and reward function. The agent first learns from its actions according to reward function then optimizes its state, where the reward function shows how good the action is and enables agent to learn what is good action by giving reward [119]. One of the frequently used RL methods

is Q-Learning, where reward function is based on Bellman Equation.

The model proposed in [120] first uses rough set theory to reduce features and discretize data with Q-learning that finds optimal cut values for features in the dataset. It achieves high accuracy on NSL-KDD dataset for anomaly detection.

1.1.1.19 Open Topics and Potential Directions for Cybersecurity

This section presents an overview of open topics and potential directions regarding new feature representations and the reliability of a model.

The first one of those potential directions are seen as novel feature representations. As new AI-based methods are available, security systems take advantage of these methods. Nevertheless, hackers use them for testing their novel attacks. For instance, GANs are used to generate new samples to train the system better, however, hackers could also take advantage of generating normal traffic data from a variety of sources with different background noise and then use their statistical properties to design a new attack.

Besides, these novel attacks can evolve to mimic normal traffic data and cheat the system. For example, DDoS attacks are a low-frequency version of DoS attacks and their characteristics exhibit a great similarity to normal traffic data. Although they are from the same attack family, low-frequency related features are more vital to detect DDoS attacks. Therefore, novel feature representations gain more importance and remain as a potential research area.

The second one of those potential directions are seen as reliability of the model. Due to the increasing number of new data, novel attacks may be labeled as normal by a human expert or their characteristics may not be differentiated as intrusion by security systems that are biased to their training dataset. Thus, the reliability of the model poses great importance for system breakdowns not to happen and TL becomes a potential research area for future works.

1.1.2 AI Methods for Biometric Systems

With the accelerated digitalization of daily-life applications such as digital banking, digital health services, digital security has gained importance. Therefore, a variety of authentication methods are employed to use digital systems safely in daily personal activities. In the beginning, simple authentication methods are used such as ID cards, passwords. However, these methods are prone to being copied easily thereby risky to secure private data. As a result, these issues created the need to deploy more trustable

authentication systems which are composed of a variety of methods.

As one of the widely-used and more secure branches of authentication systems, biometric systems are developed to prevent not only any personal information loss but also to prevent spoofing attacks. Biometrics is a field of science that recognizes the identity of an individual based on her/his personal characteristics and differentiates the individual from others. Besides, these personal characteristics can be grouped into two categories: physical and behavioral. While some frequently-used types of physical characteristics are fingerprints, palmprints, ears, finger-veins and facial characteristics such as iris, face, some types of behavioral characteristics are signature, voice and gait.

The algorithms used for biometric systems can be categorized into two general classes regarding the type of feature extraction: hand-crafted feature-based algorithms and automatically extracted feature-based algorithms. While hand-crafted features are designed specifically for the problem by experts and combined with shallow ML algorithms such as SVM, MLP, automatically extracted features are obtained via DL algorithms and they are adaptable to any other problems without expert supervision. Besides, the first category extracts low-level feature representations of input while the last category extracts not only low-level feature representations but also high-level ones thanks to the deep layer structure of DL algorithms. Since each algorithm has its advantages and disadvantages thereby, choosing the best one depends on the problem and the trade-offs. For example, DL algorithms can be performed simply and quickly. However, there may be such rare cases where spoofing is done with a high-level attacker and requires expert knowledge. Similarly, hand-crafted feature extraction may solve these cases but as a trade-off, it requires expert knowledge and cannot be adapted to new problems.

As many daily-life activities have started to employ interconnected systems, the protection of private data has become even more challenging. Moreover, attackers take advantage of using the newest algorithms to design their spoofing attacks so that distinguishing them from real ones becomes more difficult. To solve these issues, ML based approaches, in particular DL methods, are frequently used in biometrics for two main reasons. Firstly, DL methods can extract feature representations automatically without requiring any prior supervision. Therefore, they are easy to modify and adapt to any environment. Secondly, DL methods extract more informative feature representations thanks to their deep-layer structure. Thus, they are successful to find underlying patterns of data not only for real samples but also for spoofing ones. As a result of this, they increase accuracy and achieve breakthrough results.

This section presents an all-inclusive overview of the most popular and up-to-date

DL approaches for biometrics. In particular, signature is chosen as a behavioral characteristic while finger-vein is chosen as a physical characteristic among all biometric traits. Besides, a further investigation of potential future directions is presented.

1.1.2.1 AI methods for Signature Recognition and Verification

Before going into detail about up-to-date DL models, several methods concerning hand-crafted feature extraction are given in Table 1.1. With the great success of DL models, a variety of research fields have started to employ these models in their studies. Even though several ongoing types of research are on hand-crafted feature extraction, most of the recent studies for signature-based recognition and verification are dominated by DL models, particularly CNN.

Table 1.1 Some of frequently-used hand-crafted feature extractors

EXTRACTOR TYPE	FILTER NAME	REFERENCE
Edge	Sobel Filter	[121]
Edge	Canny Edge Detection	[122]
Edge	Boosted Edge Learning (BEL)	[123]
Texture	Gabor Filter	[124]
Blob	The Laplacian of Gaussian (LoG)	[125]
Blob	The difference of Gaussians (DoG)	[126]
Feature	Histogram of oriented gradients(HOG)	[127]
Feature	Scale-invariant feature transform(SIFT)	[128]
Feature	Speed Up Robust Features(SURF)	[129]

The works presented in [130, 131] use GAN for offline signature identification systems. While the first uses a hybrid approach of Writer-dependent (WD) with Writer-independent (WI), the latter adds GAN a threshold parameter for the loss function. Similarly, the study introduced in [132] employs other DL methods, Siamese RNN and LSTM, to distinguish adversarial samples under WI supervision.

The model proposed in [133] employs CNN for signature verification and obtains state-of-the-art performance. Then, the work presented in [134] extends the previous CNN approach with Model-Agnostic Meta-Learning (MAML) to learn CNN filter weights and improve performance results. Similarly, the model proposed in[135] uses CNN backbone, Inception themed GoogleLeNet for signature verification. Another model proposed in [136] uses CNN with modification using Logit layers to calculate similarities between the reference and input samples. The model achieves state-of-the-art performance results thanks to the modification.

1.1.2.2 AI methods for Finger-Vein-based Biometric Identification and Verification

Until now, most of the studies are conducted with a variety of hand-crafted extractors as given in Table 1.1. After GPUs have become common and automatic extractors have become the trailblazers, CNNs with their several backbones have started to be used. Recent studies for finger-vein based identification and verification are frequently done using CNN. For example, the works proposed in [137–139] employ VGG-16 -a CNN backbone- for input resolutions as 65x153, 128x128 and 224x224, respectively. The first study reports high accuracy over four benchmark datasets. While the second study modifies VGG-16 by adding more layers, the third one uses pre-trained weights to achieve high accuracy. Another CNN backbone, AlexNet structure is employed with small modifications on kernels in the work introduced in [140]. Similarly, LeNet backbone is used in the model proposed in [141].

Although state-of-the-art methods are successful at finding good feature representations of the data, several ongoing studies proposed in [140, 142, 143] report that even these methods are slightly weak to adversarial attacks which are designed by printing out original images and showing those images directly to the sensor. Moreover, these attack vectors can be designed using state-of-the-art AI methods such as GANs. Therefore, liveliness detection and novel sensors that detect temperature have started to be used in recent ongoing works such as the model introduced in [142].

1.1.2.3 Open Topics and Potential Directions for Biometric Systems

This section presents an overview of open topics and potential directions regarding new feature representations and the reliability of a model for biometric systems.

The first one of those potential directions are seen as novel feature representations. As new AI-based methods are available, both signature and finger-vein biometric systems have employed these methods to improve their accuracy. However, these new approaches either require more data samples to model better or use of pre-trained weights. For example, CNN and its backbones require a variety of input data from different viewpoints to model. Nevertheless, biometric benchmark datasets have small sample sizes per class and some of them are imbalanced. Furthermore, pre-trained weights may not be optimal for several biometric datasets. Therefore, new feature representations that model viewpoint invariance and equivariance using few data gain importance and remain as a potential research area.

The second one of those potential directions are seen as reliability of the model. Due

to the increasing number of new data, new spoofing attacks may not be distinguished easily by the biometric systems that are biased to their training dataset. Therefore, the reliability of the model poses great importance and remains as a potential research area for future works.

1.2 Objective of the Thesis

The main objective of the thesis is to conduct comprehensive research of domain free DL based Security Models for CFS concerning data collection, transmission and storage, where domain free indicates bias-free, more reliable systems to be used for different domains.

To decide the domain free AI algorithm, we first take open topics and potential directions for cybersecurity and biometric systems into consideration. Considering the recent state-of-the-art feature extractors such as CNN backbones: VGG, ResNet, Inception, their requirement for a huge volume of input data from different viewpoints or the adaption of pre-trained network weights to model the input data, we decided to deploy an AI algorithm that has two distinct qualities. The first quality is both the ability to model invariance and equivariance of data without requiring any augmentation and the ability to keep spatial relationships among features and model feature activations smarter than using only scalar values. The second quality is the ability to work and achieve feature representations similar to the human neural system under hierarchical supervision, abstraction and adaptability to domain-independent datasets.

As for the first quality, although CNN backbones achieve rotational robustness with pooling layers, this robustness is limited to small local rotations. Therefore, a variety of novel approaches use CNN by adapting either rotation invariant convolution outputs or convolution filters. For example, steerable CNN [144], Group Equivariant CNN (G-CNN) [145], Harmonic Networks [146], CubeNet [147] use rotated/transformed convolutional filters in different orientations while the model introduced in [148] manipulates feature maps. However, these rotations are limited to the finite-set of orientations and still require more variations in the input data for complex systems. Besides, pooling layers still lose information about spatial relations. Since CNN and its variants are on Euclidean domain, a non-Euclidean domain such as Graphs, Point Clouds and Manifolds is employed to extend the generalization ability and named as Geometric DL. For instance, Graph Convolutional Network (GCN) [149], Geodesic CNN on Riemannian Manifolds [150] and 3D Keypoints with Geometric Reasoning [151] employ graphs or manifolds to model three-dimensions [152]. Although these

methods model inputs using many orientations of tangent space thanks to Lie Algebra, they are computationally complex to use in real-time [153].

As for the second quality, several algorithms are proposed particularly for the abstraction of classes. For example, the method introduced in [154] searches embeddings attached to each class using non-linear mapping and clustering to find the abstract prototypes for each class. Similarly, the method proposed in [155] abstracts each class with the mean output vector. Even though these methods are good at simplifying, they do not use convolution filters which are the most powerful feature extractor in a two-dimensional domain.

Since Capsule Network:

- models with small data and do not require input from different viewpoints and simply models input with affine matrices,
- takes advantage of weight sharing properties of convolutional layers,
- keeps spatial relations of activations and can recognize the parts and their spatial relationships among one another like the human brain.
- abstracts the activations and routes only the most contributing ones to the next layer with routing function. Therefore, unrelated capsules become less effective and the network exhibits Explainable AI (XAI) model characteristics.

Capsule Network is employed as a domain free AI algorithm to conduct experiments.

In addition to the main objective of this study, there are several sub-objectives summarized as examining Capsule-based feature representations for different security-themed CFS for the first time, TL for one-dimensional feature space, detailed investigation of Capsule Network and its hyperparameters. Furthermore, all experiments are conducted using Python and DL models are built using TensorFlow library in Python.

1.3 Hypothesis

Original contributions are listed below:

- A comprehensive research on a domain-free AI model for security-themed CFS is conducted for the first time.

- Capsule Network performance and comparisons with its CNN equivalent are analyzed, in particular their feature representations and impact on performance results are investigated for biometric systems and cybersecurity for the first time.
- TL scenarios are extended to one-dimensional feature space for cybersecurity and conducted under the content of reliability for the first time. Moreover, their limitations are analyzed in detail under different scenarios.

2
AI-BASED APPROACHES FOR BIOMETRIC SYSTEMS

CNN is proposed for automatic feature extraction and to obtain high-level representations of input images in [156]. CNN consists of three main parts: convolutional layers, pooling layers, and fully connected layers. The convolutional layers are used for feature extraction by using a variety of kernels to find the best input representation. Then pooling layers take the output of convolution layers and discard non-informative parts of the outputs coming from the convolution layers as well as reduce dimensions for computational efficiency. After these two types of layers with different combinations have been completed, the fully connected layer uses the most informative extracted features to classify inputs. Until now, different combinations of the first two parts of CNN have been used as backbones, such as LeNet, ResNet, and VGG.

