ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DISTRIBUTED ANOMALY-BASED INTRUSION DETECTION SYSTEM FOR IOT ENVIRONMENT USING BLOCKCHAIN TECHNOLOGY

M.Sc. THESIS

Nouha HEJAZI

Applied Informatics Department

Cybersecurity Engineering and Cryptography Program

FEBRUARY 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DISTRIBUTED ANOMALY-BASED INTRUSION DETECTION SYSTEM FOR IOT ENVIRONMENT USING BLOCKCHAIN TECHNOLOGY

M.Sc. THESIS

Nouha HEJAZI
(707191006)

Applied Informatics Department

Cybersecurity Engineering and Cryptography Program

Thesis Advisor: Assoc. Prof. Dr. Enver ÖZDEMİR

FEBRUARY 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DAĞITILMIŞ ANOMALİ TABANLI SALDIRI TESPİT SİSTEMİ BLOCKCHAIN TEKNOLOJİSİ KULLANILAN IOT ORTAMI İÇİN

YÜKSEK LİSANS TEZİ

Nouha HEJAZI
(707191006)

Bilişim Uygulamaları Anabilim Dalı

Bilgi Güvenliği Mühendisliği ve Kriptografi Programı

Tez Danışmanı: Doç.Dr. Enver ÖZDEMİR

ŞUBAT 2022

Nouha HEJAZI, a **M.Sc.** student of ITU Graduate School student ID **707191006**, successfully defended the thesis entitled "DISTRIBUTED ANOMALY-BASED INTRUSION DETECTION SYSTEM FOR IOT ENVIRONMENT USING BLOCKCHAIN TECHNOLOGY", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Jury Members : **Doç. Dr. Enver Özdemir** ....
İstanbul Teknik Üniversitesi

Doç. Dr. Ergün Yaraneri ....
İstanbul Teknik Üniversitesi

Dr. Öğr. Üyesi Elif Öztaş .
Karamanoğlu Mehmetbey Üniversitesi

Date of Submission : 19 Janruary 2022
Date of Defense : 4 February 2022

To my baby Omer, my lovely family, and teachers,

FOREWORD

I would like to express my gratitude and give thanks to my supervisor Assoc. Prof. Dr. Enver Özdemir for his guidance and patient supervision as my performance and speed of achievement while working on this research were affected as a result of my changing social and health conditions during my pregnancy with my first child. He was always understanding, motivating, appreciating, and supporting in all possible ways which helped me a lot to finish my work.

I am also grateful to my spouse Dr.Mohammad Ghaith, and my parents Dr. Abdulhameed and Dr.Maha for their moral support and motivation during this journey.

February 2022

Nouha HEJAZI
(Information Systems Engineer)

TABLE OF CONTENTS

ABBREVIATIONS

AE	: Autoencoder
ANN	: Artificial Neural Network
AR	: Association Rule
BiGAN	: Bidirectional Generative Adversarial Network
BN	: Bayesian Network
CIDS	: Collaborative Intrusion Detection System
DoS	: Denial of Service
DT	: Decision Trees
EGBAD	: Efficient GAN-Based Anomaly Detection
EL	: Ensemble Learning
FL	: Fuzzy Logic
GA	: Genetic Algorithm
GAN	: Generative Adversarial Network
IDS	: Intrusion Detection System
IoT	: Internet of Things
IP	: Identification Protocol
MC	: Markov Chains
MD-GAN	: Multi-Discriminator Generative Adversarial
ML	: Machine Learning
NB	: Naïve Bayes
NN	: Neural Network
ReLU	: Rectified Linear Unit
RFID	: Radio Frequency Identification
RPL	: Routing Protocol for Low-Power and Lossy Networks
RSS	: Received Signal Strength
SVM	: Support Vector Machine
TCP	: Transfer Control Protocol

LIST OF FIGURES

DISTRIBUTED ANOMALY-BASED INTRUSION DETECTION SYSTEM FOR IOT ENVIRONMENT USING BLOCKCHAIN TECHNOLOGY

SUMMARY

The IoT world is growing rapidly. One of the most important challenges facing the commercialization of IoT-related innovations is preserving system security and privacy of users' information as well as achieving high acceptance levels. Unfortunately, IoT inherits security threats from its enabling technologies and adds many constraints on any applicable security solution because of the special characteristics of IoT systems which make preserving the system's security more challenging. This increases the landscape of threats and makes the system vulnerable to inside as well as outside attacks.

However, IoT networks are usually implemented on a vast scale which makes them produce a huge amount of data during communication. This fact makes machine learning a promising solution for securing IoT systems. This huge data can be analyzed and used to detect abnormal behavior or anomalies. Nevertheless, according to resource and power constraints that IoT devices operate in, it is vital to reduce the needed storage and processing power needed for the detection algorithm or to propose an architecture that distributes the load over network nodes. Instead of implementing the Intrusion Detection System in a centralized way and handling data from the whole IoT system - which makes the system exposed to attacks and create single point failure or puts it at risk if the central server is compromised - distributed collaborative architecture could be used to take advantage of the massive deployment of IoT devices. The collaborative intrusion detection systems have better knowledge of their protected environments and provide a solution for the applications that are sensitive to user privacy.

In this work, we are going to introduce a new security solution for intrusion detection in IoT systems. Our proposed solution utilizes distributed collaborative architecture trying to take advantage of IoT structure and overcome its limitations. A federated learning method is proposed in this thesis. Using the private dataset, the local model gets trained by each node. Then, the parameters of its local model are shared with other nodes for the sake of generating a better global model.

This thesis proposes utilizing a Generative Adversarial Network (GAN) for the purpose of detecting anomalies. The model will be trained on the normal system behavior and let the generator mimic attacks while the discriminator detects anomalies based on their difference from the normal behavior. This technique could offer a solution for the problem of limited data points that represents malicious behavior.

Additionally, this paper suggests employing an autoencoder for feature extraction. There are four main purposes for doing so. The first is to improve the efficiency of the GAN training process by lowering system congestion. The second is minimizing the sample size required. Similarly, the third purpose is to make the training and classification process lighter and easier. Finally, it can also conceal the data for scenarios where the device shares its data along with its model's parameters to gain trustworthiness.

On the other hand, our solution will employ data sharing and mutual trust between system devices using blockchain technology. The collaborated devices share their model's parameters over the blockchain. In this way, they can compute the general global model by averaging all shared models or they can check their results using their neighbors' models. Furthermore, in distributed peer-to-peer IDS network alert exchange between the different IDS nodes is vital to detect anomalies and determine the trustworthiness of the nodes of the network.

Additionally, system devices might share an encoded version of their data over the blockchain along with their models' parameters to enable other devices to verify the detected intrusion. To determine the trustworthiness of a node, a calculation can be initiated based on the fulfillment of received alert-related information. Then, the blockchain registry would include the alerts generated by each IDS node. Consequently, the collaborating nodes would depend on the consensus protocol to judge the validity of the alerts before inserting them on the blockchain.

However, since each IoT system might have a different structure and characteristics according to its functionality and the circumstances it is implemented in, different IoT systems might apply our suggested solution with different settings. Also, according to the limitations that faced our research in terms of time and research equipment we are going to present a general structure for the proposed system and discuss it from security aspects that govern collaborative distributed IDSs.

DAĞITILMIŞ ANOMALİ TABANLI SALDIRI TESPİT SİSTEMİ BLOCKCHAIN TEKNOLOJİSİ KULLANILAN IOT ORTAMI İÇİN

ÖZET

IoT, endüstriden sağlığa, eğlenceye kadar birçok alanda birçok soruna çözüm sunuyor. Bu çözümlerin etkinliği, bu teknolojiye olan talebin her geçen gün artmasına neden oldu ve kullanımlarını her yerde görmekteyiz. IoT dünyası hızla büyüyor. IoT teknolojisini ticarileştirilmenin en önemli zorluklarından biri, sistem güvenliğini ve kullanıcı bilgilerinin gizliliğini korumanın yanı sıra yüksek memnuniyet seviyelerine ulaşmaktır. Ne yazık ki, IoT sistemleri genellikle büyük ve yaygın olarak dağıtıldığında, içerisinde birçok güvenlik zayıflığı ortaya çıkar. Buna ek olarak, IoT güvenlik tehditlerini etkinleştiren teknolojilerinden devralır ve sistemin güvenliğini korumayı daha zor hale getiren IoT sistemlerinin özel özellikleri nedeniyle uygulanabilen herhangi bir güvenlik çözümüne birçok kısıtlama ekler. Bu, tehdit ortamını artırır ve sistemi hem içeriden hem de dışarıdan saldırılara karşı savunmasız hale getirir. Bu tür sistemlerde bu güvenlik tehditlerinin varlığı hala birçok endişe uyandırmaktadır. Ayrıca bu teknolojinin yüksek düzeyde bilgi güvenliği gerektiren birçok alanda kullanıma sunulmasının önünde bir engel olarak durmaktadır.

Ancak, IoT ağları genellikle iletişim sırasında büyük miktarda veri üretmelerini sağlayan geniş bir ölçekte uygulanır. Büyük miktarda veriye sahip olmak, makine öğrenimi için verimli bir ortamdır. Bu gerçek, makine öğrenimini IoT sistemlerini güvence altına almak için umut verici bir çözüm haline getiriyor. Bu devasa veriler analiz edilebilir ve anormal davranışları veya anormallikleri tespit etmek için kullanılabilir. Çeşitli makine öğrenimi algoritmaları, anormallikleri tespit etmeyi başarmış ve birçok uygulamada etkinliklerini göstermiştir. Bununla birlikte, IoT cihazlarının içinde çalıştığı kaynak ve güç kısıtlamalarına göre, algılama algoritması için ihtiyaç duyulan depolama ve işlem gücünü azaltmak veya yükü ağ düğümleri üzerinden dağıtan bir mimari önermek hayati önem taşımaktadır. Makine öğrenimi algoritmasını uygulama yöntemini Nesnelerin İnterneti sistemine uyacak şekilde uyarlamak, güvenlik sistemini kurmanın ilk adımıdır. Genel olarak birçok Saldırı Tespit Sistemler uygulaması merkezileştirilirken, Saldırı Tespit Sistemini merkezi bir şekilde uygulamak ve sistemi saldırılara açık hale getiren ve tek nokta hatası oluşturan veya merkezi sunucunun güvenliği ihlal edildiğinde riske sokan tüm IoT sisteminden gelen verileri işlemek yerine, dağıtılmış işbirlikçi mimari kullanılabilir IoT cihazlarının devasa dağıtımından yararlanmak için. Dağıtılmış bir mimari, saldırı tespit sistemlerine birçok avantaj sağlar. Bu avantajlar, cihazlarındaki sınırlamalar nedeniyle IoT ortamında açıkça görülmektedir İşbirliğine dayalı saldırı tespit sistemleri, yerel verileri işledikleri için korunan ortamları hakkında daha iyi bilgiye sahiptir. Bu şekilde, her nokta çevresinin iyi huylu davranışı hakkında daha iyi bir anlayış oluşturur. Bu tür bir yapı, verilerin üçüncü bir tarafla paylaşılmasına gerek kalmadan yerel olarak sınıflandırılabileceği, kullanıcı gizliliğine duyarlı uygulamalar için de bir çözüm sunar. Dağıtılmış mimari aynı zamanda yükün ağdaki birkaç nokta üzerinden dağıtılmasını sağlar ve bu, Nesnelerin İnterneti cihazlarının sınırlı yetenekleri sorununun atlanmasına yardımcı olur.

Bu araştırma çalışmasında, IoT sistemlerinde saldırı tespiti için yeni bir güvenlik çözümü sunacağız. Önerilen çözümümüz, Nesnelerin İnterneti yapısından yararlanmaya ve sınırlamalarının üstesinden gelmeye çalışan dağıtılmış bir işbirlikçi mimari kullanır. Önerilen bu çözüm, ağ düğümlerinde dağıtılmış makine öğrenimi kullanacaktır. Bu makine öğrenimi biçimine işbirlikli federe öğrenme denir. Önerilen

sistemimizde her bir düğüm, kendi veri setini kullanarak yerel makine öğrenimi tabanlı bir anomali algılama modelini eğitir ve daha iyi bir genel model oluşturmak için yerel model parametrelerini diğer düğümlerle paylaşır. İşbirlikçi federe öğrenme, makine öğrenimi algoritma yükünün ağ noktalarına daha iyi dağıtılmasını sağlar ve yerel verilerin merkezi sunucu ile paylaşılması artık gerekli olmadığından kullanıcı gizliliğinin korunmasını destekler.

Bu araştırma çalışmasında anormallikleri tespit etmek için Generative Adversarial Network (GAN) kullanacağız. Model, normal sistem davranışı konusunda eğitilecek ve ayrımcı, normal davranıştan farklılıklarına dayalı olarak sapmaları tespit ederken, jeneratörü saldırıları simüle etmesi için bırakacağız. Bu teknoloji, oluşturucunun yapay örnekler oluşturduğu kötü niyetli davranışları temsil eden sınırlı veri noktaları sorununa bir çözüm sağlayabilir. Son olarak, ayrımcı bir anomali tespit modeli olarak kullanılır. Daha kesin olarak, bu araştırma, çok ayrımcılı Generative Adversarial Network yapısının kullanımını önermektedir, çünkü Generative Adversarial Network bu yapısı birleşik öğrenmeyi destekler ve her cihazın yerel olarak kendi ayrımcı edicisini geliştirmesine katkıda bulunur.

Ek olarak, bu araştırma çalışmasında özellikleri çıkarmak için Autoencoder kullanacağız. Özellik çıkarmanın faydası, verilerin boyutunu küçültmektir ve bu, sistemdeki tıkanıklığı önlemeye ve GAN eğitim sürecinin verimliliğini artırmaya yardımcı olacaktır. Örnek boyutunu küçültmek, eğitim ve sınıflandırma sürecini daha hafif ve kolay hale getirmeye yardımcı olacaktır. Autoencoder'ın sağladığı bir diğer avantaj, veri gizleme. Bu şekilde, her cihaz kendi model parametreleriyle birlikte verilerini diğer cihazlarla şifrelendikten sonra paylaşabildiğinden, her cihaz diğer ağ cihazlarından güvenilirlik kazanabilir.

Öte yandan, çözümümüz, merkezi sunucuyu değiştirmek için Blockchain teknolojisini kullanacak ve sistem cihazları arasında veri paylaşımı ve karşılıklı güven zorluklarını ortadan kaldıracaktır. İşbirliği yapılan cihazlar, modellerinin parametrelerini blok zinciri üzerinden paylaşır. Bu şekilde, tüm paylaşılan modellerin ortalamasını alarak genel global modeli hesaplayabilir veya komşularının modellerini kullanarak sonuçlarını kontrol edebilirler. Bu sayede blok zinciri, ağdaki cihazlar arasında aracı görevi görür ve merkezi sunucunun yerini alır. Ayrıca, dağıtılmış eşler arası saldırı tespit sistemi ağda uyarı alışverişi, bir anormallik olup olmadığına karar vermek ve ağ içindeki bir düğümün güvenilirliğini hesaplamak için kullanılabilen çeşitli saldırı tespit sistemi düğümleri arasında son derece önemlidir. Ağdaki her cihazın güvenilirliği, cihaz tarafından verilen alarmların geçerliliği ve güvenilirliği esas alınarak hesaplanmalıdır. Bu güvenilirlik, ağdaki diğer cihazlar tarafından model parametrelerinin yanı sıra alarmla ilgili veriler veya bu verilerin şifreli bir versiyonu aracılığıyla hesaplanabilir. Sistem cihazları, diğer cihazların tespit edilen izinsiz girişi doğrulamasını sağlamak için kendi modellerinin parametreleriyle birlikte blok zinciri üzerinden verilerinin kodlanmış bir sürümünü paylaşabilir. Bir düğümün güvenilirliği, alınan uyarıyla ilgili bilgilerin yerine getirilmesine dayalı olarak hesaplanabilir. Her bir IDS düğümü tarafından oluşturulan ham uyarıların blok zincirindeki işlemler olarak değerlendirilmesi önerilir. Blok zinciri sistemlerinde, genellikle belirli bir işlemin geçerliliği hakkında birden fazla düğüm topladığınızda, blok zincirine eklenir. Aynı şekilde önerdiğimiz sistemde, alarm işbirliği yapan düğümler arasında çoğaltılabilir. Ardından, tüm işbirliği yapan düğümler, uyarıları blok zincirine yerleştirmeden önce geçerliliğini garanti etmek için bir fikir birliği protokolü benimser.

Önerilen sistemin çalışma şeklini kısaca özetlemek gerekirse, önerdiğimiz sistem eğitim aşaması ve saldırı tespit aşaması olmak üzere iki aşamada çalışmaktadır. Eğitim aşamasında geçici olarak merkezi bir sunucu kullanılacaktır. Merkezi sunucu, otomatik kodlayıcıyı Federal Öğrenme tarzında eğitir. Ardından merkezi sunucu, otomatik kodlayıcının genel formunu tüm ağ cihazlarına dağıtır ve ardından her cihaz kendi veri setini şifreler. Bir sonraki adımda, merkezi sunucu aynı zamanda çoklu ayrımcılığa dayalı üretici hasım ağının federe öğrenme sürecinde bir arabulucu görevi görür. Merkezi jeneratörün sentetik verinin bir örneğini oluşturduğu ve bu verileri ağdaki cihazlara dağıttığı, her cihazın normal davranışı temsil eden yerel veri kümesine ek olarak anomaliyi temsil eden bu sentetik örneği kullanarak ayrımcısını eğittiği ve merkezi sunucudaki jeneratör ile geri bildirim. Bu işlemin birkaç tekrarından sonra, her cihazdaki ayırıcı, saldırıları temsil eden verileri normal davranıştan ayırt edebilir. Saldırıları tespit etme aşamasında, her cihaz kendi yerel verilerini izler ve kendi ayırıcısını kullanarak anormal davranışları tespit eder. Bir saldırı tespit edilirse, cihaz uyarıyı, ayırıcı model parametreleriyle birlikte ilgili şifreli bilgileriyle paylaşır. Bu bilgiyi kullanarak, ağdaki diğer cihazlar alarmın gerçekliğini kontrol eder ve mevcut uyarılarla ilgili bir fikir birliğine varmak için fikir birliği protokolünü uygular. Uyarı ile ilgili fikir birliğine varıldıktan sonra uyarı, ilgili bilgilerle birlikte blok zincirine eklenir.

Ancak her IoT sistemi, işlevselliğine ve uygulandığı koşullara göre farklı bir yapıya ve özelliklere sahip olabileceğinden, farklı IoT sistemleri, önerilen çözümümüzü farklı ayarlarla uygulayabilir. Ayrıca, zaman ve araştırma ekipmanı açısından araştırmamızın karşılaştığı kısıtlamalara göre, önerilen sistem için genel bir yapı sunacağız ve bunu orta.

1. INTRODUCTION

IoT can be defined as an interconnected network of physical objects with embedded systems which is connected to the internet by wired or wireless communication technologies [1]. Those devices gather and process relevant data to provide useful information to end-user applications. The applications of IoT range from smart house devices to more sophisticated ones like Radio Frequency Identification (RFID) devices, health monitoring systems, smart grids, and self-driving automobiles [2]. Inasmuch, the IoT world is growing rapidly. It is estimated that the total number of active IoT devices would hit 30.9 billion units by 2025, while the consumer's spending on smart homes worldwide reached 115 billion USD by 2020 [3]. One of the most important challenges to commercializing IoT technology is preserving system security and maintaining the confidentiality of users' information while achieving high satisfaction levels [4].

In fact, IoT inherits security threats from Cloud Computing, Software-Defined Networking, and Fog Computing technologies [5]. Further, it adds many constraints to the suggested security solution because of the special characteristics of IoT systems which increases the attenuating circumstances for the attackers [6].