For this chapter, the CNN-based equivalent model was chosen to demonstrate that the output of two convolutional layers is not as informative as the output of Capsule layers. Moreover, it has a similar complexity to Capsule Network in terms of kernel sizes in convolutional layers and neuron sizes in fully connected layers. In addition to the CNN-based equivalent model, LeNet-5, which is one of the backbone models of CNN, is employed due to its similarity to the model structure of Capsule Network for a fair evaluation.

Although CNN offers translational invariance with pooling, it has limited rotational invariance. Therefore, CNN requires data from different viewpoints. Capsule Network is proposed to model feature representations of an object without requiring samples from different viewpoints by ensuring translational and rotational invariance [157]. The network structure is shown in Figure 2.1 and algorithm flowchart of Capsule Network is shown in Figure 2.2. As can be seen from Figure 2.1, the model consists of four main parts which are convolution layers, Primary Capsules, Signature Capsules and fully connected layers, respectively. Here, Primary Capsules puts activation outputs from convolution layers into capsules to obtain vector representation of features instead of numerical activation values in CNN.

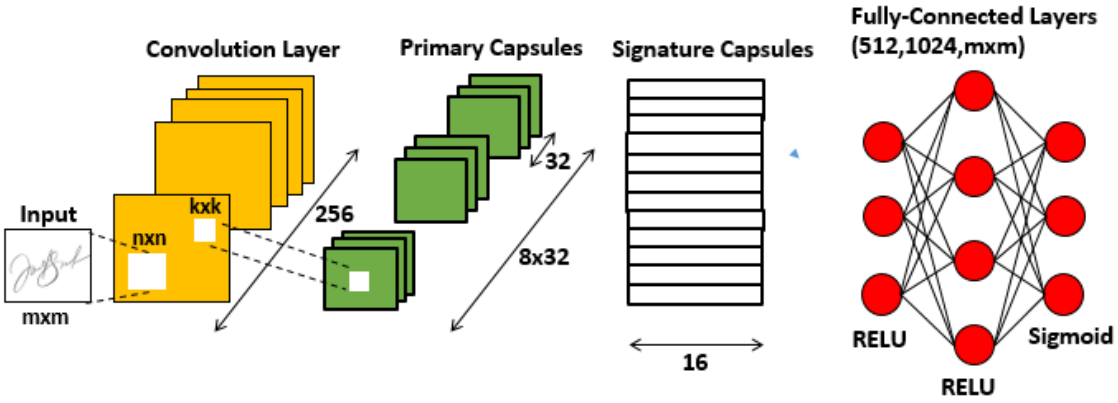

Figure 2.1 Capsule Network

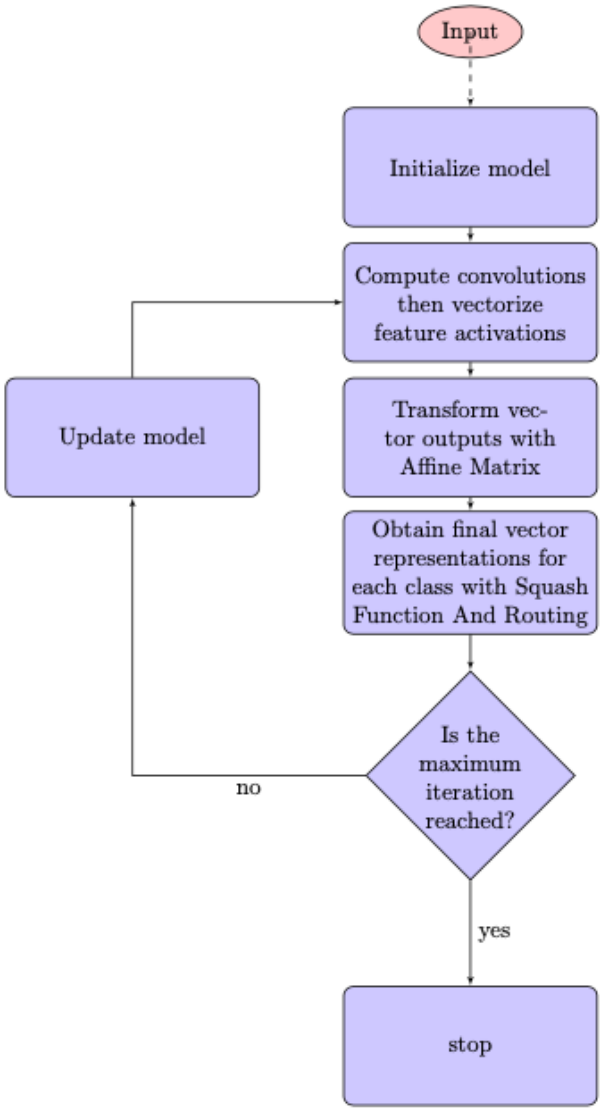

Figure 2.2 Algorithm flowchart for Capsule Network

Firstly, a variety of convolutions are applied to input images to obtain local low-level features in convolution layers. After obtaining activations as the output of these convolution layers, all these scalar-valued activations are given to primary capsules to be grouped into multi-dimensional vector representations. Then, these multi-dimensional vectors are multiplied with the affine transformation matrix to obtain many different variations of these vectors for better modeling of the input images. To select the most informative feature vectors, the routing algorithm is employed. Before employing this routing algorithm, all transformed feature vectors are squashed according to the Equation 2.1 to make discriminative feature vectors more apparent and to fade less-informative ones out. After squashing, the most informative vectors are routed to signature capsules to form an entity. When the routing algorithm is agreed, signature capsules with one multidimensional capsule per class is created. The new capsule keeps the information about all outputs of capsules from the previous layer and keeps absolute characteristic features for each class.

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} x \frac{s_j}{||s_j||} \tag{2.1}$$

where v_j is output of a capsule, s_j is total input of a capsule and s_j also includes affine transformed versions of convolution outputs which can be studied in detail from [157].

Lastly, these signature capsules are fed into fully connected layers to classify. The loss function is designed as a combination of margin loss, which is obtained from false predictions, and reconstruction loss. This loss function is calculated according to the Equation 2.2.

$$L = T_c max(0, m^+ - ||V_c||)^2 + \lambda(1 - T_c) max(0, ||V_c|| - m^-)^2 \tag{2.2}$$

where L is loss term for one signature capsule, $m^+ and$m are constants and chosen as 0.9 and 0.1, T_c is a constant that is 1 if the signature capsule is the correct, otherwise it is 0. First-term of loss equation is to calculate correct prediction probability while the latter term is employed to calculate incorrect prediction probability.

All in all, Capsule Network provides three main innovations compared to Convolutional Neural Network:

- Inner affine matrix multiplication instead of data augmentation
- Vector representation instead of scalar-valued representation
- Forwarding only the most informative feature representations by Dynamic Routing algorithm instead of forwarding all extracted feature activation values

2.1 Model for Offline-Signature-based Identification and Verification Systems

Biometrics is a field that uses behavioral and biological traits to identify/verify a person. Due to ease of collection and being non-invasive, signature-based biometric systems are frequently used. These systems are divided into two sub-systems depending on their collection method; online and offline systems. The first one collects dynamic signature features as a sequence of time. In this manner, features such as speed and pressure can be extracted. The second one uses the image after signing is done. Even though the online signature is advantageous for keeping more details about a signature, the offline signature is the most frequently used behavioral trait in daily life [158].

Figure 2.3 Two genuine (first two rows) and one forgery signature (last row) samples from CEDAR, GPDS and MCYT databases, respectively [159–161]

In offline signature verification systems, the main aim is to separate the genuine signatures from forgeries, which can be random, simple or skilled done by a forger. Unlike random and simple forgeries, skilled forgeries are not always easy to distinguish due to the intra-class variance of genuine signatures shown in Figure 2.3. Therefore, a detailed investigation of not only local but also global features of genuine signatures is required to achieve high verification results. Moreover, insufficient prior knowledge about forgeries during training and limited genuine samples make the verification process even more challenging.

A great deal of research has been devoted to extracting the most informative global and -in particular- local feature representations to differentiate forgeries. These hand-crafted local descriptors can be texture-based such as gray level co-occurrence matrix [162], direction-based such as Histogram of Gradients (HOG) [163], Scale

Invariant Feature Transform (SIFT) [164] or combination of two or more different local descriptors [165]. While research on hand-crafted local descriptors is still ongoing, recent studies have been conducted by employing automatic feature extractor algorithms such as CNN. Since data samples per user are limited, a few studies are dedicated to using transfer learning instead of conducting data augmentation. The works presented in [166, 167] use a pre-trained CNN-based model after training the model with other benchmark datasets for coarse-tuning weight parameters. After this coarse-tuning process, limited training data from the original dataset is used for fine-tuning. In brief, coarse-tuning is employed to narrow down signature feature space while fine-tuning is used to guarantee optimal decision boundaries. Similarly, the model proposed in [168] employs Siamese CNN with an inception layer. To cope with the few data samples per user, the model generates augmented samples for training. The model achieves 99.15% and 99.82% Area Under the ROC Curve (AUC) for sub CEDAR and MCYT databases.

Unlike pre-trained CNN-based models, the method proposed in [157] narrows feature space down by only modeling with few data from the original dataset. From this point of view, the first goal of this chapter is to investigate the feature modeling capability of Capsule Network and to evaluate Capsule Network under different input resolutions, such as 64x64 and 32x32, which are four to eight times lower than the usual signature resolutions for practical usage of signature verification and identification tasks. This goal is chosen not only to investigate the modeling capability of Capsule Network without requiring pre-trained weights under extremely low resolutions but also to fasten evaluation times and lessen memory usage. The second goal is to obtain a comparison among Capsule Network and its CNN equivalent on three benchmark databases to understand how well algorithms can keep features as informative as possible under extreme conditions.

2.1.1 Benchmark Datasets and Preprocessing Steps

2.1.1.1 Benchmark Datasets

In this chapter, three frequently-used offline signature databases are employed for identification and verification tasks.

CEDAR: CEDAR database consists of 1320 genuine and 1320 forgery samples in total and 24 genuine and 24 forgery samples are collected per user among 55 users [159].

MCYT: MCYT database consists of 1125 genuine and 1125 forgery samples in total and 15 genuine and 15 simulated forgery samples are collected per user among 75 users [161].

GPDS: GPDS database consists of 96000 genuine and 120000 forgery samples in total and 24 genuine and 30 simulated forgery samples are collected per user among 4000 users [160]. In this chapter, we employed GPDS-100, which is only the first 100 users for identification and verification tasks [163, 169].

2.1.1.2 Preprocessing Steps

Before the evaluation procedure, benchmark databases are preprocessed as shown in Figure 2.4. Firstly, data samples for each database are cropped regarding the center of signatures to discard unnecessary parts. Then, these data samples are resized to 64x64 and 32x32 extreme image resolutions. After resizing is done, data samples are converted into binary values with Otsu's method. As a final step of preprocessing, binarization is done to make background pixels black, foreground pixels white.

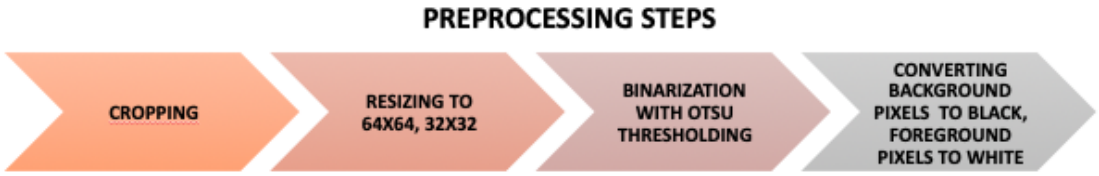

Figure 2.4 Preprocessing steps for signature benchmark databases

2.1.2 Experiments and Conclusion

2.1.2.1 Experimental Setups and Experiments

General settings for the identification task are given below:

- One model is trained for all users in a specific dataset.
- Only genuine samples are used for training and testing. Train and test partitions are set as the first half and the second half of genuine samples per user, respectively.
- Two-fold cross-validation is employed for both tasks.
- For training, epoch size and batch size are chosen as 50 and 16.
- Image resolutions of 64x64 and 32x32, which are 4-8 times lower than the usual, are used for identification task [166, 170–172].
- Capsule Network hyper-parameters such as layer structure, the routing number are chosen as the original in [157]. Only convolution kernel sizes and capsule dimensions are modified as given in Table 2.1 and Figure 2.1.

- Evaluation metric is chosen as accuracy since classes in datasets are balanced.

General settings for the verification task are given below:

- A model is trained for each user separately, which is also known as the Writer-dependent approach. For example, 55 separate models are created for 55 users in CEDAR dataset and only average accuracy of all models is reported.
- Genuine as well as random and simple forgery samples, which are treated as a separate class, are used for training and testing. Train partition is set as the first half of genuine and forgery samples per user while test partition is set as the remaining half.
- Two-fold cross-validation is employed.
- For training, epoch size and batch size are chosen as 50 and 16, respectively.
- Only image resolution 64x64 is used since the inner variance of genuine signatures makes modeling genuine signatures difficult for smaller resolutions such as 32x32.
- Model hyper-parameters such as layer structure and the routing number are chosen as the original in [157]. Capsule dimension is set the same as in the identification task. Only convolution kernel sizes and capsule dimensions are modified separately as given in Table 2.2 and Figure 2.1 for each dataset.
- Evaluation metric is chosen as accuracy since classes in datasets are balanced.

Table 2.1 Test accuracy for Offline Signature Identification tasks

Resolution	**Dataset**	**nxn(stride)**	**kxk(stride)**	**Train**	**Test**	**Accuracy**	**Equiv. Acc.**
64x64	CEDAR	21x21(1)	21x21(2)	12	12	%97	%55
32x32	CEDAR	13x13(1)	11x11(2)	12	12	%96	%54
64x64	GPDS-100	21x21(1)	21x21(2)	12	12	%94	%54
32x32	GPDS-100	13x13(1)	11x11(2)	12	12	%89	%51
64x64	MCYT	21x21(1)	21x21(2)	12	12	%95	%55
32x32	MCYT	13x13(1)	11x11(2)	12	12	%91	%51

All experimental results for offline signature identification tasks are given with the information of input resolutions, train-test partitions and convolution kernel sizes with stride in Table 2.1. As can be seen, even outputs of big convolution kernels are good enough at modeling and separating signatures from one another using Capsule

Table 2.2 Test accuracy for Offline Signature Verification tasks

Resolution	Dataset	nxn(stride)	kxk(stride)	Train	Test	Avg. Acc.
64x64	CEDAR	21x21(1)	21x21(2)	14+14	5+5	%91
64x64	GPDS-100	3x3(1)	5x5(2)	12+15	12+15	%86
64x64	MCYT	21x21(1)	21x21(2)	8+8	7+7	%89

Network while they are not enough to model using CNN-equivalent. Furthermore, identification at input resolutions of 32x32 achieves average %92 accuracy over three benchmark datasets for Capsule-based representations.

All experimental results for offline signature verification tasks are given in Table 2.2. As can be seen, only 64x64 image resolution is used due to the difficulty level at differentiating forgeries from genuine.

Moreover, genuine and forgery samples in the GPDS-100 dataset seem almost identical. Therefore, they require smaller kernels to extract local features in detail for verification tasks. Only for this dataset, additional two convolution layers before primary capsules are added as 3x3(1) and 5x5(2). Additionally, train and test samples are given as genuine+forgery format.

2.1.2.2 Conclusion

Capsule Network acquires promising results while using at least four times lower resolutions than frequently used ones for identification tasks. This indicates that Capsule Network is reliable enough to classify signatures and to have a unique ability to model local features better under extremely low resolutions for practical purposes. Moreover, results for identification tasks reveal that even using bigger sized (around one-third of input resolutions) convolutions are useful to separate signatures owing to the modeling capability of Capsule Network while CNN equivalent cannot perform well and requires bigger input resolutions and/or more layers.

Similarly, verification results also indicate that the algorithm has a great capability to cope with differentiating genuine signatures from forgeries. However, unlike identification tasks, high-similarity between genuine and forgery samples requires the extraction of low and mid-level features together. Moreover, different benchmark datasets require different levels of convolution layers. For instance, two-layer kernels are enough to extract enough information in CEDAR and MCYT datasets while GPDS-100 dataset requires more. Therefore, convolution layers and convolution kernel sizes are arranged for that requirement for all datasets.

For future works, there are a couple of things to be taken into consideration for offline

signature identification and verification tasks.

- Different model combinations for Capsule Network may be tried, such as modifying the stride and convolution layers.
- Capsule Network and state-of-the-art CNN models such as VGG-16 can be compared for high input resolutions using the same convolution layers to investigate the modeling capability of Capsule Network in detail.
- For verification task, performance comparison can be generalized with the use of adversarial attacks, such as adding noise to genuine signatures.
- Visualising feature representations before and after capsule layers can be done to increase the explainability of capsule-based feature modeling.

In conclusion, the main point of this chapter is to investigate Capsule Network's advantages in terms of data representation, using few data in signature identification and verification tasks for CPS and to encourage a community that is interested in online signature verification to think one step further to obtain better feature representations for the future.

2.2 Model for Finger-Vein-based Biometric Identification Systems

In comparison to other biometric system types, finger-vein-based recognition provides some advantages due to its non-invasive and low-cost procedure, simplicity of collection, and the fact that it is one of the biological characteristics that is affected only by internal factors [173]. Although finger-vein-based biometric systems have these advantages, they possess some drawbacks, such as poor quality of finger-vein images related to internal factors that have a negative impact on the accuracy of finger-vein recognition methods [174]. These internal factors could be finger tissue-based problems as well as the quantity of fat and water levels under the skin.

There are two types of finger-vein identification systems: finger-dependent and individual-dependent systems. The first focuses on each finger and creates separate feature spaces for individuals, while the latter uses all fingers belonging to an individual and creates feature space for each individual. Since each finger vein has its characteristics, a person's fingers lack common features. Thus, many types of researches are conducted using a finger-dependent approach, such as [137, 175]. In recent years, there have been significant technical advances in the technology of graphics processing units (GPUs) of computers. Moreover, with the increase in the

number of open benchmark databases, these technical advances have stimulated an increase in CNN-based implementations in biometrics. Besides, unlike conventional methods, CNN accomplishes automatic feature extraction. This motivates most of the ongoing researches to use CNN-based approaches to improve recognition performance as well as the robustness of the recognition system. Li et al. proposed a system based on CNN backbones, VGG-16 and AlexNet, that use pre-trained weights [176]. Similarly, Hong et al. proposed a method of applying VGG-16 and VGG-19 backbones with pre-trained weights [177]; additionally, Das et al. used a VGG-16 backbone for finger-vein identification [137]. Even though the VGG-16 backbone dominates the other CNN-based backbones in recent researches, whether VGG-16 with pre-trained weights achieves satisfying results due to extracting the best representations is open for debate [157].

From this point of view, the first goal of this chapter is to investigate the feature extraction capability of Capsule Network for finger-vein-based identification. The second goal is to obtain a comparison among Capsule Network, its CNN equivalent, and LeNet-5 on four benchmark sub-databases. Moreover, all of these evaluations are made using 32x32 image resolutions for practical purposes, which is much lower than the usual setup in use, such as 224x224. Therefore, this comparison also evaluates how well algorithms can keep features as informative as possible under extreme conditions.

2.2.1 Benchmark Datasets and Preprocessing Steps

2.2.1.1 Benchmark Datasets

Four publicly available finger-vein image databases are used for the experiments. These are SDUMLA from Shandong University, UTFVP from University of Twente, HKPU from Hong Kong Polytechnic University and MMCBNU-6000 from Chonbuk National University. Details about these databases are given in Table 2.3. For the experimental setup, the first eighteen fingers are chosen from each database while creating sub-databases for evaluation.

2.2.1.2 Preprocessing Steps

In vein identification systems, there are several frequently used pre-processing techniques, such as repeated line tracking and maximum curvature. Before the evaluation procedure, benchmark databases are preprocessed as shown in Figure 2.5.

They are also summarized in detail below:

Table 2.3 Benchmark databases for Finger-vein Identification

Name	Number of Individuals	No of samples per Individual	Total Samples	Image Resolutions
SDUMLA [178]	106	36 (6Sx6F)	3816	320x240
UTFVP [179]	60	24 (4Sx6F)	1440	672x380
HKPU [180]	156	24 (12Sx2F)*	3132	513x256
MMCBNU-6000 [181]	100	60 (10Sx6F)	6000	320x240

S: samples per finger
F: total no of fingers
*: 12S(first 105)and 6S(last 51)

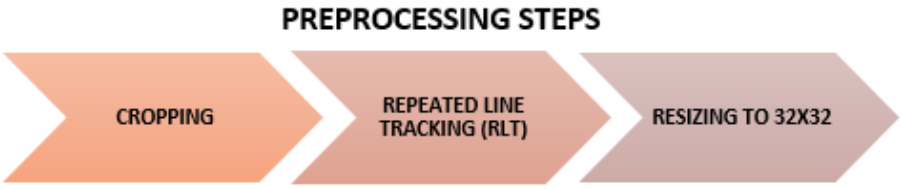

Figure 2.5 Preprocessing steps for finger-vein benchmark databases

- **Cropping** : It is done to discard the irrelevant parts of the finger-vein image.
- **Repeated Line Tracking (RLT)** : To capture the edges of finger veins, RLT is employed to track local black lines (veins) and separate them from the background in a pixel-wise manner until there is no longer any local black lines tracked [182].
- **Resizing** : It is done to convert input images to 32x32, which is a much lower resolution than those in the literature.

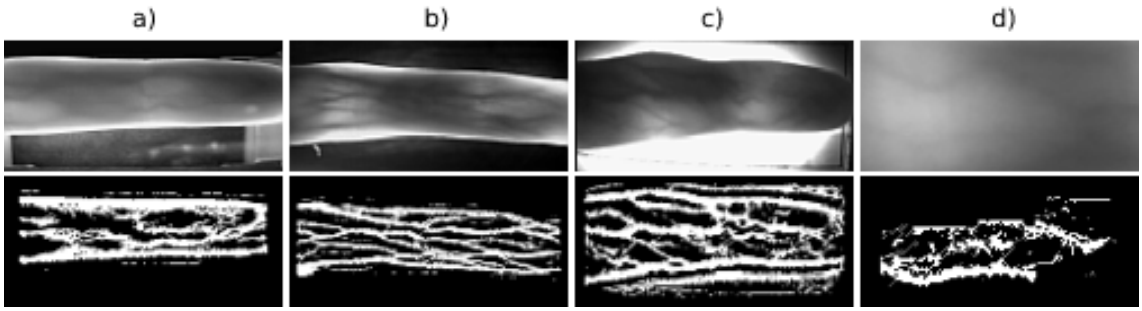

Figure 2.6 Original and pre-processed finger-vein samples of a) SDUMLA [178] b) UTFVP (Twente) [179] c) HKPU [180] d) MMCBNU-6000 [181] databases

Samples from each database and pre-processed versions of these samples can be seen in Figure 2.6.