IoT systems have some characteristics which make preserving the system's security more challenging. First of all, heterogeneity is where a variety of devices with different features, communicate using different protocols with each other, with different standards for communication, and varying constraints on their resources. On the other hand, the vast scale that IoT is implemented in also brings challenges related to identifying and protecting against malicious attacks. Furthermore, the connectedness of IoT devices to the global network makes them accessible anywhere and anytime. However, sometimes IoT is required to serve real-time applications like remote surgery and monitoring industrial processes. Still, performance can be constrained by issues such as delay and dependability. Moreover, to serve the efficiency purpose, IoT devices will operate dynamically, the devices go into sleep mode and wake up when it is needed. In the seeking to efficiency, also the communication might be held directly when there is no need to connect through the internet. Briefly, this makes IoT network a very dynamic environment.

Low power consumption is essential to achieve the required efficient and widespread IoT devices. It is clear that the resource constraints complicate the implementation of available regular security solutions in IoT networks [7].

IoT networks produce a huge amount of data during communication. It can be analyzed and used in detecting abnormal behavior. In fact, anomaly detection techniques that utilize Machine Learning algorithms are already used in security field to detect intrusions. Thus, there was a considerable research effort to develop ML-based solutions for IoT as well [8].

However, the problem of getting labeled data for training ML models especially the shortage in the data points that represent malicious behavior still one of the main obstacles facing developing stable accurate anomaly-based Intrusion Detection Systems (IDS) [9].

According to resource and power constraints that IoT devices operate in, it is vital to reduce the storage and processing power for detection algorithm. This might make the direct application of some conventional ML techniques unsuitable [7].

Traditionally, the IDS is implemented on computationally powerful data centers or nodes that handle data from the whole IoT system or a selection of nearby nodes. Still, centralized systems suffer the problem of exposing the system to attack and create single point failure or makes the remainder of the IoT susceptible to threats if the centralized IDS was compromised [6]. In order to solve the above issues within the current limitations distributed collaborative system could be employed. This kind of collaborative architecture enables taking advantage of the massive deployment of IoT devices.

Distributed IDS can serve two objectives in IoT systems. First, most IDSs need a large number of observations about the network's normal status and failure circumstances. However, in IoT, most centralized IDSs have little knowledge of their protected environments. The collaborative intrusion detection architecture, on the other hand, allows multiple IDS nodes to comprehend the context by exchanging data and knowledge where an IoT device may have a set of data that comprises a small fraction of the network's condition. Second, in services that are sensitive to user privacy, like as health or financial-related services, it might not be possible to share end users' data with the system's administrator. In turn, this would render data-centric IDS solutions ineffectual [10].

Despite extensive research regarding intrusion detection, trust computing and data sharing in a collaborative setting remain key problems. Data sharing is a main concern for a collaborative detection system, because gaining the trust of all involved parties is not easy. Furthermore, some parties are unwilling to contribute data due to privacy concerns, yet without adequate observations, it is impossible to refine the algorithms of detection and develop a viable solution for detecting suspicious occurrences. On the other hand, to collect the behavioral data and traffic from the nodes, a central server is deployed. It also serves to calculate every node's trust value. Nevertheless, the larger the company, the harder to conduct proper trust management. This is because of the difficulties in finding a trustworthy third party as it is possible to compromise a central server [11][12].

It may be possible to address the problems mentioned above in intrusion detection using Blockchain technology. There are two requirements that limit the ability to share data, namely data privacy, and mutual trust. Data privacy means that the shared data could contain information about to the company. This includes, shared traffic, including IP addresses and packet payloads which could cause serious threats to the company's privacy. On the other hand, mutual trust means that when sharing data, the parties involved must be able to trust one another with regards to not leaking the data.

Blockchains can address those challenges. In this regard, the transactions recorded on the block can be the shared data. The involved parties are also required signing digitally the data sharing agreement which would be kept in the blockchain. As such, the blockchain can be accessible by third parties who can read the agreement and affirm the ownership of the data. Further, to secure data privacy, transformed data can be shared instead of raw data [13].

However, in distributed peer-to-peer IDS network alert exchange is of a great importance among the different IDS nodes. This can be utilized to aid in deciding whether an anomaly exists or not and in computing the trustworthiness of nodes in the network. The node's trustworthiness can be determined based on the fulfillment of received alert-related information. However, it is still a challenge to perform trust computation robustly. Blockchain technology can be a potential solution to tackle this issue by considering the alerts produced by each IDS node as transactions on the blockchain. The collaborating nodes can easily replicate that. As a result, all collaborating nodes would utilize a consensus protocol that serves a guarantor of transactions' validity of the transactions prior to putting them in a block. This, in turn, would prevent tampering with the alerts stored in the blockchain and that they can be relied on to determine the trust value of each node in the system [14].

In this research, we are going to introduce a new security solution for intrusion detection in IoT systems. This solution will utilize Machine Learning for anomaly detection. Generative Adversarial Network (GAN) is used which is a sort of neural network. GAN could give more accurate results since in this technique one neural network learns to produce new data that have similar statistical characteristics as the training set [15]. This technique could offer a solution to the problem of limited data point that represents malicious behavior. Moreover, the suggested solution is going to be distributed to achieve better efficiency, accuracy and preserve data privacy. On the other hand, the proposed solution will employ Blockchain technology to address the issues of data sharing and trust management.

2. LITERATURE REVIEW

Since the invention of the internet, numerous security issues have been raised because of the unpredictable uses of networks. Many statistical evidence show that the number of attacks is increasing year by year. To manage this problem, serious effort to create an efficient intrusion detection system (IDS) was done since the 1980s. As a result, several literature surveys IDS since that time [15]-[32].

Throughout the development of IDS, intrusion detection methodologies have been developed into two categories: Signature-based detection, Anomaly-based detection. Signature-based detection methodology is similar to anti-virus software, whereby it defines patterns or strings which are linked to known attacks or threats. They are then compared against captured events in an attempt to detect possible intrusions. This methodology is also called the Knowledge-based Detection [33][34]. Signature-based detection normally gives an admirable detection accuracy for known intrusions while suffering difficulty in detecting zero-day attacks when there is no matching signature available in the database. Signature-based detection is deployed in many popular tools, like Snort and NetSTAT.

In the other category, anomaly-based detection, a normal machine learning-based model of the behavior of a computer system is created, and any deviation from the known normal behavior of hosts, users, or network connections over a period of time might be detected. Anomaly-based detection has taken huge interest from many researchers because of its ability to address the limitation of signature-based detection. This made the most of efforts to develop IDS focused on developing suitable anomaly detection algorithm[31][35].

Remarkably, trying to develop an accurate IDS using machine learning models was a hot topic during the last two decades. Used ML methods diverged through almost all known ML methods, like Artificial Neural Networks (ANN) [36]-[38], Association Rules (AR) [39][40], Fuzzy Logic (FL) [41], Bayesian Network (BN) [42][43], Clustering [44], Decision Trees (DT) [45][46], Ensemble Learning (EL) [47][48], Markov Chains (MC) [49][50], Naïve Bayes (NB) [51][52], Support Vector Machine (SVM) [53]-[56], Genetic Algorithm (GA) [57]-[59] , and Neural Networks.

However, only few research in the literature compare the performance of different ML algorithms. Not to mention making the compression in the context of intrusion detection. Overall, according to the empirical comparison done by [60], in order to appropriately conduct a performance comparison, there should be a comparison before and after calibration of the estimations with a suitable approach. This is because of the smoothing effect that calibrating can cause on the output which, in turn, can make the model fit more suitably to the distribution of the input. Generally, ANNs and Random Forests give the finest results before calibration while after calibration SVMs and boosted trees do better. Moreover, generalizations do not hold because there are important differences between the distinct problems where various metrics are used, and this makes the fit of the model not always consistent. Generally, while certain algorithms are considered to perform better than others, the performance of an ML technique is highly reliant on the application and implementation [35].

Nevertheless, some peculiarities make the implementation of ML in IDS arise some difficulties as L. Buczak has mentioned in [35]. In IDS models are trained repeatedly, every day, or whenever the analyst requires since new intrusion is identified. This makes the training time of the model very important and makes starting from the trained model and continuing training it in incremental learning more efficient and reasonable. The other problem is to obtain suitable training data, while ample data can be collected through installing sensors on the networks. However, there is a need to label some of the observations for them to be useful. This can cause a laborious task. For this reason, most research tended to use preprepared public data sets like DARPA 1998, DARPA 1999, DARPA 2000, KDD 1999, or NSL-KDD. But the main problem with these data sets is that they lack freshness.

Fundamentally, IoT is a sort of computing system which uses the internet to communicate and transmit data from sensors to users or embedded systems and vice versa. Consequently, this makes IoT vulnerable to numerous threats as any other network-connected computing system. Moreover, some characteristics of IoT make it more probably attackable than a traditional system like the number of endpoints. However, using traditional IDS systems in IoT is hard because of its specific features like specific protocol stacks, constrained-resource devices, and standards as mentioned previously [61].

Throughout the last decade, developing security solution for IoT was a big challenge, which made this topic attracts a huge research effort. The work done in this field varies by the aspects of intrusion detection method and placement strategy. The detection method can be anomaly-based or signature-based. The placement strategy depends on the structure of the system and can be centralized or distributed.

In the signature-based detection method, IDS detects attacks that match a previous attack signature stored in the database of this IDS. The signature-based method is very effective if the intended threat is a known attack. But this method tends to fail when it is intended to detect new attacks. For this reason, the need for another detection method arises over the last years since new threats appear every day [31].