2.2.2 Experiments and Conclusion

2.2.2.1 Experimental Setups and Experiments

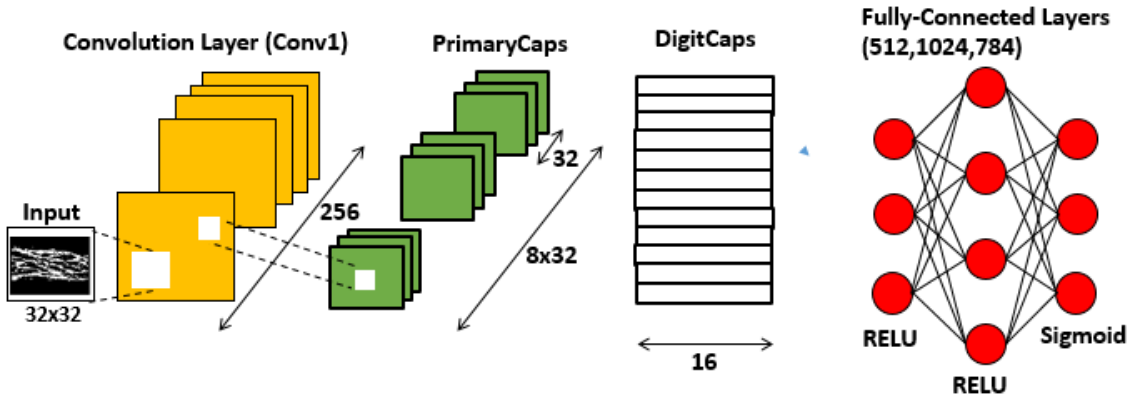

Figure 2.7 Capsule Network model for Finger-vein Identification

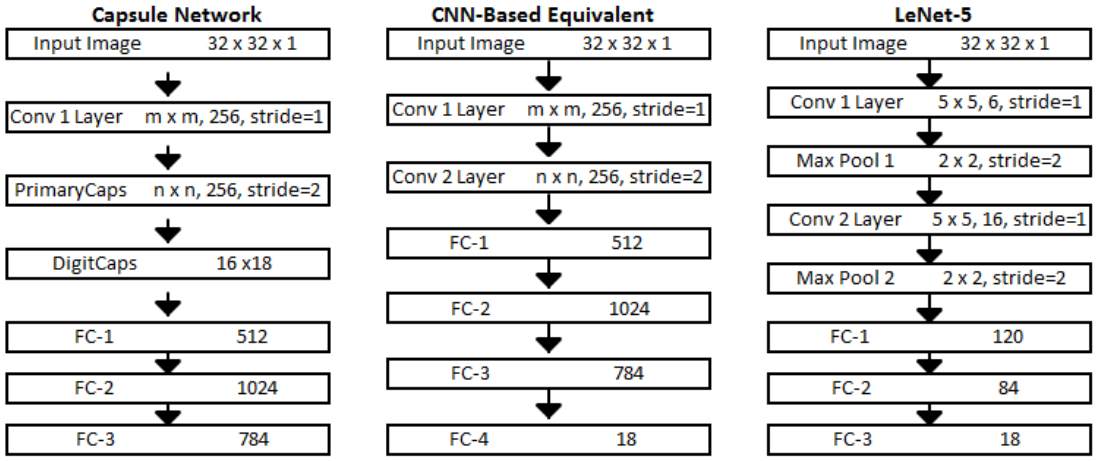

Figure 2.8 Capsule Network, CNN-based equivalent and LeNet-5 model structures

Model layer structure is given in Figure 2.7 and chosen similar to the original work proposed in [157]. Similarly, all models and their layer parameters can be seen in Figure 2.8. Here, convolution filter kernels are chosen as 2x2 for CONV1, 3X3 for PrimaryCaps which are at least 10% of the input image size to secure absolute information similar to work proposed in [137]. For all experimental setups, epoch size, routing number, and capsule size are chosen 500, 3, and 32, respectively.

In addition to the hyperparameter setup of the model, train-test partitions are chosen according to the sample size per finger. Due to the small sample size, these partitions are arranged as half for training half for testing at least and given in Table 2.4. For SDUMLA database, each finger has one session with six samples. Therefore, three train-test percentages are chosen as 3-3, 4-2 and 5-1. For UTFVP (TWENTE) database, each finger has one session with four samples. Therefore, two train-test percentages

Table 2.4 Evaluation results for Finger-vein Identification

			Accuracy		
Database	**Train**	**Test**	**Capsule Network**	**CNN Equivalent**	**LeNet-5**
SDUMLA	3	3	87%	42%	74%
	4	2	88%	52%	83%
	5	1	**100%**	88%	**100%**
UTFVP	2	2	66%	41%	77%
	3	1	**94%**	55%	88%
HKPU	6*	6**	56%	25%	60%
	6+1	5	67%	30%	63%
	6+2	4	79%	34%	75%
	6+3	3	83%	47%	83%
	6+4	2	**88%**	50%	86%
MMCBNU-6000	5	5	95,5%	77%	92%
	6	4	95,8%	77%	90%
	7	3	98%	79%	94%
	8	2	**100%**	80%	97%

*: from session one
**:from session two

are chosen as 2-2 and 3-1. For HKPU database, each finger has two sessions with six samples each (12 samples in total for per finger). Therefore, five train-test percentages are chosen as (3+3)-(3+3), (6+1)-5, (6+2)-4, (6+3)-3 and (6+4)-2. MMCBNU-6000 database, each finger has one session with ten samples. Therefore, four train-test percentages are chosen as 5-5, 6-4, 7-3 and 8-2. For evaluation metric, accuracy is chosen since classes in datasets are balanced.

2.2.2.2 Conclusion

Performance results are given in Table 2.4. For all databases, the results indicate that even though Capsule Network and the CNN-based equivalent use the same CNN-extracted features, Capsule Network achieves better performance results at modeling finger-vein.

For SDUMLA database, it is also pointed out that Capsule Network and LeNet-5 obtain the same results for 5-1 train-test partition. This may indicate that if CNN-extracted features are discrete enough among classes and there is a sufficient amount of training data, LeNet-5 works as well as Capsule Network. However, for the opposite case, where CNN-extracted features are not discrete, Capsule Network tops the accuracy of 94%, 88% and 100% for UTFVP, HKPU and MMCBNU-6000 databases, respectively.

One intriguing point to note is that all evaluations are done under 32x32 image resolution, which is much lower than usual and satisfactory for practical purposes. Furthermore, there is not a single finger-vein-based biometric system that uses this

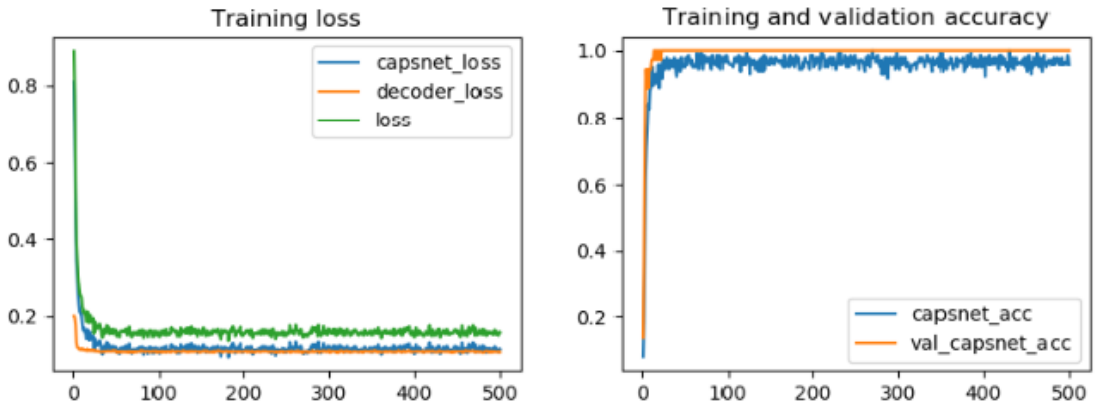

Figure 2.9 Accuracy vs Epoch for MMCBNU-6000 database with train-test partition as 7-3

kind of extreme image resolution and obtains results as high as those in this chapter.

Besides, accuracy becomes stable by 50 epoch for all vein databases thanks to fast convergence. As an example, results for MMCBNU-6000 database with Train-Test partition as 7-3 are given in Figure 2.9.

To sum up, performance results show that Capsule Network is quite robust in that it can achieve high results while only using a small number of samples and outperforms its opponent algorithms. Moreover, it achieves an average 95.5% accuracy over four benchmark sub-databases, while opponents, which are the CNN-based equivalent and LeNet-5, obtain a maximum average of 92.5%. Performance results also show that 32x32 image resolutions are enough for finger vein identification and Capsule Network-based finger vein identification obtains promising results for all practical purposes.

For future works, there are five main steps to be taken: Firstly, Capsule Network can be evaluated on other finger-vein databases using whole samples. Secondly, the capability of Capsule Network towards adversarial attacks can be analysed on benchmark databases with adversarial samples, such as VERA database. Thirdly, model parameters such as convolution kernel sizes and the number of layers for Capsule Network can be modified. Fourthly, different CNN backbones, both with and without pre-trained weights, can be employed to generalise results. Lastly, using other commonly used pre-processing methods, such as Contrast Limited Adaptive Histogram Equalization, the tests conducted above can be repeated to obtain a more comprehensive comparison.

3
AI-BASED APPROACHES FOR CYBERSECURITY

Cybersecurity systems are designed to prevent any hardware or software-based system connected to the internet from the information/privacy loss. These systems can be classified into three categories: misuse, anomaly and hybrid-based detection systems. While misuse-based detection systems analyse concerning characteristics of known intrusions, anomaly-based detection systems classify novel intrusions by detecting divergent data pattern than normal. The hybrid-based detection systems take advantage of the strategies of both misuse and anomaly-based systems.

According to August 2019 McAfee Labs threats report, besides known intrusions, novel intrusions present a great challenge for cybersecurity systems. Since current anomaly-based detection systems are trained on particular datasets with previously known intrusions and heavily biased on the characteristics of known intrusions, there may be a question arising on how reliable these systems are at detecting novel intrusions with totally different characteristics than previous ones or even closer characteristics to normal traffic. Moreover, benchmark datasets generally have unbalanced network traffic data, where some known intrusions are well-represented while others are insufficient for training purposes.

To increase the effectiveness of anomaly detection systems, in theory, one possible solution is to use different datasets or extract new feature representations while training since each benchmark dataset for cybersecurity has a unique internal bias regarding the data collection process of network traffic and variety of intrusion types. However, in practice, adaptability becomes a huge problem at using different datasets due to uniqueness in feature spaces and distribution differences/shifts in network traffic data. Another possible solution is to extract common features using sniffer programs such as Wireshark from benchmark datasets with raw data. However, not all benchmark datasets have raw labeled network traffic data. The last but not least possible solution is to create a new dataset with all known intrusions by collecting traffic data from new distribution and label. However, it is expensive and time-consuming.

Table 3.1 Types of Transfer Learning

2***Type of Transfer Learning***	**Domain**		**Task**	
	Source	***Target***	***Source***	***Target***
Inductive	o	o	x	x
Transductive	x	x	o	o
Unsupervised	x	x	x	x

[a] o: indicates same, x: indicates related but not same

Starting from these possible solutions, this chapter focuses not only on new Capsule-based feature representations to improve detection rates for minority classes but also TL for Cybersecurity to lessen bias issues and generalize similar domains of interests for the first time.

Types of TL are categorized according to the relation between source and target domains as well as the source and target tasks, which are given in Table 3.1 [183]. Since the source and target domains are not similar in cybersecurity, we introduce and conduct an inner dataset TL for Cybersecurity which can be done by using different attack types from the same dataset and projecting them as an intrusion on a different plane for this chapter.

Few studies in the literature have investigated TL for cybersecurity systems until now. One of the very first methods is proposed in [184]. Since different intrusions show distinct patterns in feature space, the method first projects features to a latent space via spectral transformation then uses a variety of shallow classifiers for anomaly detection. On NSL-KDD dataset, performance results without projection are reported as low as random classification while the proposed method achieves much more improved results. Similarly to the study presented in [184], an extended version is proposed in [185]. This version uses kmeans++ based clustering approach instead of manual pre-settings to find the similarity between known and novel intrusions before projection. After obtaining new feature representations of different intrusions on the same latent space, it employs shallow classifiers for anomaly detection. On NSL-KDD dataset, it achieves higher accuracy and sensitivity than the results in [184]. Although it is promising regarding the flexibility of using different feature sets and the ability to map them into the same latent space, it may even lead to tangling data due to linear mapping/transformation. Besides, the works proposed in [184, 185] use different feature sets from only one particular dataset. Therefore, these feature sets are collected under the same network traffic distributions. However, it might not fully represent the real-time scenario due to the variations in background traffic distributions.

In contrast to [184, 185], the method proposed in [186] employs two different benchmark datasets using their common feature space to conduct TL. This method transforms all common features to novel representations via Domain Adaptation Manifold Alignment. Then, different intrusion types in NSL-KDD dataset are trained under intrusion label and tested on Kyoto2006 dataset using SVM. Similar to [186], the method proposed in [187], employs common features between NSL-KDD and CIDD datasets. Results are reported as insufficient due to limited amount -three- of overlapping features between these two datasets. Similarly, the study proposed in [188], uses the same extracted features from raw Netflow data of CTU-13 dataset and transforms all features into new latent space by minimizing the distance between the novel and known malware intrusions. The study described in [189] also employs common features for DoS attacks from UNSW-NB15 and CICIDS2017 as train and test data, respectively. Then, it maps these common features to a latent space via Correlation Alignment and classifies using Siamese NN. Another study presented in [190] uses directly raw malware traffic -which is divided into train and test data with different malware intrusions- and detect anomalies with DNNs. Another TL under domain adaptation is discussed in [191]. It employs bag of samples method using traffic logs for evolving intrusions to have more robust representations. Although it improves recall only using linear transformation via the self-similarity matrix without requiring classification loss function or probability distribution similarity calculation, it may overlook the necessity of new feature space/features of evolving intrusions. As an example, DDoS attacks and low-frequency DDoS attacks can be given.