There is many published research that used signatures as a detection method, for example, Liu et al. [62] proposed an IDS that uses detectors that store attack signatures and work as immune cells by classifying each datagram as malicious or benign. In their proposed system, detectors can develop to adapt to changes in their environment as Artificial Immune System. However, they do not consider how this approach can be adapted to low-capacity nodes as in IoT. In IOT environment the computational and storing cost might be a problem. Another work done by Kasinathan et al. [63] [64] integrated a signature-based IDS to detect DoS attacks in 6LoWPAN networks. In this system, the IDS sends the warnings to a DoS protection manager which evaluates some other variables like packet dropping and channel interference rates to assert the existence of an intrusion. Yet, the way the data updating of signatures databases is unclear. Additionally, in a latest work, Ioulianou et al. [65] proposed a Signature-based IDS for the IoT. In their proposed design the suggested system includes both centralized and distributed IDS modules. The authors used the Cooja simulator to implement a Denial-of-Service attack scenario on Internet of Things devices exploiting the widely used RPL protocol. Attack mitigation is done through gauging the Received Signal Strength (RSS), packet sending rate, or packet data drop rate. It is also achieved through observing the number of node IDs in the network.

As it was mentioned before, the signature-based approach failed in the rapidly evolving hostile environment. This boosts the researchers to develop an approach that can detect zero-day attacks. For this reason, a heavy effort was done in the field of anomaly detection. Many proposed systems have been developed based on anomaly detection or integrated anomaly detection with other techniques. Anomaly-based IDS compares the activities or the behavior of a system against the normal behavior and triggers an alert when the detected action deviates from normal behavior in a way that exceeds the predefined threshold. However, since anything that deviates from normal behavior is considered as an attack, the main problem anomaly detection is suffering is the high false-positive rate which increases if the training set does not well represent the normal and abnormal scope. [66]

Used learning methods for IoT security can be categorized into supervised, semi-supervised, and unsupervised approaches. However, the used machine learning algorithms can be considered heavy for low-capacity nodes of IoT networks. Thus, the efficiency of the proposed anomaly detection-based systems for IoT networks should take that into account during the design stage.

An example of anomaly-based IDS for IoT is what Cho et al. [67] proposed in their work. They suggested a detection scheme for botnets. The authors build their work based on an assumption that botnets create unanticipated alterations in the traffic of 6LoWPAN sensor nodes. So, they suggested a system that calculates the average of the sum of TCP control field, the number of active connections of each node, and length of the packet, to compose the normal behavior profile. The system triggers an signal when the specifications for any node diverge from the calculated averages. Another system was proposed by Lee et al. [68]. In their work, the researchers defined multiple models for energy consumption to both route-over routing and mesh-under routing. If the energy consumption of a node differs from the anticipated value, the IDS detects this node as a malicious node. This system was specially created for low-capacity networks, but the authors did not put forward the results about false-positive rates. Also, Summerville et al. [69] developed an anomaly-based IDS based on the fact that small IoT devices mostly use limited and fairly simple protocols. Those kinds of protocols produce highly similar payloads. Network payloads are handled as bytes, and the feature selection is performed by bit-pattern matching by overlapping tuples of bytes. Pongle and Chavan [70] presented a specialized IDS for detecting wormhole attacks in IoT. The authors depend on signs of wormhole attack like a high number of neighbors get created after the attack and the large number of control packets used during communication of the two ends of the tunnel. By knowing those facts authors used three algorithms for anomaly detection. According to the authors finding, both the memory consumption and power requirements of the proposed system are low making it suitable for IoT systems.

Noticeably, almost all ML techniques have been used to develop detection models, but some ML promising techniques earns much interest recently according to their superior results like Random Forest and Neural Networks. For instance, Alrashdi et al [71] proposed IDS system for smart cities. They proposed an anomaly detection system utilizing Random Forest machine learning algorithm where the detection modules are implemented in fog nodes. In evaluation UNSW-NB15 dataset was used and they claimed that their proposed system can achieve highest classification accuracy effectively. They mentioned that the accuracy reached 99.34% with lowest false positive rate. Also, Ferdowsi et al. [10] proposed a distributed intrusion detection system for IoT networks using GAN as classification algorithm which is a special sort of Neural Networks. The authors employed distributed GAN instead of the traditional GAN to avoid the communication cost caused by the centralized architecture of GAN. The authors reported that their approach could achieve higher detection accuracy with 20% rate, and a higher precision with 25% rate, and the most recognizable is getting 60% lower false positive. Those statistics were defined compared to the centralized GAN. In different work Intrator et al. [72] have presented another anomaly detection approach based on multi discriminator GAN. The system comes with one generator, two discriminators forming a complex network to determine if the artificial generated samples are of sufficiently acceptable and an autoencoder that was used in their approach as the anomaly detector. The proposed architecture empowers the GAN to generate complicated samples that can mimic the instances of training dataset. Also, the generated data instances still can be accurately reconstructed by an autoencoder. According to the authors, the suggested structure of the system improves the performance as they evaluated it using different datasets.

On the other hand, the way the IDS is placed in the IoT system has a great effect on the efficiency and effectiveness of the IDS. For this reason, IDS placement approach was also field of intensive study. As we mentioned before, there are two different approaches according to IDS placement: centralized and distributed. In the centralized IDS placement, it's possible to locate the IDS in the border router or the host. Here the IDS of the border router can evaluate the past traffic through them from the end nodes to the Internet and vice versa. [73][74] But the problem is that sometimes the malicious traffic does not pass-through border routers. Instead, it passed between nodes in the same region. Cho et al. [67] proposed a solution to detect botnet attacks. In their solution, the IDS analyzes the traffic's packets which pass through the border router of the local network where the centralized approach can effective. Kasinathan et al. [63][64] also employed the centralized approach by deploy the IDS at dedicated host. The system is consisting of IDS sensors at the end nodes perform sniffing on the network and monitor the traffic and reporting the results to the centralized intrusion detection system located in the host. Wallgren et al. [75] presented in their work a suggestion about centralized IDS based on a heartbeat protocol. In their suggestion, the IDS is located in the border router. As a replacement for checking the traffic passing through the border router, all the nodes receive ICMPv6 echo requests from the border router at frequent intervals. The the return responses are assumed to detect attacks if any and detect the availability of the node. The energy overhead was minimal even when this way creates more traffic in the network as the author mentioned.

However, because of some IoT systems' characteristics including the large scale and the need to protect users' privacy, traditional centralized IDSs may do not give the best result. Depending on centralized nodes to operate IDS may make these nodes vulnerable to attacks and as a result, the remainder of the IoT system can be attacked if the centralized IDS is exposed. Additionally, the majority of IDSs require large amounts of observations about the normal state of the network as well as attacks or failures. But in many applications, the final customer may be unwilling to share the available dataset with the centralized host of the IoT which can make the centralized IDS solutions fruitless.

As a result, significant work has been done to develop a distributed IDS, whereby IDSs are installed in every object in the IoT system. Here IDS is deployed according to resource-constrained. Oh et al. [76] and Lee et al. [68] proposed distributed IDSs that are lightweight to serve the purpose of distribution. Oh et al. created a lightweight algorithm in order to monitor packet payloads and match them with attack signatures using two techniques, early decision, and auxiliary shifting. The main idea was to decrease the number of matches required for detecting attacks. The authors' proposed method gives fast results even when applied in a resource-constrained platform. Likewise, Lee et al. proposed a lightweight solution that depends on monitoring the energy consumption in each node to detect intrusions. Concentrating on one single parameter minimizes the computational resources required for intrusion detection.

Other proposed solutions make the nodes also responsible for monitoring their neighborhood as guards. Cervantes et al. [77] suggested a combination of watchdogs' concept with node reputation to help detect attacks. In the suggested system, nodes are created in a hierarchical structure. Some nodes are designated as leaders, associated, or member nodes. However, the role of each node can shift over time according to attack events. When a node spots an attack by monitoring inbound and outbound traffic, it broadcasts an alert in the system trying to isolate the attacker. Le et al. [78] suggested a solution that organizes the network in regions. There are monitor nodes that cover the entire network, those nodes work by sniffing the traffic from its neighbors to detect an attack if a node is compromised. Le et al. [79] again suggest in other latest work to organize the network into small regions with a node in each as region head. An IDS case is located in each region head which monitors the other region's nodes by sniffing all ongoing communication. In this design, all nodes should report information about their behavior and also their neighbors to the region head node. The head nodes are considered to be more powerful. Still, the authors decided to design a lightweight IDS solution.

Still, many distributed solutions feel the need for centralized authority for the sake of organizing of monitoring and detecting processes. For instance, Pongle and Chavan [70] suggested making network nodes responsible for discovering any abnormal behavior in their neighborhood and reporting it to centralized IDS in the border router. This centralized IDS works to analyze all data that comes from other nodes in that region to detect any intrusion. Additionally, Thanigaivelan et al. [80] proposed an IDS module in which each network node observes its neighbors. if an abnormal occurrence is detected, the network node transmits a warning to the IDS located on the border router which, in turn, works as manager and coordinator by matching up warnings from different nodes and make a conclusive judgment on the attack.

Alternatively, some studies introduce blockchain technology as a promising possible substitute for the centralized authority in cooperative distributed IDS. Alexopoulos et al. [81] discussed the concept of employing blockchain technology in improving collaborative IDS to improve trust between different entities and to give away in which decisions about intrusions and faulty nodes can be made. The authors proposed a generic architecture about how blockchains can be implemented in the field of Collaborative IDSs. Alkadi et al [82] proposed a Deep Blockchain Framework (DBF) for Internet of Things networks which provides distributed intrusion detection system that preserves privacy using a blockchain smart contract. They implemented a deep learning algorithm as an intrusion detection method which is Bidirectional Long Short-Term Memory. To offer the required privacy based on blockchain, they developed a smart contract using Ethereum. Deep Blockchain Framework is evaluated using the datasets of UNSW-NB15 and BoT-IoT. The authors evaluated the DBF framework and discussed a comparison with other models and claimed that the experimental outcomes show that DBF gives better accuracy with a lower false-positive rate. Additionally, Liang et al. [83] proposed a geographical dynamic Intrusion Detection that utilizes Micro-Blockchain. This system was proposed specifically for IoT in transportation applications. The authors developed a micro-blockchain in a nested structure wherein each small region there is a micro-blockchain can create local intrusion detection strategies for vehicles. Another interesting collaborative IDS was proposed by Hu et al. [84]. They introduced distributed intrusion detection in multimicrogrid systems that substitute the central trusted authority with blockchain. The suggested system implements a proposal generation method. Using correlation model and consensus mechanism final detection results can be brought out from the generated proposals. The final detection results are recorded on the blockchain.

3. THEORETICAL BACKGROUND

In this chapter, we are going to explain the background concepts on which our system is built theoretically. First, we will discuss the federated distributed learning. After that, the concept of deep learning and Neural Network (NN) will be introduced with further discussion about both the Generative Adversarial Networks (GAN) and the Autoencoders (AE). In this discussion, the structure of those model will be explained, how GAN can be used for anomaly detection, and how can AE serve for feature reduction and provide blinding and hiding of the original data. Finally, the blockchain technology is going to be discussed and how it can be used to serve federated distributed learning and collaborative IDS.

3.1 Federated Learning

Federated learning basically is training Machine Learning (ML) models over remote devices or different separated data centers along with store and maintain device's data locally. This approach is introduced to deal with challenges rising during training ML models in heterogeneous environment and in supposedly very big networks. Which led to switch from the traditional learning techniques to cross-the-board machine learning techniques that employs the distributed approach for training and optimization as well as preserves privacy of user's data. However, Due to the increasing computational power of the edge devices and raising concerns about information privacy the solution of keeping data on the local device and let the edges do the required computation became a very reasonable solution.

Federated learning methods have been used in practice in different fields from smartphones to large scale organizations, while one of the most active fields is IoT. IoT networks typically contain many sensors that collect data from the environment which can be used for ML training in real time. While using this data in centralized training approach may be difficult due to the privacy concerns and the limited connection bandwidth, federated learning can serve to train models in IoT systems efficiently.

In federated learning, learning process involves one single global ML model. Which aimed to be learnt using device-generated data which is processed locally with periodically updates transmitted to the central server. The objective is in general to minimize the aggregated loss function:

$$\min \boldsymbol{F}(\boldsymbol{w}), \boldsymbol{where}\ \boldsymbol{F}(\boldsymbol{w}) = \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{m}} \boldsymbol{p_k F_k}(\boldsymbol{w}) \qquad \mathbf{(3.1)}$$

With the total number of devices $\boldsymbol{m}$ involving in federated learning process, $\boldsymbol{F_k}(\boldsymbol{w})$ the loss function of the *k*th device, *w* is the weights or parameters of the model, and $\boldsymbol{p_k} \geq \mathbf{0}$ and $\sum_{\boldsymbol{k}} \boldsymbol{p_k} = \mathbf{1}$. $\boldsymbol{p_k}$ is a factor that is defined according to the impact of the *k*th device and its dataset on the whole model.

Model learning process in federated learning consist of four steps. In the first step the central server distributes an initial model across the different devices. Next, each device or node train the model and make the optimization according to its local dataset. After each node have its own model with new parameters, each node sends those parameters to the central server. Finally, the central server uses all received parameters to generate a moderate model and distributes it across all nodes again. This process is explained in Figure 3.1.

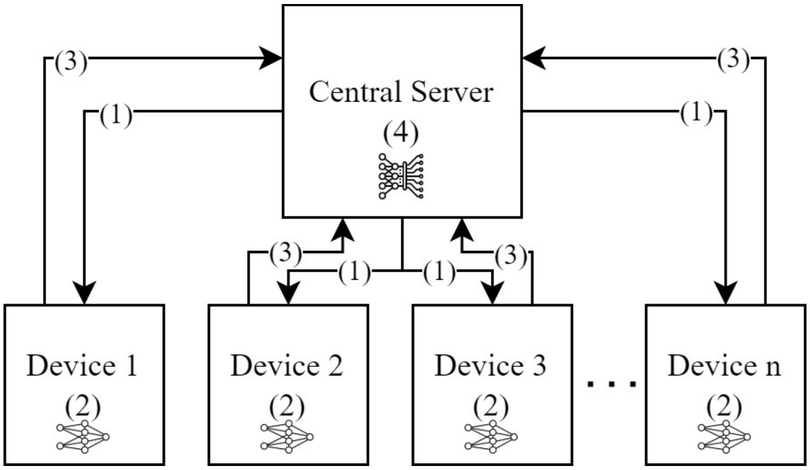

Figure 3.1: Model learning process in federated learning.

However, federated learning may face some challenges according to distributed optimization. Those challenges presented in expensive communication, heterogeneity, and privacy.

First, having smooth communication is a significant need for the federated networks. So, to keep the general model updated according to the generated data by all the devices in the network we need efficient communication method which periodically send model updates during training process instead of transmitting the entire data set over the network. Additionally, the heaviness of the communication might be further reduced by decreasing the communication rounds needed during training using local updating methods. Also, by reducing the messages' size that need to be transmitted on each round using compression schemes. Additionally, decentralized training can be used making nodes in local area communicates between each other instead of communicating with central training server.

The second challenge is the heterogeneity. Typically, in term of storage, computational power, and communication capability of the different devices in the network. Consequently, federated learning must deal with system's heterogeneity by expecting limited participation and tolerating heterogeneous hardware. Nevertheless, heterogeneity also occur in the generated data across the network. Thus, according to heterogeneity situation severeness solutions like training distinct local models simultaneously instead of training one single global model can be used.

Finally, while federated learning was proposed as a kind of a step toward protecting privacy of the data generated on each device, privacy still is a major concern during developing most federated learning-based systems. The reason is that the model updates which is transmitted throughout the training process may in somehow reveal sensitive information. However, unfortunately as the solution becomes more rigorous in term of privacy its performance and efficiency decreased. So, finding the balance still is an important challenge for federated learning. [85]

3.2 Neural Networks and Deep Learning

Artificial Neural Networks (ANN) consist of number of basic units or nodes called neurons. Those neurons are interconnected forming a dense and complex network called Neural Network. ANN models can learn by tuning parameters or weights which describe the connectivity of the network.

The simplest unit of a neural network called perceptron. Figure 3.2 shows how the single perceptron processes data. At each neuron, the input data x_i is weighted by a value w_i related to that input, then all weighted inputs are summed up to single value.

This single value then is summed up with a bias or counterbalance value denoted by *b*. Lastly, the output of the neuron is generated by substituting the previously resulted value in a function called the activation function denoted by *f(x)*.

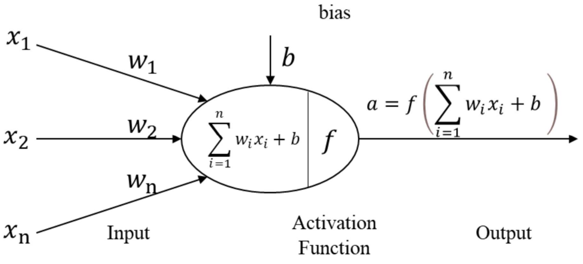
bias
b
x_1
w_1
x_2
w_2
w_n
x_n
$\sum_{i=1}^{n} w_i x_i + b$
f
$a = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$
Input
Activation Function
Output

Figure 3.2: Artificial neuron (perceptron) [87]

Neural network can be described as a mathematical model formed in a shape of multilayer perceptron. It is consisting of input layer, output layer, and hidden layer(s). For each hidden layer we need one or more neurons. If neural network consists of an only one single hidden layer, it is called a shallow neural network. While there is also the deep neural network which is a neural network consists of multiple stacked hidden layers one after another it is called.

In each neuron, activation function works as a threshold operator. Activation function can be linear function or non-linear function. There are a number of frequently used activation functions in neural networks like linear, sigmoid function, tangent function, rectified linear unit (ReLU), as well as softmax activation functions.

During training process of the neural network, iterative optimization algorithm is used to justify the parameters *w* and *b* in such a way that improves the results according to objective or loss function. Loss function is a function that represents the difference between estimated and real value of the target. There are number of loss function that can be used for the optimization process like cross-entropy, mean absolute error, and mean squared error.

However, instead of trying countless parameters to optimize model's result, optimization algorithm like gradient descent and Newton's method can be used. The main idea of using optimization algorithms is to try only limited number of iterations in the aim of minimization of the loss and figure out whether increasing or decreasing of the parameters will result in better values. This can be achieved by finding the derivative of the loss function with respect to certain parameters. Model's parameters can be adjusted step by step with a step size of value α. This value is called learning rate.

Generally, the process of training and optimizing of neural networks is called backpropagation. The whole process consists of four steps:

1. Forward propagation: In this step random parameters are chosen, and the output is calculated according to these parameters using the input. The error or loss is calculated using the ground truth.
2. Backward propagation: According to the total error or loss the parameters are modified seeking to get more accurate outputs.
3. Finding the total gradient using the gradients of each layer in the network.
4. Updating parameters of the model according to the total gradient and learning rate.

However, during training neural network model instead of training the model with the whole dataset once in single iteration, the dataset can be divided into number of batches. For each batch one forward pass and one backward pass is done which is called one iteration [88][89].

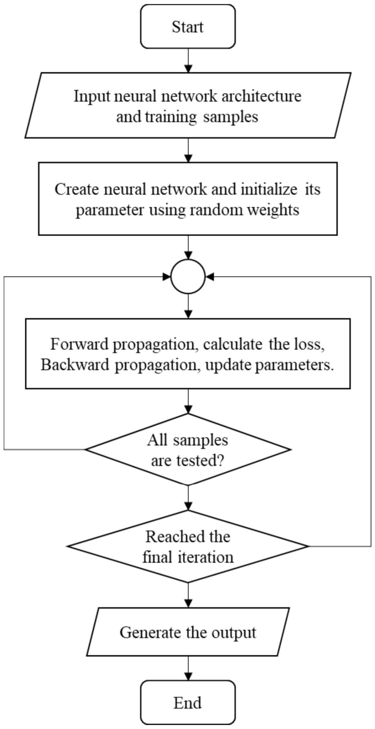
Start
Input neural network architecture and training samples
Create neural network and initialize its parameter using random weights
Forward propagation, calculate the loss, Backward propagation, update parameters.
All samples are tested?
Reached the final iteration
Generate the output
End

Figure 3.3: Backpropagation process for training neural networks.[89]

3.3 Generative Adversarial Network

Generative Adversarial Network (GAN) was proposed in 2014 by I. Goodfellow et al. [86]. GAN can be described as a pair of neural networks which is trained in competition with each other. One is the generator G which create fake samples trying to make them look like real as much as possible. The other one is the discriminator D, which receives both fake and real data samples and try to distinguish them apart. Both generator and discriminator are trained simultaneously compete in minmax game.