Unlike other methods, the research proposed in [192] uses pre-trained ResNet-50 backbone for TL, where network traffic data is represented as grayscale images to make data compatible with a two-dimensional domain. Here, while lower layers of a pre-trained ResNet-50 are frozen, higher layers are fine-tuned for cybersecurity.

3.1 Benchmark Datasets and Preprocessing Steps

3.1.1 Benchmark Datasets

Although there are a variety of available traffic databases, most of them are not in use due to reasons of being outdated, only raw data, unrealistic background traffic, lack of novel attacks etc. For this chapter, only frequently-used and robust ones are selected while several datasets such as ADFA, ISCX2012, DEFCON and CDX are excluded due to:

- ADFA database lacks of diversity in terms of attacks.

- ISCX2012 has no realistic background as well as does not include new protocols.
- DEFCON datasets do not have a realistic background and mostly include intrusive traffic.
- CDX dataset lacks of volume as well as attack type diversity.
- DARPA(1999-2000) is outdated in terms of attacks.

For TL scenario, several benchmark datasets are chosen in two steps. As a first step, only frequently-used benchmark datasets are limited to the ones which are labelled and feature-extracted. Details about these datasets are given below and more details can be found at [193].

1. **CICIDS2017**: This dataset consists of full-packet network traffic data including raw network traffic files. Not only it includes a variety of attack types such as flooding, bruteforce, but also it meets the criteria for a reliable dataset in cybersecurity [194, 195].
2. **KDD99**: Similar to CICIDS2017, it consists of full-packet network traffic data including raw network traffic files [196]. Even though it is the most-frequently-used benchmark dataset for cybersecurity and has a huge diversity of attack types such as R2L and U2R, it has reliability issues regarding unbalancing among attack types and the absence of novel attacks.
3. **NSL-KDD**: This dataset is directly created from KDD99 [197]. Although this dataset becomes reliable regarding the balance among attacks by sampling methods as well as deleting duplicated samples, the absence of novel attacks is an issue.
4. **KYOTO**: This dataset is collected from honeypot network traffic [198]. It is often criticized since the honeypot data does not encounter advanced attacks.
5. **UNSW-NB15**: Similar to CICIDS2017 and KDD99, this dataset consists of full-packet network traffic data including raw network traffic files. Even though it has missing samples, it consists of a variety of attacks such as backdoors and reconnaissance [199].

As a second step, these benchmark datasets are investigated by analyzing the diversity of attack types and features to decide the ultimate datasets. After these steps, only CICIDS2017 is chosen for the inner-dataset TL scenario for two reasons. First, it is proposed as the most comprehensive and accurate traffic among benchmark datasets

in [194] and meets a variety of important reliability criteria such as rich feature variety from traditional to novel features, realistic traffic with noisy background and diversity in attack types including recent attack types for a fair evaluation. Second, this scenario is proposed to take full-advantage of feature space and only two benchmark datasets, AWID2018 and CICIDS2017, have a huge diversity of features over 150 while others are below 50. Between these two datasets, CICIDS2017 has eight main attack types while AWID2018 has only three.

Similarly, considering the popularity and recent CNN-based works, NSL-KDD is chosen to evaluate Capsule-based representations and to compare with these CNN-based works.

3.1.2 Preprocessing Steps

For Capsule-based representations, preprocessing steps are used:

- One-hot encoding is applied to categorical features.
- All remaining features are discretized and normalized.

For Inner-dataset TL, standard preprocessing steps for cybersecurity are used:

- One-hot encoding is applied to categorical features.
- All remaining features are discretized and normalized.
- Conversion into two-dimensional input is done for several experiments.

3.2 Experiments and Conclusion

3.2.1 Experimental Setups and Experiments for Capsule-based Representations

One of the recent papers using LeNet [17] is chosen as a baseline model. A variety of preprocessing methods are used with this baseline model except for Capsule Network. For Capsule Network, kernels are modified as 2x2 for the first layer and 1x1 identity kernels for the second layer, respectively. Other hyperparameters for Capsule Network are chosen the same as the previous chapter.

All evaluation results and comparisons with a variety of preprocessing methods are given in Table 3.2 for train size of 494021 and test size of 311029. These evaluations are done for both binary classification (attack-normal) and multiclass classification (DoS, Probe, R2L, U2R and normal).

Table 3.2 Accuracy for LeNet-5 and Capsule Network on NSL-KDD dataset

Multiclass Classification			
Features	**Model Structure**		
	Model	***Input Size***	***Accuracy***
AE(100)	LeNet-5	100	94%
PCA(25)	LeNet-5	361	92.3%
PCA(1)+Categorical	LeNet-5	361	92%
PCA(2)+Categorical	LeNet-5	361	92.1%
PCA(3)+Categorical	LeNet-5	900	92%
All	**Capsule Network**	122	**93,9**%
Binary Classification			
Features	**Model Structure**		
	Model	***Input Size***	***Accuracy***
PCA(1)+Categorical	LeNet-5	361	93.3%
PCA(2)+Categorical	LeNet-5	625	92.9%
PCA(3)+Categorical	LeNet-5	900	92.8%
All	**Capsule Network**	122	**94,2**%

3.2.2 Experimental Setups and Experiments for Inner-dataset TL

Inner-dataset TL is designed to examine the true capability of TL for novel attack detection by taking full advantage of whole feature space extracted from network traffic data. Moreover, this type of TL can be used where a dataset lacks sufficient common feature space with other datasets.

The scenario is done in two steps. As the first step, the model is trained on each attack separately and tested on other attack types one by one. Since each attack generally has quite distinct characteristics, the first step is expanded with the second step. In the second step, some distinct attack types are grouped to train the model separately and tested on other distinct groups of attacks. These groups are formed according to their divergence to one another. To sum up, the basic idea behind this scenario is to train the model on a known attack or a distinct group of known attacks then to test on an unseen attack or a distinct group of unseen attacks. Therefore, the scenario concludes with the prospective advantages and disadvantages of TL for cybersecurity.

Results for inner-dataset TL are given in Table 3.3. Here, the confusion matrix is used as a performance metric to calculate precision, recall, accuracy and f-score are given for binary classification/anomaly detection, where the attack is labeled as 0 while normal data is labeled as 1.

Table 3.3 Confusion Matrices of Inner-Dataset Transfer Learning on CICIDS2017 dataset

Train On	Test On			
	DoS	***DDoS***	***Brute Force FTP***	***Brute Force SSH***
DoS Attacks	-	$\begin{bmatrix} 67229 & 151606 \\ 761 & 1047481 \end{bmatrix}$	$\begin{bmatrix} 0 & 5933 \\ 761 & 1047481 \end{bmatrix}$	$\begin{bmatrix} 0 & 3219 \\ 761 & 1047481 \end{bmatrix}$
	BotNet	***Web Attack***	***Infiltration***	***Heartbleed***
	$\begin{bmatrix} 0 & 1953 \\ 761 & 1047481 \end{bmatrix}$	$\begin{bmatrix} 2 & 2141 \\ 761 & 1047481 \end{bmatrix}$	$\begin{bmatrix} 0 & 36 \\ 761 & 1047481 \end{bmatrix}$	$\begin{bmatrix} 11 & 0 \\ 761 & 1047481 \end{bmatrix}$
	DoS	***DDoS***	***Brute Force FTP***	***Brute Force SSH***
DDoS Attacks	$\begin{bmatrix} 1818 & 191930 \\ 56 & 1048186 \end{bmatrix}$	-	$\begin{bmatrix} 0 & 5933 \\ 56 & 1048186 \end{bmatrix}$	$\begin{bmatrix} 0 & 3219 \\ 56 & 1048186 \end{bmatrix}$
	BotNet	***Web Attack***	***Infiltration***	***Heartbleed***
	$\begin{bmatrix} 0 & 1953 \\ 56 & 1048186 \end{bmatrix}$	$\begin{bmatrix} 0 & 36 \\ 56 & 1048186 \end{bmatrix}$	$\begin{bmatrix} 0 & 36 \\ 56 & 1048186 \end{bmatrix}$	$\begin{bmatrix} 0 & 11 \\ 56 & 1048186 \end{bmatrix}$
	DoS	***DDoS***	***Brute Force FTP***	***Brute Force SSH***
Brute Force FTP Attacks	$\begin{bmatrix} 0 & 193748 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 218835 \\ 0 & 1048242 \end{bmatrix}$	-	$\begin{bmatrix} 0 & 3219 \\ 0 & 1048242 \end{bmatrix}$
	BotNet	***Web Attack***	***Infiltration***	***Heartbleed***
	$\begin{bmatrix} 0 & 1953 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 2143 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 36 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 11 \\ 0 & 1048242 \end{bmatrix}$
	DoS	***DDoS***	***Brute Force FTP***	***Brute Force SSH***
Brute Force SSH Attacks	$\begin{bmatrix} 0 & 193748 \\ 0 & 10482442 \end{bmatrix}$	$\begin{bmatrix} 0 & 218835 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 5933 \\ 0 & 1048242 \end{bmatrix}$	-
	BotNet	***Web Attack***	***Infiltration***	***Heartbleed***
	$\begin{bmatrix} 0 & 1953 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 5933 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 36 \\ 0 & 1048242 \end{bmatrix}$	$\begin{bmatrix} 0 & 11 \\ 0 & 1048242 \end{bmatrix}$

3.2.3 Conclusion

As can be seen in Table 3.2, even using one element of PCA with categorical features is as effective as using 25 elements of PCA. This is either because the image domain contributes features to be more expressive or dataset has a distribution where it can be obtained easily via PCA. Also, evaluation results for AE indicate that AE separate classes better due to its internal non-linear structure compared to PCA. From this point of view, we either employ algorithms that are highly capable of non-linear separation or change pre-processing where all features are more expressive. Owing to the capability of Capsule Network for non-linear modeling, it achieves close to state-of-the-art accuracy for multiclass classification while it achieves 94,2% accuracy for binary classification. For the future work, we will modify convolution filters in Capsule Network and experiment with different preprocessing steps.

In Table 3.3, it can be seen that similar type of attacks such as DoS and DDoS could be trained and tested on behalf of each other while a totally different attack types such as web attacks could not. As a conclusion, promising results are achieved for attacks with close characteristics as train-test pairs. Although some preliminary results show some encouraging results, there are still a variety of ways to test TL and its capabilities for different scenarios. Therefore, for inner dataset TL, we combined groups of divergent attacks to make the training process more generalizable. Although we obtained

slightly better results than Table 3.3, we are still trying to obtain state-of-the-art results to publish.

For the next steps, this chapter will be extended to cross-dataset TL using common features and attack types among datasets which can be seen in Table 3.4, Table 3.5 and Table 3.6, respectively. Since common feature space among datasets is limited, datasets for this scenario will be chosen according to criteria that require the dataset to have at least seven common features with others. To make common feature space reliable enough to conduct TL, k-NN is employed and evaluated only using one feature each time to find the most contributing features for each dataset. Then, the best ten features are listed for each benchmark dataset. Only benchmark datasets that share seven best common features at least will be employed for cross-dataset TL.