Theoretically, Generator and Discriminator can be implemented using any form of differentiable system that can map data from input space to output space. However, both are normally implemented using neural networks.

The generator learns through interaction with the discriminator. The discriminator takes the fake samples and the real samples and create an error signal or loss using ground truth of the origin of the samples. This error signal or loss is used to train the generator to be able to create better quality fake samples. Figure 3.2 explain Generative Adversarial Network training process.

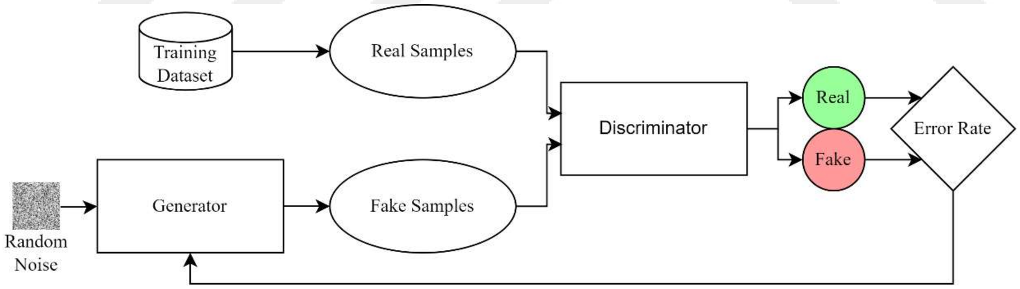

Figure 3.4: GAN Training Process. [91]

The training of GAN consists of minmax game which try to find the parameters that maximize the discriminator's classification accuracy as well as tuning the parameters of the generator to maximize its ability to perplexing the discriminator. The training involves solving the following value function

$$\mathbf{min\,max}\,\mathbb{E}_{p_{data}(x)}\,\mathbf{log}\,\mathcal{D}(x) + \mathbb{E}_{p_z(z)}\,\mathbf{log}(1 - \mathcal{D}(G(z))) \qquad \mathbf{(3.2)}$$

Practically, both generator and discriminator are multilayer perceptrons. To learn the distribution p_g of the generator over data x, a random noise z is used as input where $p_z(z)$ can be used to represent its distribution. Then a differentiable function $G(z, \theta_g)$

is defined represented by a multilayer perceptron where the weight parameters of the generator is θ_g. The same is for the discriminator. A second differentiable function $D(x.\theta_d)$ is defined with another multilayer perceptron which outputs a single scalar which corresponds to the probability that the input data sample belongs to real dataset rather than p_g.

By training D we trying to maximize the probability of its ability to distinguish between samples from training dataset and samples created by the generator. On the other hand, training of the generator is aiming to minimize $\log(1 - \mathcal{D}(G(z)))$, or in another word minimize discriminator ability to assign 'fake' label to generator's samples. Theoretically, during the training process, the discriminator's parameters are justified to get the optimal result according to the fixed parameters of the generator then the generator's parameters are justified. However, in fact the discriminator is trained for some limited Iterations. After that the generator is updated according to it.

Actually, the training process of GAN is very challenging since it can be very difficult to reach a convergence point for both generator and discriminator. Also, in some cases the generator may become confined and only generate samples which is similar to only a small part of the whole distribution of the dataset. Furthermore, in some cases the discriminator can be able to converge very fast letting no chance for the generator to be improved.

In fact, GAN was developed aiming to generate totally artificial images that seems to be real. However, GAN was also used to synthesize sounds, speech, tabular data, time series beside generating images. Many applications of GAN were discussed in [90]. However, GAN was used in many important fields other than generating artificial samples. One of those fields is anomaly detection.

There were multiple proposed architectures of GAN to achieve the goal of detecting abnormal samples. One architecture named AnoGAN [92] utilizes a standard GAN and by training the model only with positive samples. The aim is to find out a mapping from the latent space representation z to the generated sample x' which is artifcial sample. Also, new samples can be mapped to laten space using this learned representation. In this suggested architecture, since only normal samples are used to train GAN, the generator learn the distribution of normal samples. So, the generator learns how to generate normal samples from any given laten space. If an abnormal image is encoded to its laten space the rebuilding that the generator will generate a non-anomalous image and the different parts between the original and the generated images simply show the anomalies.

Another suggested architecture was EGBAD [93] EGBAD utilizes the BiGAN architecture in anomaly detection. Also [94] introduce the GANomaly architecture which was trying to improve AnoGAN and EGBAD. In this architecture, generator network concept is used. Generator network consists of three components, a generator encoder G_E, a generator decoder G_D, and another encoder E. The generator network is trained on normal samples only to find out their distribution and the autoencoder is trained to discover how to encode the samples to the latent space representation. There is also the discriminator network D, the other part of GAN architecture. Figure 3.5 shows those three architectures [95].

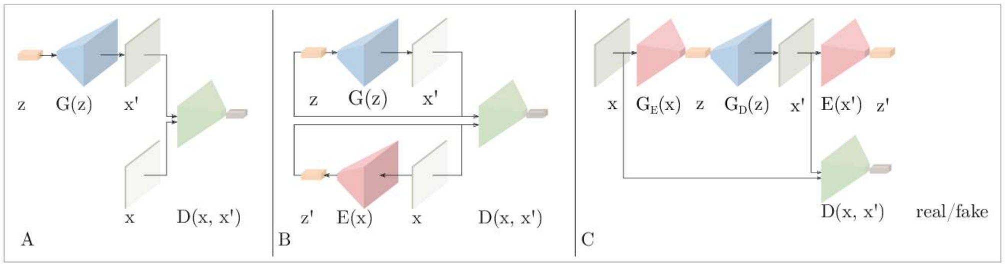

Figure 3.5: Architectures comparison. A) AnoGAN B) EGBAD C) GANomaly. [95]

Beside the previously described methods were utilized to detect anomalies in the fields of computer vision and time series, there is many other approaches were suggested to detect anomalies in term of intrusion detection. One is suggested by A. Ferdowsi and W. Saad in [10] to detect intrusions in distributed IoT system.

In their suggested system every device has a generator and discriminator and dataset which represent the normal status of the device. The generator attempts to generate data samples close to the normal state data to find the best estimate of distribution of the previously transmitted data points. On the other side, the discriminator tries to discriminate the data points which come from the generator from device's real dataset. Actually, the data points that generated by the generator represent anomalies in the system since the distribution which is generated by the generator is not equal to the distribution of the normal dataset. This is a result of the fact that the parameters of the generator are randomly chosen initially. Both generator and discriminator try to optimize their parameters in such that the generator can generate samples similar to the normal state and the discriminator is able to discriminate between the abnormal and normal samples.

3.4 Autoencoder

Autoencoder is a kind of neural network which is trained to reconstruct its input. The main goal is to produce lighter size representation of the data so it can be used as dimensionality reduction or feature learning approach. Autoencoder's structure consist of an encoder that compress the input and generate the code or latent representation and a decoder which tries to reproduce the input sample using that code. Both encoder and decoder are trained together using backpropagation aiming to minimize the loss or error of reconstruction. Figure 3.6 shows the structure of autoencoder [96].

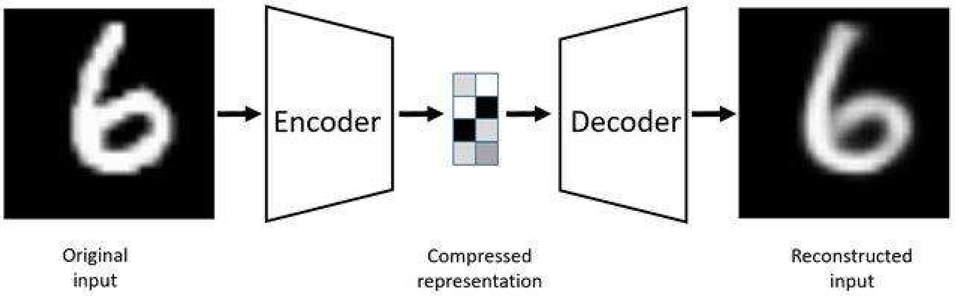

Figure 3.6: Autoencoder example. [96].

As other deep learning models, there are several hyperparameters must be set to train the model according to them. For autoencoder there are model's structure parameter which define the shape of the model. One is the size of the code or latent representation. For autoencoder that is used for features extraction the code size is less than the size of the input. However, there are also overcomplete autoencoder that its code size is larger than the input. This kind of autoencoder aim to capture better knowledge from the available data. The other structural parameter is the number of layers and size of each layer in both encoder and decoder. According to layers number the autoencoder can be shallow single layer vanilla autoencoder, or deep stacked autoencoder. Likewise, the number of neurons in each layer determines the complexity of the model.

Also, there are training parameters which the model is trained according to. One is the used activation function at each layer (linear, sigmoid, tangent, rectified linear unit

(ReLU), and softmax). Other is the used loss function (mean squared error, or cross entropy) as well as the used optimizer (gradient descent and its extensions) [88].

The tuning of those hyperparameters is important to achieve accurate results. However, this process is considered to be exhausting during the training a multilayered neural network. Some approaches have been suggested for hyperparameter tuning. Basically, in our work the manual hyperparameter search is used. According to this approach, different combinations of parameters can be tried and based on results the best combination is selected [97].

In our suggested system we are going to use autoencoder to extract features by projecting samples to a less-dimensional space. The projection of samples into a less-dimensional space serving in different aspects. First, it reduces the needed communication capacity by reducing samples size and this reduces congestion in the system. Also, since training GAN as a deep learning model is relatively heavy task making sample size less will make training smoother and faster. Last but not least, dealing with data as encoded codes will perceive privacy and provide blindness during transmitting data through the whole system for training the model and intrusion detection as well.