Table 3.4 Attack types for benchmark datasets

Attack Types	**Benchmark Datasets**			
	NSL-KDD	***KYOTO***	***CICIDS2017***	***UNSW-NB15***
DoS	o	attack unknowns	o	o
DDoS	x	x	o	x
BruteForce (Password)	x	x	o	x
Injection	x	x	o	x
Infiltration	x	x	o	x
U2R	o	x	x	x
R2L	o	x	x	x
Probe	o	x	x	x
Fuzzlers	x	x	x	o
Analysis	x	x	x	o
Backdoors (Password)	x	x	x	o
Exploits	x	x	x	o
Generic	x	x	x	o
Reconnoissance	x	x	x	o
Shellcode	x	x	x	o
Worms (Malware)	x	x	x	o

Table 3.5 Common features extracted via top-ten selection using KNN

Feature Name	**Datasets**		
	KDD99	***KYOTO***	***UNSW-NB15***
Service	o	o	o
src bytes	o	o	o
dst bytes	o	o	o
count	o	o	x
dst host count	o	o	o
dst host srv count	o	o	o
dst host same src port rate	o	o	o
protocol	o	x	o

[a] o: indicates "exists", x: indicates "do not exist"

Table 3.6 Common features used for Cybersecurity

Feature Description	**Benchmark Datasets**			
	NSL-KDD	***KYOTO***	***CICIDS2017***	***UNSW-NB15***
Duration	O	O	O	O
Protocol Type	0	X	X	O
Service	O	O	X	O
Flag	O	O	O	X
Source Bytes	O	O	O	O
Destination Bytes	O	O	O	O
Count	O	O	X	O
Same Service (SS) Rate	O	O	X	X
Same Error Rate	O	O	X	X
Same Service Error Rate	O	O	X	X
Count of Same Destination (SD) IP	O	O	X	O
Count of SS from SD IP	O	O	X	O
Source Port is from SD IP Rate	O	O	X	X
SYN errors found in Count of SD IP	O	O	O	X
SYN errors when SS from SD IP	O	O	O	X
Start time of connection	X	O	X	O
Finish Time of Connection	X	O	X	O
Source Port Number	X	O	X	O
Source ID Number	X	O	X	O
Destination Port Number	X	O	O	O
Destination IP Number	X	O	X	O

4
RESULTS AND DISCUSSION

In this study, we particularly investigate one reliable model for all in CFS and we used the model in a variety of feature spaces. The challenging point of the study is to recognize and model patterns of data from different feature spaces and problems using only one DL based algorithm. While investigating, we also argued that 1) the advantages and disadvantages of probably one of the next frontier algorithms in Computer Vision, -Capsule-based feature representations-, 2) experimented with the undiscovered areas like more expressive feature representations, 3) experimented TL from another perspective.

Concretely, in Chapter 2, we developed two models based on Capsule representations over the most frequently used benchmark datasets for biometrics. These models not only outperform opponent algorithms using a small number of data samples and obtain high accuracy for the tasks but also show domain-free consistency and reliability. In addition to the powerful feature extraction ability from signatures and finger-veins in lower resolutions than frequently used ones, they are adaptable to any biometric systems that require fast convergence for practical purposes changing capsule sizes. In identification and recognition tasks, these two models show robustness using only low-level feature representations of data while CNN equivalent requires bigger input resolutions and/or more layers. In the verification task, results indicate that low-level features extracted from extreme resolutions are not enough to differentiate highly similar forgery samples from genuine ones. Therefore, either higher resolutions, yet lower than the most frequently used ones, or deeper layers to extract low and mid-level features together are necessary.

In Chapter 3, we experiment with Capsule-representations in cybersecurity. Although the domain is in time and they cannot be used easily without mapping one-dimensional features into two-dimensions, we achieved high detection rates for both minority and majority classes and overall accuracy thanks to capsule representations. Furthermore, results indicate that mapping features into another domain makes features more expressive for cybersecurity. In the anomaly detection

task, results show that Capsule-based representations can be another way of nonlinear mapping such as AEs for cybersecurity. Besides, TL task states the importance of bias-free AI-based model and conducts tests on attacks with distinct characteristics. The results indicate that training with a group of attack instead of a specific attack type is necessary to detect anomalies. However, limited common feature spaces and differences in attack patterns make detection more difficult. Therefore, new scenarios for domain adaptation and nonlinear mapping have great importance for future of cybersecurity.

Finally, several challenges on feature representations are left for the future researches. Those challenges include:

- Instead of supervised learning, discovering and modeling the data using manifold embeddings then learning by clustering to maximize the spatial distance among data samples,
- Conducting research on performance comparison with the use of adversarial attacks,
- Investigating bias-free AI models more to achieve less biased results in the literature,
- Conducting more researches on domain adaptation ways to reuse model and make domain-free approaches more usable in the future.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[2] M.-K. Yoon, S. Mohan, J. Choi, M. Christodorescu, L. Sha, "Learning execution contexts from system call distribution for anomaly detection in smart embedded system," in *Proc. of IoTDI*, 2017, pp. 191–196.

[3] G. R. Kumar, N. Mangathayaru, G. Narsimha, "A novel similarity measure for intrusion detection using gaussian function," *CoRR*, vol. abs/1604.07510, 2016.

[4] A. M. Chandrasekhar K. Raghuveer, "Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers," in *2013 International Conference on Computer Communication and Informatics*, Jan. 2013, pp. 1–7. DOI: 10.1109/ICCCI.2013.6466310.

[5] K. Faraoun, "Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions," *INFOCOMP journal of computer sciences ISSN: 1807-4545*, vol. 5, pp. 28–36, Sep. 2006.

[6] A. P. Muniyandi, R. Rajeswari, R. Rajaram, "Network anomaly detection by cascading k-means clustering and c4.5 decision tree algorithm," 2012.

[7] Z. Muda, W. Mohamed, m. n. Sulaiman, N. Udzir, "K-means clustering and naive bayes classification for intrusion detection," *Journal of IT in Asia*, vol. 4, pp. 13–25, Apr. 2016. DOI: 10.33736/jita.45.2014.

[8] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[9] B. Abolhasanzadeh, "Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features," *2015 7th Conference on Information and Knowledge Technology (IKT)*, pp. 1–5, 2015.

[10] M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. K. Tupakula, "Autoencoder-based feature learning for cyber security applications," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3854–3861, 2017.

[11] A. Javaid, Q. Niyaz, W. Sun, M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS)*, ser. BICT'15, New York City, United States: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 21–26, ISBN: 9781631901003. DOI: 10.4108/eai.3-12-2015.2262516. [Online]. Available: https://doi.org/10.4108/eai.3-12-2015.2262516.

[12] N. Shone, T. N. Ngoc, V. D. Phai, Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018, ISSN: 2471-285X. DOI: 10.1109/TETCI.2017.2772792.

[13] X. Li, W. Chen, Q. Zhang, L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection," *Computers Security*, vol. 95, p. 101 851, Apr. 2020.

[14] L. R. Parker, P. D. Yoo, A. T. Asyhari, L. Chermak, Y. Jhi, K. Taha, "Demise: Interpretable deep extraction and mutual information selection techniques for iot intrusion detection," in *ARES '19*, 2019.

[15] Y. Yu, J. Long, Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders," *Security and Communication Networks*, vol. 2017, 4184196:1–4184196:10, 2017.

[16] S. Park, M. Kim, S. Lee, "Anomaly detection for http using convolutional autoencoders," *IEEE Access*, vol. 6, pp. 70 884–70 901, 2018.

[17] Y. Xiao, C. Xing, T. Zhang, Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42 210–42 219, 2019.

[18] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, M. C. Chan, "Gee: A gradient-based explainable variational autoencoder for network anomaly detection," *2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 91–99, 2019.

[19] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz, "Learning latent distribution for distinguishing network traffic in intrusion detection system," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[20] S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun, K. Taha, "Impact: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction," *IEEE Access*, vol. 8, pp. 65 520–65 529, 2020.

[21] Y. Yang, K. Zheng, B. Wu, Y. Yang, X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42 169–42 184, 2020.

[22] V. L. Cao, M. Nicolau, J. McDermott, "A hybrid autoencoder and density estimation model for anomaly detection," in *PPSN*, 2016.

[23] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D.-k. Cho, H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *ICLR*, 2018.

[24] C. Ieracitano, A. Adeel, F. C. Morabito, A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219315759.

[25] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[26] S. Shin, I. Lee, C. Choi, "Anomaly dataset augmentation using the sequence generative models," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1143–1148.

[27] B. Dowoo, Y. Jung, C. Choi, "Pcapgan: Packet capture file generator by style-based generative adversarial networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1149–1154.

[28] L. Han, Y. Sheng, X. Zeng, "A packet-length-adjustable attention model based on bytes embedding using flow-wgan for smart cybersecurity," *IEEE Access*, vol. 7, pp. 82 913–82 926, 2019.

[29] T. Schlegl, P. Seeböck, S. Waldstein, U. Schmidt-Erfurth, G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," Mar. 2017, pp. 146–157, ISBN: 978-3-319-59049-3. DOI: 10.1007/978-3-319-59050-9_12.

[30] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *ArXiv*, vol. abs/1802.06222, 2018.

[31] T. Kohonen, "The self-organizing map," *Proc. of IEEE*, vol. 78, pp. 1464–1480, Oct. 1990.

[32] H. Gunes Kayacik, A. Nur Zincir-Heywood, M. I. Heywood, "A hierarchical SOM-based intrusion detection system," *Eng. Appl. Artif. Intell.*, vol. 20, no. 4, pp. 439–451, Jun. 2007.

[33] A. Ortiz, E. Hoz, E. De la Hoz, J. Ortega, B. Prieto, "Pca filtering and probabilistic som for network intrusion detection," *Neurocomputing*, Sep. 2014.

[34] O. Depren, M. Topallar, E. Anarim, M. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert Syst. Appl.*, vol. 29, pp. 713–722, Nov. 2005. DOI: 10.1016/j.eswa.2005.05.002.

[35] M. Bahrololum M. Khaleghi, "Anomaly intrusion detection system using Gaussian Mixture Model," in *Proc. of ICCIT*, 2008, pp. 1162–1167.

[36] S. Parsazad, E. Saboori, A. Allahyar, "Fast feature reduction in intrusion detection datasets," in *Proc. of MIPRO*, 2012, pp. 1023–1029.

[37] P. Casas, J. Mazel, P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Comput. Commun.*, vol. 35, pp. 772–783, Apr. 2012.

[38] W.-C. Lin, S.-W. Ke, C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, Jan. 2015. DOI: 10.1016/j.knosys.2015.01.009.

[39] W. Meng, W. Li, L.-F. Kwok, "Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection," *Sec. and Commun. Netw.*, vol. 8, no. 18, pp. 3883–3895, Dec. 2015.

[40] S. Mukherjee N. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, Dec. 2012. DOI: 10.1016/j.protcy.2012.05.017.

[41] D. M. Farid M. Z. Rahman, "Learning intrusion detection based on adaptive bayesian algorithm," in *Proc. of ICCIT*, 2008, pp. 652–656.

[42] M. Albayati B. Issac, "Analysis of intelligent classifiers and enhancing the detection accuracy for intrusion detection system," *Int. J. Comput. Intell. Syst.*, vol. 8, pp. 841–853, 2015.

[43] L. Koc, T. A. Mazzuchi, S. Sarkani, "A network intrusion detection system based on a hidden naıve bayes multiclass classifier," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13 492–13 500, Dec. 2012, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2012.07.009. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.07.009.

[44] Y. Wahba, E. ElSalamouny, G. ElTaweel, "Improving the performance of multi-class intrusion detection systems using feature reduction," *ArXiv*, vol. abs/1507.06692, 2015.

[45] D. Barbara, N. Wu, S. Jajodia, "Detecting novel network intrusions using bayes," Apr. 2001. DOI: 10.1137/1.9781611972719.28.

[46] S. R. Safavian D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.

[47] P.-F. Marteau, "Sequence covering for efficient host-based intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 994–1006, 2019.

[48] H. G. Kayacik, A. N. Zincir-Heywood, M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99," in *Proc. of PST*, 2005.

[49] C. Xiang, P. C. Yong, L. S. Meng, "Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees," *Pattern Recogn. Lett.*, vol. 29, no. 7, pp. 918–924, May 2008, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.01.008. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2008.01.008.

[50] H.-J. Zimmermann, *Fuzzy Set Theory&Mdash;and Its Applications (3rd Ed.)* Kluwer Academic Publishers, 1996.

[51] A. Tajbakhsh, M. Rahmati, A. Mirzaei, "Intrusion detection using Fuzzy association rules," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 462–469, Mar. 2009.

[52] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abro, M. L. Proena, "Network anomaly detection system using Genetic Algorithm and Fuzzy Logic," *Expert Syst. Appl.*, vol. 92, no. C, pp. 390–402, Feb. 2018.

[53] S. Elhag, A. Fernández, A. Altalhi, S. Alshomrani, F. Herrera, "A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems," *Soft Comput.*, vol. 23, no. 4, pp. 1321–1336, Feb. 2019.

[54] S. Kamalanathan, M. Karuppiah, S. Lakshmanan, S. H. Islam, M. Hassan, G. Fortino, K.-K. R. Choo, "Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs," *Inform. Sciences*, vol. 497, May 2019.