3.5 Blockchain Technology and Its Role in Federated Learning and Collaborative IDSs

Blockchain is simply a data structure that store information in hash-chain form over a distributed ledger. The main function of this such system is to provide cryptographically secure method to share immutable series of records or blocks. This series of blocks is shared between the participants in a peer-to-peer network. New blocks can be added to the blockchain using a hash function after the nodes in network have been consent on this addition. The record data is visible to all participants in the blockchain network and based on a consensus protocol, every member in the network can reject or accept any added record data. The consensus protocol which network nodes validate and maintain the ledger in the blockchain according to it is a set of rules that decide how nodes can reach a mutual agreement. Therefore, blockchain technology enabled distributed system to reach high security without the need to central authority as well as it boosted the presence of peer-to-peer systems [98]. Figure 3.7 shows the basic structure of blockchain system based on peer-to-peer network [99].

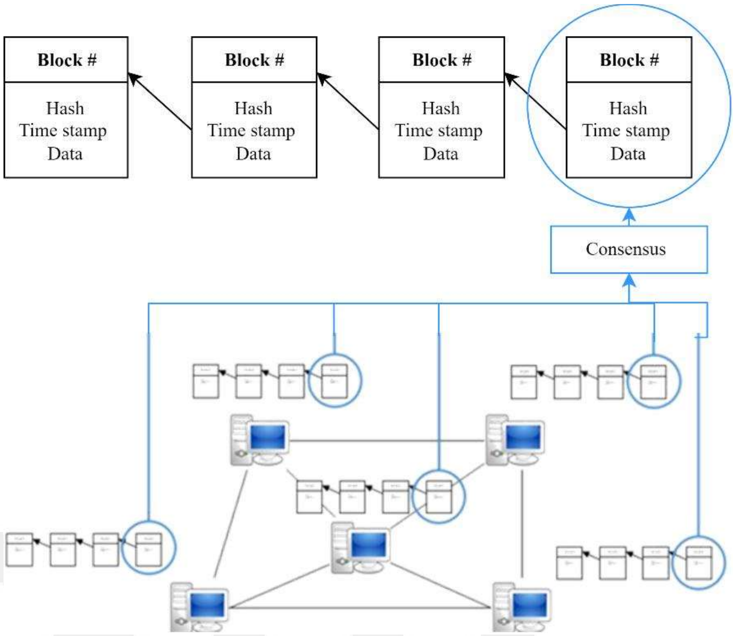

Figure 3.7: Blockchain system’s structure. [99].

Blockchains are rest on some cryptographic primitive and the most important one is hash functions. The main function of hash functions is to map a different length input to a predefined length output. Secure hash function must satisfy three requirements. First it must be preimage resistance which means that for a hash value h, it should require $\mathcal{O}(2^n)$ effort to compute an x such that $H(x) = h$. Also, it must be second preimage resistance that means for a given an input x and its hash value $h = H(x)$, it should require $\mathcal{O}(2^n)$ effort to find an $x' \neq x$ where $H(x') = h$. The third requirement is to be collision resistance which makes the require effort to find any two $x \neq x'$ such that $H(x) = H(x')$ have to be $\mathcal{O}(2^{n/2})$. In blockchain application, the presence of the hash value of the previous block in the current block makes it computationally difficult to make any modification previous blocks’ content. Such modification of previous blocks’ content would require calculating the second preimages of the hash function of the modified block or recalculation of all subsequent blocks. This condition makes the block that the blockchain structure has been extended beyond it much trust worthier as the blockchain has been extended because of the difficulty modification of this block. The other cryptographic primitive used in blockchain is digital signature (public/private key signature). Entities in the blockchain network are often identifiable by the possession of a public/private key combination that enables them to sign transactions and engage with the network in other ways. These keys are often issued and verified by the central entity in some blockchain systems which called permissioned blockchains. This central entity subsequently operates as a certificate authority in a public-key infrastructure (PKI).

Blockchains are designed for conditions where there is no universal confidence among all players. However, according to this lack of universal confidence, blockchain networks require a distributed consensus method for block validation. This consensus method is known as consensus protocol and there are three most popular protocols: proof of work, proof of stake, and proof of elapsed time. Basically, in a proof of work based system, a node in the network succeeds in getting a block approved if it can show that it spent a certain amount of computing resources on it. Bitcoin network is an example of such system. Unless a single entity controls more than half of the total computing resources in the network, Sybil attacks are prevented by this need of such spend of significant resources. On another side there is proof of stake consensus protocol which is a combination of random selection and the wealth of the participants. This kind of consensus protocols are based on the assumption that participants who own a larger stake in the blockchain would be much circumspect and have a crucial interest in assuring blockchain's integrity. A less popular consensus protocol from the previous two is proof of elapsed time. Here consensus is reached by having every possible validator node requests a random waiting time from the used execution environment which is integrated in the computing platform. Every node waits for the allocated period, and the first node finish its waiting period takes the lead.

Interestingly, blockchain technology was a hot topic since it first started in 2008 by Satoshi Nakamoto [100]. Indeed, blockchain technology was adapted for many applications like finance, healthcare, business, supply chain and industry. However, there is an opportunity to employ blockchain in all peer-to-peer or distributed systems.

Because of its special characteristics, Blockchain technology was a promising solution for federated learning systems. Ordinary federated learning depends on a central server to aggregate and averaging devices local models updates to generate a global model which is distributed after that to all devices. This reliance on the central server makes the system vulnerable with single point of failure. Instead, by utilizing blockchain in place of the central server we can overcome this vulnerability. Blockchain network enables exchanging local models' weights of network devices though the distributed ledger.

As suggested in [101] Blockchain-based federated learning system consists of devices that generate local models according to their local datasets and miners which maintain the distributed ledger. Any subset of network devices can be the miners, those can be end devices or edge devices according to power and computation constraints. In this federated learning network, each device generates its local model and train it with its local private data. After that it transmits model's weights to its nearest associated miner. The miners verify models weights they receive and transmit them to other miners according to the used gossiping protocol and add new blocks to the ledger according to consensus protocol. Each device finds out the global model's weights from the updated ledger. Figure 3.8 presents the basic structure of blockchain based federated learning [102].

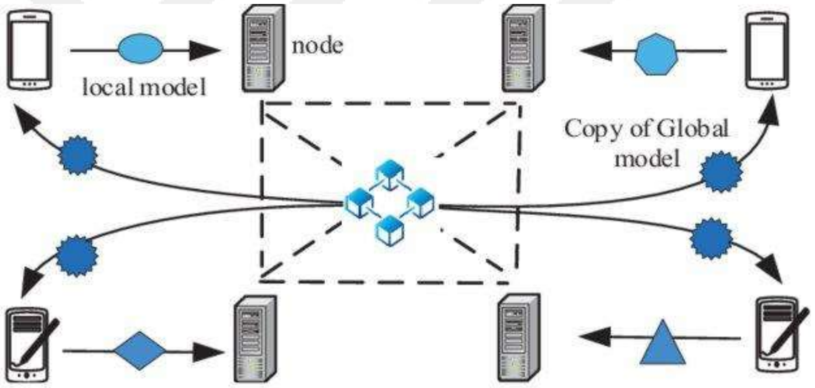

Figure 3.8: Blockchain-based federated learning. [102].

Blockchain is a promising solution to solve collaborative intrusion detection system issues. As we mentioned previously, data privacy and mutual trust between nodes are the main challenges facing collaborative intrusion detection systems. To preserve data privacy users can share transformed form of data instead of raw data over the blockchain to help other users to verify their IDS models as well as maintain transparency between network members. Also, collaborative IDS have to have reliable approach for alert exchange between various IDS nodes. Alert exchange helps to detect anomalies occurrence in the system and to decide the trustworthiness of each node.

Collaborative IDS can employ blockchain to maintain trust among IDS nodes. The generated alerts by IDS node can be handled as transactions in the ordinary blockchain systems. If an alert is replicated by multiple nodes, then the nodes adopted a consensus protocol collaboratively to validate the alert before inserting it in a block. In this way the system can store alerts in the ledger in tamper resistant way. Figure 3.9 shows how peer-to-peer network's nodes can exchange alerts and reach consensus on a detected intrusion [14].

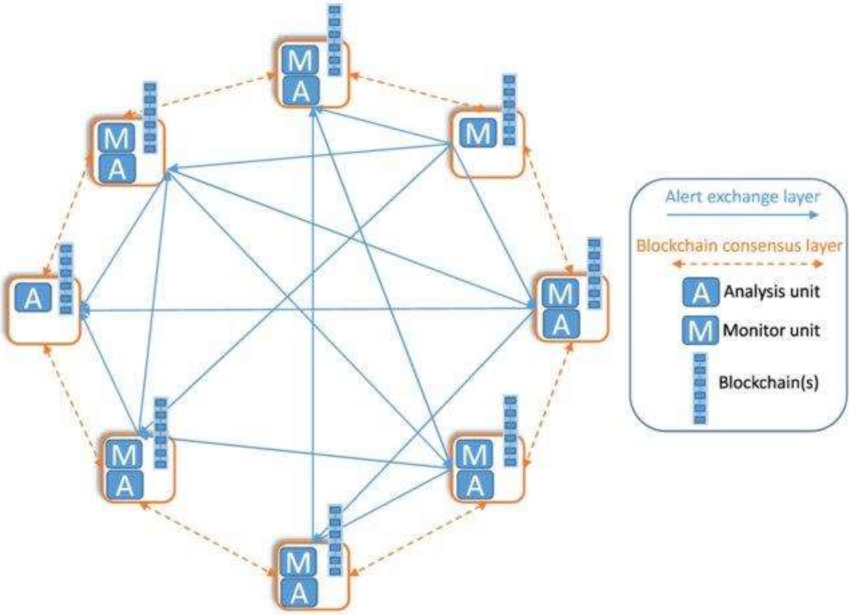

Figure 3.9: Alert exchange in blockchain based IDS. [14]

4. SYSTEM ARCHITECTURE AND OPERATION

In this work, we proposed a collaborative distributed intrusion detection system for IoT environments. This proposed architecture supports the security of the system operating in an IoT environment from several aspects. First, the privacy of each device's data is preserved by letting each device build its model locally without sharing data but sharing only model parameters. Even if the device needs to share a part of its data, it will be encoded. Additionally, the problem of a single point of failure is eliminated by replacing the central server with blockchain in the intrusion detection phase. Also, mutual trust can be managed between network nodes by the suggested alert exchange method using blockchain, whereas devices might need to share an encoded version of their data which is suspected to represent an intrusion.