[55] J. Liu, Z. Wuxia, Z. Tang, Y. Xie, T. Ma, J. Zhang, G. Zhang, J. Niyoyita, "Adaptive intrusion detection via GA-GOGMM-based pattern learning with fuzzy rough set-based attribute selection," *Expert Syst. Appl.*, vol. 139, p. 112 845, Jul. 2019.

[56] G. Wang, J. Hao, J. Ma, L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and Fuzzy Clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, Sep. 2010.

[57] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd. Prentice Hall PTR, 1998.

[58] W. Tian J. Liu, "A new network intrusion detection identification model research," in *Proc. of CAR*, vol. 2, 2010, pp. 9–12.

[59] Y. Yao, Y. Wei, F. Gao, Y. Yu, "Anomaly intrusion detection approach using hybrid mlp/cnn neural network," *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 1095–1102, 2006.

[60] A. Saied, R. E. Overill, T. Radzik, "Detection of known and unknown ddos attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[61] C. Cortes V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[62] X. Bao, T. Xu, H. Hou, "Network intrusion detection based on Support Vector Machine," in *Proc. of MASS*, 2009, pp. 1–4.

[63] K. Zheng, X. Qian, P. Wang, "Dimension reduction in intrusion detection using manifold learning," in *Proc. of CIS*, vol. 2, 2009, pp. 464–468.

[64] B.-j. Kim I. K. Kim, "Kernel based intrusion detection system," vol. 2005, Feb. 2005, pp. 13–18, ISBN: 0-7695-2296-3. DOI: 10.1109/ICIS.2005.78.

[65] G. Xiaoqing, G. Hebin, C. Luyi, "Network intrusion detection method based on agent and SVM," in *Proc. of ICIME*, 2010, pp. 399–402.

[66] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, pp. 424–430, 2012.

[67] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[68] J. Zhang, M. Zulkernine, A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008.

[69] P.-F. Marteau, S. Soheily-Khah, N. Béchet, "Hybrid isolation forest - application to intrusion detection," *ArXiv*, vol. abs/1705.03800, 2017.

[70] A. Tesfahun D. L. Bhaskari, "Intrusion detection using random forests classifier with SMOTE and feature reduction," in *Proc. of CUBE*, Nov. 2013, pp. 127–132.

[71] R. Elbasiony, E. A. Sallam, T. E. Eltobely, M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," 2013.

[72] J. Li, Z. Zhao, R. Li, "Machine learning-based IDS for software-defined 5g network," *IET Networks*, vol. 7, no. 2, pp. 53–60, 2018.

[73] A. Madbouly, A. Gody, T. Barakat, "Relevant feature selection model using data mining for intrusion detection system," *Int. Journal of Engineering Trends and Technology*, vol. 9, Mar. 2014.

[74] S. Aljawarneh, M. Aldwairi, M. Yasin, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, Mar. 2017. DOI: 10.1016/j.jocs.2017.03.006.

[75] W. Hu, W. Hu, S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, Apr. 2008. DOI: 10.1109/TSMCB.2007.914695.

[76] S. Chebrolu, A. Abraham, J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Comput. Secur.*, vol. 24, no. 4, pp. 295–307, Jun. 2005.

[77] M. Gudadhe, P. Prasad, L. Kapil Wankhade, "A new data mining based network intrusion detection model," in *Proc. of ICCCT*, 2010, pp. 731–735.

[78] H. Saxena V. Richariya, "Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain," *Int. J. Comput. Appl.*, vol. 98, pp. 25–29, Jul. 2014.

[79] Y. Gong, S. Mabu, C. Chen, Y. Wang, K. Hirasawa, "Intrusion detection system combining misuse detection and anomaly detection using Genetic network programming," in *Proc. of ICCAS-SICE*, 2009, pp. 3463–3467.

[80] R. Elhefnawy, H. Abounaser, A. Badr, "A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks," *IEEE Access*, vol. 8, pp. 98 218–98 233, 2020.

[81] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, May 2015. DOI: 10.1038/nature14539.

[82] Z. Li, Z. Qin, K. Huang, X. Yang, S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *ICONIP*, 2017.

[83] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, F. Iqbal, "Malware classification with deep convolutional neural networks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018, pp. 1–5.

[84] T. Kim, S. C. Suh, H. Kim, J. Kim, J. Kim, "An encoding technique for cnn-based network anomaly detection," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 2960–2965.

[85] R. Blanco, P. Malagón, J. J. Cilla, J. M. Moya, "Multiclass network attack classifier using cnn tuned with genetic algorithms," in *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2018, pp. 177–182.

[86] K. Wu, Z. Chen, W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50 850–50 859, 2018.

[87] S. Z. Lin, Y. Shi, Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[88] L. Nie, Z. Ning, X. Wang, X. Hu, Y. Li, J. Cheng, "Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.

[89] L. Yong Z. Bo, "An intrusion detection model based on multi-scale cnn," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019, pp. 214–218.

[90] G. Feng, B. Li, M. Yang, Z. Yan, "V-cnn: Data visualizing based convolutional neural network," in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Sep. 2018, pp. 1–6. DOI: 10.1109/ICSPCC.2018.8567781.

[91] S.-N. Nguyen, V.-Q. Nguyen, J. Choi, K. Kim, "Design and implementation of intrusion detection system using convolutional neural network for dos detection," in *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ser. ICMLSC '18, Phu Quoc Island, Viet Nam: Association for Computing Machinery, 2018, pp. 34–38, ISBN: 9781450363365. DOI: 10.1145/3184066.3184089. [Online]. Available: https://doi.org/10.1145/3184066.3184089.

[92] Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, "Flowguard: An intelligent edge defense mechanism against iot ddos attacks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[93] H. Yang F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64 366–64 374, 2019.

[94] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, M. Zhu, "Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[95] M. Ito H. Iyatomi, "Web application firewall using character-level convolutional neural network," in *2018 IEEE 14th International Colloquium on Signal Processing Its Applications (CSPA)*, Mar. 2018, pp. 103–106. DOI: 10.1109/CSPA.2018.8368694.

[96] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362, ISBN: 026268053X.

[97] H. Liu, B. Lang, M. Liu, H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowl.-Based Syst.*, vol. 163, Sep. 2018.

[98] S. Lv, J. J. Wang, Y. Yang, J. Liu, "Intrusion prediction with system-call sequence-to-sequence model," *IEEE Access*, vol. 6, pp. 71 413–71 421, 2018.

[99] S. Hochreiter J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[100] G. Kim, H. Yi, J. Lee, Y. Paek, S. Yoon, "Lstm-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," *ArXiv*, vol. abs/1611.01726, 2016.

[101] F. Jiang, Y. Fu, B. B. Gupta, Y. Liang, S. Rho, F. Lou, F. Meng, Z. Tian, "Deep learning based multi-channel intelligent attack detection for data security," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 204–212, 2020.

[102] O. Alkadi, N. Moustafa, B. Turnbull, K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[103] R. Dong, X. Li, Q. Zhang, H. Yuan, "Network intrusion detection model based on multivariate correlation analysis – long short-time memory network," *IET Information Security*, vol. 14, no. 2, pp. 166–174, 2020.

[104] R. Vinayakumar, K. P. Soman, P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[105] K. Jiang, W. Wang, A. Wang, H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32 464–32 476, 2020.

[106] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002, ISSN: 0899-7667. DOI: 10.1162/089976602760128018. [Online]. Available: https://doi.org/10.1162/089976602760128018.

[107] K. Alrawashdeh C. Purdy, "Reducing calculation requirements in fpga implementation of deep learning algorithms for online anomaly intrusion detection," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, 2017, pp. 57–62.

[108] S. Seo, S. Park, J. Kim, "Improvement of network intrusion detection accuracy by using restricted boltzmann machine," in *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2016, pp. 413–417.

[109] Nguyen Thanh Van, Tran Ngoc Thinh, Le Thanh Sach, "An anomaly-based network intrusion detection system using deep learning," in *2017 International Conference on System Science and Engineering (ICSSE)*, 2017, pp. 210–214.

[110] M. Z. Alom T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, 2017, pp. 63–69.

[111] R. Salakhutdinov G. Hinton, "Deep boltzmann machines," *Proceedings of AISTATS 2009*, vol. 5, 448â455, Jan. 2009.

[112] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," 2011.

[113] M. Z. Alom, V. Bontupalli, T. M. Taha, "Intrusion detection using deep belief networks," in *2015 National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 339–344.

[114] F. Qu, J. Zhang, Z. Shao, S. Qi, "An intrusion detection model based on deep belief network," in *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*, ser. ICNCC 2017, Kunming, China: Association for Computing Machinery, 2017, pp. 97–101, ISBN: 9781450353663. DOI: 10.1145/3171592.3171598. [Online]. Available: https://doi.org/10.1145/3171592.3171598.

[115] D. Liang P. Pan, "Research on intrusion detection based on improved dbn-elm," in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2019, pp. 495–499.

[116] G. Zhao, C. Zhang, L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1, 2017, pp. 639–642.

[117] K. Alrawashdeh C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 195–200.

[118] N. Gao, L. Gao, Q. Gao, H. Wang, "An intrusion detection model based on deep belief networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, 2014, pp. 247–252.

[119] R. S. Sutton A. G. Barto, *Introduction to Reinforcement Learning*, 1st. Cambridge, MA, USA: MIT Press, 1998.

[120] N. Sengupta, J. Sen, J. Sil, M. Saha, "Designing of on line intrusion detection system using rough set theory and q-learning algorithm," *Neurocomputing*, vol. 111, pp. 161–168, 2013.

[121] I. Sobel, "Camera models and machine perception," 1970.

[122] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[123] P. Dollar, Zhuowen Tu, S. Belongie, "Supervised learning of edges and object boundaries," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1964–1971.

[124] A. R. Rao G. L. Lohse, "Identifying high level features of texture perception," *CVGIP Graph. Model. Image Process.*, vol. 55, pp. 218–233, 1993.

[125] G. E. Sotak K. L. Boyer, "The laplacian-of-gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output," *Comput. Vis. Graph. Image Process.*, vol. 48, pp. 147–189, 1989.

[126] R. A. Young, "The gaussian derivative model for spatial vision: I. retinal mechanisms.," *Spatial vision*, vol. 2 4, pp. 273–93, 1987.

[127] N. Dalal B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, 886–893 vol. 1.

[128] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004. DOI: 10.1023/B%3AVISI.0000029664.99615.94.

[129] H. Bay, A. Ess, T. Tuytelaars, L. [Gool], "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008, Similarity Matching in Computer Vision and Multimedia, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2007.09.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314207001555.

[130] Z. Zhang, X. Liu, Y. Cui, "Multi-phase offline signature verification system using deep convolutional generative adversarial networks," in *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, 2016, pp. 103–107.

[131] S. Wang S. Jia, "Signature handwriting identification based on generative adversarial networks," *Journal of Physics: Conference Series*, vol. 1187, no. 4, p. 042 047, Apr. 2019. DOI: 10.1088/1742-6596/1187/4/042047. [Online]. Available: https://doi.org/10.1088%2F1742-6596%2F1187%2F4%2F042047.

[132] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, J. Ortega-Garcia, "Exploring recurrent neural networks for on-line handwritten signature biometrics," *IEEE Access*, vol. 6, pp. 5128–5138, 2018.

[133] L. G. Hafemann, R. Sabourin, L. E. S. de Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognit.*, vol. 70, pp. 163–176, 2017.

[134] L. G. Hafemann, R. Sabourin, L. S. Oliveira, "Meta-learning for fast classifier adaptation to new users of signature verification systems," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1735–1745, 2020.

[135] R. K. Mohapatra, K. Shaswat, S. Kedia, "Offline handwritten signature verification using cnn inspired by inception v1 architecture," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 263–267.

[136] C. Li, F. Lin, Z. Wang, G. Yu, L. Yuan, H. Wang, "Deephsv: User-independent offline signature verification using two-channel cnn," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 166–171.

[137] R. Das, E. Piciucco, E. Maiorana, P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, Feb. 2019, ISSN: 1556-6013. DOI: 10.1109/TIFS.2018.2850320.

[138] H. Qin M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017, ISSN: 1556-6013. DOI: 10.1109/TIFS.2017.2689724.

[139] H. G. Hong, M. B. Lee, K. R. Park, "Convolutional neural network-based finger-vein recognition using nir image sensors," *Sensors (Basel, Switzerland)*, vol. 17, no. 6, Jun. 2017, ISSN: 1424-8220. DOI: 10.3390/s17061297. [Online]. Available: http://europepmc.org/articles/PMC5492434.

[140] R. Raghavendra C. Busch, "Presentation attack detection algorithms for finger vein biometrics: A comprehensive study," in *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, Nov. 2015, pp. 628–632. DOI: 10.1109/SITIS.2015.74.

[141] F. Radzi, M. Khalil-Hani, R. Bakhteri, "Finger-vein biometric identification using convolutional neural network," *TURKISH JOURNAL OF ELECTRICAL ENGINEERING COMPUTER SCIENCES*, vol. 24, pp. 1863–1878, Jan. 2016. DOI: 10.3906/elk-1311-43.

[142] P. Tome, M. Vanoni, S. Marcel, "On the vulnerability of finger vein recognition to spoofing," in *2014 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sep. 2014, pp. 1–10.

[143] P. Tome S. Marcel, "On the vulnerability of palm vein recognition to spoofing attacks," in *2015 International Conference on Biometrics (ICB)*, May 2015, pp. 319–325. DOI: 10.1109/ICB.2015.7139056.

[144] T. Cohen M. Welling, "Steerable cnns," *ArXiv*, vol. abs/1612.08498, 2017.

[145] T. S. Cohen M. Welling, "Group equivariant convolutional networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16, New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.

[146] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7168–7177, 2017.

[147] D. E. Worrall G. J. Brostow, "Cubenet: Equivariance to 3d rotation and translation," *ArXiv*, vol. abs/1804.04458, 2018.

[148] P. Follmann T. Bottger, "A rotationally-invariant convolution module by feature map back-rotation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 784–792.

[149] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 305–11 312.

[150] J. Masci, D. Boscaini, M. M. Bronstein, P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, ser. ICCVW '15, USA: IEEE Computer Society, 2015, pp. 832–840, ISBN: 9781467397117. DOI: 10.1109/ICCVW.2015.112. [Online]. Available: https://doi.org/10.1109/ICCVW.2015.112.

[151] S. Suwajanakorn, N. Snavely, J. Tompson, M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, Eds., 2018, pp. 2063–2074. [Online]. Available: http://papers.nips.cc/paper/7476-discovery-of-latent-3d-keypoints-via-end-to-end-geometric-reasoning.

[152] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5425–5434, 2017.

[153] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[154] J. Snell, K. Swersky, R. S. Zemel, "Prototypical networks for few-shot learning," *ArXiv*, vol. abs/1703.05175, 2017.

[155] P. Mettes, E. van der Pol, C. G. M. Snoek, "Hyperspherical prototype networks," in *NeurIPS*, 2019.

[156] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 0018-9219. DOI: 10.1109/5.726791.

[157] S. Sabour, N. Frosst, G. E. Hinton, "Dynamic routing between capsules," *CoRR*, vol. abs/1710.09829, 2017. arXiv: 1710.09829. [Online]. Available: http://arxiv.org/abs/1710.09829.

[158] L. G. Hafemann, R. Sabourin, L. S. Oliveira, "Offline handwritten signature verification — literature review," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov. 2017, pp. 1–8. DOI: 10.1109/IPTA.2017.8310112.

[159] M. K. Kalera, S. N. Srihari, A. Xu, "Offline signature verification and identification using distance statistics," *IJPRAI*, vol. 18, pp. 1339–1360, 2004.

[160] F. Vargas, M. Ferrer, C. Travieso, J. Alonso, "Off-line handwritten signature gpds-960 corpus," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, Sep. 2007, pp. 764–768. DOI: 10.1109/ICDAR.2007.4377018.

[161] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J. .-.-. Igarza, C. Vivaracho, D. Escudero, Q. .-.-. Moro, "Mcyt baseline corpus: A bimodal biometric database," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, Dec. 2003, ISSN: 1350-245X. DOI: 10.1049/ip-vis:20031078.

[162] L. Hiryanto, A. R. Yohannis, T. Handhayani, "Hand signature and handwriting recognition as identification of the writer using gray level cooccurrence matrix and bootstrap," in *2017 Intelligent Systems Conference (IntelliSys)*, Sep. 2017, pp. 1103–1110. DOI: 10.1109/IntelliSys.2017.8324267.

[163] M. B. Yilmaz B. Yanikoglu, "Score level fusion of classifiers in off-line signature verification," *Information Fusion*, vol. 32, Feb. 2016. DOI: 10.1016/j.inffus.2016.02.003.

[164] J. Ruiz-Del-Solar, C. Devia, P. Loncomilla, F. Concha, "Offline signature verification using local interest points and descriptors," in *Proceedings of the 13th Iberoamerican Congress on Pattern Recognition: Progress in Pattern Recognition, Image Analysis and Applications*, ser. CIARP '08, Havana, Cuba: Springer-Verlag, 2008, pp. 22–29, ISBN: 9783540859192. DOI: 10.1007/978-3-540-85920-8_3. [Online]. Available: https://doi.org/10.1007/978-3-540-85920-8_3.

[165] M. B. Yilmaz, B. Yanikoglu, C. Tirkaz, A. Kholmatov, "Offline signature verification using classifier combination of hog and lbp features," in *2011 International Joint Conference on Biometrics (IJCB)*, Oct. 2011, pp. 1–7. DOI: 10.1109/IJCB.2011.6117473.

[166] L. G. Hafemann, R. Sabourin, L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163–176, 2017, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.05.012.

[167] M. B. Yilmaz K. Öztürk, "Hybrid user-independent and user-dependent offline signature verification with a two-channel cnn," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 639–6398, 2018.

[168] V. Ruiz, I. Linares, A. Sanchez, J. F. Velez, "Off-line handwritten signature verification using compositional synthetic generation of signatures and siamese neural networks," *Neurocomputing*, vol. 374, pp. 30–41, 2020, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.09.041.

[169] S. Pal, A. Alaei, U. Pal, M. Blumenstein, "Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 72–77.

[170] L. G. Hafemann, R. Sabourin, L. Oliveira., "Characterizing and evaluating adversarial examples for offline handwritten signature verification," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019, ISSN: 1556-6013. DOI: 10.1109/TIFS.2019.2894031.

[171] O. Mersa, F. Etaati, S. Masoudnia, B. N. Araabi, "Learning Representations from Persian Handwriting for Offline Signature Verification, a Deep Transfer Learning Approach," *arXiv e-prints*, Feb. 2019. arXiv: 1903.06249 [cs.CV].

[172] V. L. F. Souza, A. L. I. Oliveira, R. Sabourin, "A writer-independent approach for offline signature verification using deep convolutional neural networks features," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct. 2018, pp. 212–217. DOI: 10.1109/BRACIS.2018.00044.

[173] L. Yang, G. Yang, Y. Yin, L. Zhou, "A survey of finger vein recognition," in *Biometric Recognition*, Z. Sun, S. Shan, H. Sang, J. Zhou, Y. Wang, W. Yuan, Eds., Cham: Springer International Publishing, 2014, pp. 234–243, ISBN: 978-3-319-12484-1.

[174] S.-I. Khalid, F. Radzi, N. Mohd Saad, N. Abdul Hamid, W. H. Bin Mohd Saad, M. Saad, F. Kejuruteraan, E. Dan, K. Komputer, "A review of finger-vein biometrics identification approaches," May 2016.

[175] Y. Lu, S. Xie, S. Wu, "Exploring competitive features using deep convolutional neural network for finger vein recognition," *IEEE Access*, vol. PP, pp. 1–1, Mar. 2019. DOI: 10.1109/ACCESS.2019.2902429.

[176] X. Li, d. Huang, Y. Wang, "Comparative study of deep learning methods on dorsal hand vein recognition," Oct. 2016, pp. 296–306, ISBN: 978-3-319-46653-8. DOI: 10.1007/978-3-319-46654-5_33.

[177] H. G. Hong, M. B. Lee, K. R. Park, "Convolutional neural network-based finger-vein recognition using nir image sensors," *Sensors*, vol. 17, no. 6, 2017, ISSN: 1424-8220. DOI: 10.3390/s17061297. [Online]. Available: http://www.mdpi.com/1424-8220/17/6/1297.

[178] Y. Yin, L. Liu, X. Sun, "Sdumla-hmt: A multimodal biometric database," in *CCBR*, 2011.

[179] B. T. Ton R. N. J. Veldhuis, "A high quality finger vascular pattern dataset collected using a custom designed capturing device," in *2013 International Conference on Biometrics (ICB)*, Jun. 2013, pp. 1–5. DOI: 10.1109/ICB.2013.6612966.

[180] A. Kumar Y. Zhou, "Human identification using finger images," *Trans. Img. Proc.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012, ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2171697. [Online]. Available: https://doi.org/10.1109/TIP.2011.2171697.

[181] Y. Lu, S. Juan Xie, S. Yoon, J. Yang, D. Sun Park, "Robust finger vein roi localization based on flexible segmentation," *Sensors (Basel, Switzerland)*, vol. 13, pp. 14 339–14 366, Nov. 2013. DOI: 10.3390/s131114339.

[182] N. Miura, A. Nagasaka, T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Machine Vision and Applications*, vol. 15, no. 4, pp. 194–203, Oct. 2004, ISSN: 1432-1769. DOI: 10.1007/s00138-004-0149-2. [Online]. Available: https://doi.org/10.1007/s00138-004-0149-2.

[183] S. J. Pan Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[184] J. Zhao, S. Shetty, J. W. Pan, "Feature-based transfer learning for network security," in *Proc. of MILCOM*, 2017, pp. 17–22.

[185] J. Zhao, S. Shetty, J. Wei Pan, C. Kamhoua, K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP Journal on Information Security*, vol. 2019, Dec. 2019.

[186] Z. Taghiyarrenani, A. Fanian, E. Mahdavi, A. Mirzaei, H. Farsi, "Transfer learning based intrusion detection," in *Proc. of ICCKE*, Oct. 2018, pp. 92–97.

[187] R. Ahmadi, R. D. Macredie, A. Tucker, "Intrusion detection using transfer learning in machine learning classifiers between non-cloud and cloud datasets," in *Proc. of IDEAL*, Springer International Publishing, 2018, pp. 556–566.

[188] R. Kozik, M. Choraś, J. Keller, "Balanced efficient lifelong learning (b-ella) for cyber attack detection," *Journal of Universal Computer Science*, vol. 25, pp. 2–15, Jan. 2019.

[189] C. Kneale K. Sadeghi, "Semisupervised adversarial neural networks for cyber security transfer learning," *ArXiv*, vol. abs/1907.11129, 2019.

[190] I. Rosenberg, G. Sicard, E. David, "End-to-end deep neural networks and transfer learning for automatic analysis of nation-state malware," *Entropy*, vol. 20, p. 390, May 2018. DOI: 10.3390/e20050390.

[191] K. Bartos M. Sofka, "Robust representation for domain adaptation in network security," in *ECML/PKDD*, 2015.

[192] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, P. de Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2017, pp. 1011–1014. DOI: 10.1109/ICMLA.2017.00-19.

[193] D. Gümüşbaş, T. Yıldırım, A. Genovese, F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Systems Journal*, pp. 1–15, 2020.

[194] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP*, 2018.

[195] C. I. for Cybersecurity, *Intrusion detection evaluation dataset (cicids2017)*, https://www.unb.ca/cic/datasets/ids-2017.html, 2017.

[196] K. C. 1999, *Intrusion detection evaluation dataset (cicids2017)*, http://www.kdd.org/kdd-cup/view/kdd-cup-1999, 1999.

[197] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. of CISDA*, 2009, pp. 1–6.

[198] K. University, *Traffic data from kyoto university's honeypots*, http://www.takakura.com/Kyoto_data, 2015.

[199] N. Moustafa J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. of MilCIS*, 2015, pp. 1–6.

PUBLICATIONS FROM THE THESIS

Contact Information: gumusbasdilara@gmail.com

Papers

1. D. Gümüşbaş, T. Yıldırım, A. Genovese, F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Systems Journal*, pp. 1–15, 2020.

2. D. Gumusbas, T. Yıldırım, "Offline signature identification and verification based on capsule representations," *Cybernetics and Information Technologies*, vol. 20, 2020.

Conference Papers

1. D. Gumusbas, T. Yildirim, "Offline signature identification and verification using capsule network," in *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2019, pp. 1–5.

2. D. Gumusbas, T. Yildirim, M. Kocakulak, N. Acir, "Capsule network for finger-vein-based biometric identification," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 437–441.