4.1 System model

we build our proposed system on the architecture from [10][101][103]. By incorporating the principles, they introduced in their work like federated learning, distributed GAN, and blockchain the proposed system will be more suitable for the characteristics of the IoT environment and its operational conditions.

Let's consider an IoT system that consists of a set of devices D with n devices $D = \{d_1, d_2, d_3, \dots, d_n\}$ and each device d_i has its own set of private data samples X_i. So, the complete set of data samples in the system can be denoted by $X = \{X_1 \cup X_2 \cup X_3 \cup \cdots \cup X_n\}$.

The federated learning model of the system consists of a federated Autoencoder and a Multi-Discriminator Generative Adversarial Network. Each IoT device trains a local autoencoder and shares its parameters w_{z_i} with the central server that averages all models' parameters according to the federated learning loss function that we have mentioned before

$$\min \boldsymbol{F}(\boldsymbol{w}), \boldsymbol{where}\ \boldsymbol{F}(\boldsymbol{w}) = \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{m}} \boldsymbol{p_k F_k}(\boldsymbol{w}) \tag{4.1}$$

Where

$$\boldsymbol{F_k(w)} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{x'_i} - \boldsymbol{x_i})^{\mathbf{2}} \qquad \textbf{(4.2)}$$

for all data samples $x_i$and its reconstruction result x'_i in each local dataset X_i.

After training the federated autoencoder the central server discards the decoder while saving the encoder and broadcasts its parameters over the network of the IoT devices. So, all devices end up having the encoder E after the training phase.

The other part of the system's federated learning process is the Multi-Discriminator Generative Adversarial Network (MD-GAN) which consists of a single central generator G and a discriminator D_i in each device d_i. The goal is to train the generator using the whole data X which is the aggregation of all partial datasets owned by all IoT devices. MD-GAN makes GAN a 1-N game instead of being 1-1 game where G faces all D_i. IoT devices use their local datasets to train their discriminator to differentiate the generated data coming from the central generator from real data coming from their local datasets X_i.

Additionally, the proposed system will include a blockchain to store alerts and manage trust between IoT system devices during the intrusion detection phase. Let's consider that the system consists of a set of devices D with n devices. Any partial set of D can act to be the miners of the blockchain system. However, according to the resource-constraint characteristic of IoT end devices, we need to limit the computational load sustained by them. So, we suggest letting the edge devices maintain a lightweight proof-of-work consensus protocol and be the set of the miners M in the system. So, the contributed nodes in the system are the end devices that trigger alerts when an anomaly occurs in their local data according to their intrusion detection local model edge devices that collect the triggered alerts and establish a consensus to add them into the blockchain.

4.2 System Operation

The operations of the proposed system can be divided into two phases training phase and the intrusion detection phase. The system's life cycle starts with the training phase and moves forward to the intrusion detection phase. As an IDS system, the proposed system spends most of its operational time in the intrusion detection phase. However, in order to keep the system updated and tune the distributed discriminators, the system might perform incremental training periodically.

4.2.1 Training Phase

During this phase, the system set up the federated learning model. This phase consists of two steps, training the federated autoencoder and training the multi-discriminator generative adversarial network.

First of all, to train the federated autoencoder, each IoT device trains a vanilla autoencoder using its local dataset. After that, it shares the weight of the local model w_{z_i} with the central server which aggregates all models' weights and creates a single global model. The central server discards the decoder and saves the encoder E with weight w and broadcasts it over all devices. Using the encoder E each device encodes its local data samples into the latent space *z.*

The second step in the training phase is training the multi-discriminator generative adversarial network. The generator of the central server generates a set of artificial samples with a size equal to the latent space *z* using random noise. Those artificially generated samples are divided randomly into batches and distributed over the IoT devices in the system in such a way that each device gets two distinct batches let call them X_i^g and X_i^t. Thereafter, each IoT device trains its discriminator D_i using its local dataset X_i after encoded it into the latent space, and one batch of the artificial samples given from the central generator X_i^g. Next, each IoT device calculates error feedback using its discriminator D_i and the other batch given from the central generator X_i^t. Then each device sends the feedback to the central server. Consequently, the central server optimizes and updates the central generator *G* parameters by computing the gradient of its loss function using the feedback with the optimization algorithm.

At the end of the training phase, each IoT device ends to have a discriminator and an encoder. The discriminator is trained to differentiate anomalies that are simulated by the artificial sample generated by the generator. While using the encoder to reduce feature space before starting training GAN reduce the needed computational power and time as well as it will serve in data hiding in the next phase. The system architecture in the training phase is explained in Figure 4.1.

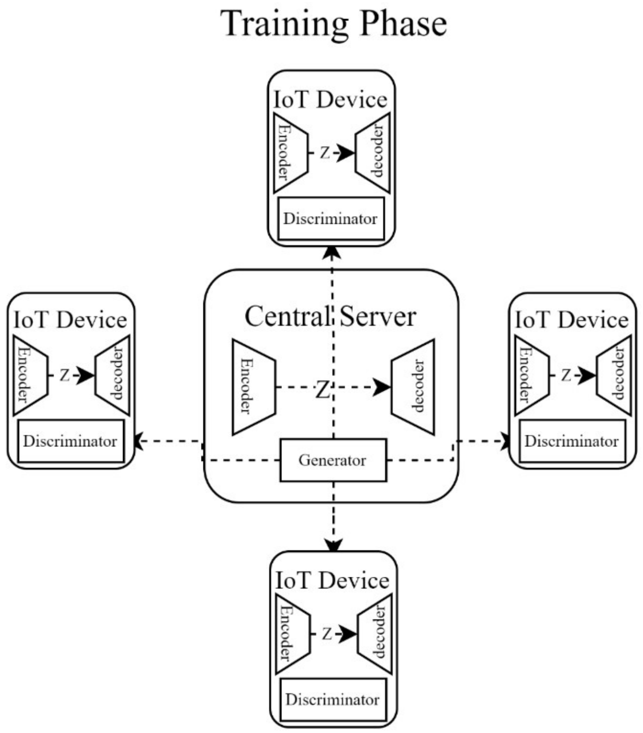

Figure 4.1: System architecture in the training phase.

4.2.2 Intrusion Detection Phase

During the intrusion detection phase, each IoT device watches its local data and detects abnormal behavior using its discriminator. If a device detects an anomaly, it triggers an alert and shares an encoded version of its data along with its discriminator's parameters. Using model parameters and the encoded data, other devices can check the truthfulness of the triggered alert and gossip it if that device was honest. After running consensus protocol (lightweight proof of work), the edge devices add the detected intrusion into the blockchain along with the alert's related information. This way, the privacy of the data of devices stays preserved and at the same time, the nodes in the system can examine the trustworthiness of each other. Figure 4.2 shows the architecture of the system during the intrusion detection phase. However, to keep devices' discriminators updated the system might do periodically incremental training.

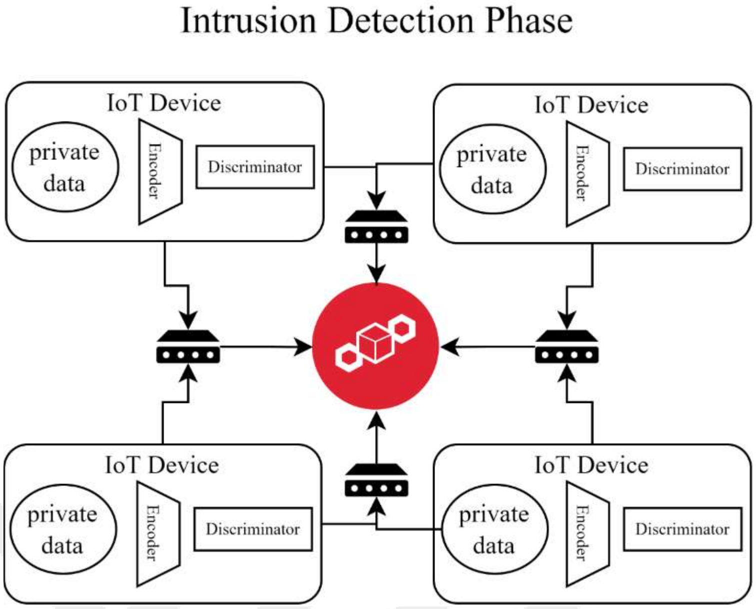

Figure 4.2: System architecture in the intrusion detection phase.

5. CONCLUSION

In the sake of securing IoT systems we proposed in this work a structure of collaborative distributed intrusion detection system. The proposed solution employs federated machine learning to train a global autoencoder model and multi discriminator generative adversarial network. The autoencoder will serve in term of feature reduction to improve the efficiency of the model by reducing the needed computational power and storage and communication capacity, as well as it serves in hiding data before sharing it with other parties in the system. On the other hand, the multi-discriminators of generative adversarial network are working as classifiers in each device to detect anomalies from the normal behavior.

To avoid reliance on a central server in detection phase, the proposed solution suggests employing blockchain as a medium to exchange alerts of intrusions together with event's related information (weight of the detecting discriminator with the encoded version of data). Based on the consensus protocol performed by a subset of nodes in the system, alerts are added to the distributed ledger. Since the alert is shared along with event's related information, the trustworthiness of each node can be decided by the other nodes in the system.

The future work can be focused on adapting the suggested solution to serve different systems' structures. Also, we also suggest developing a lightweight consensus protocol to suit the capabilities of the nodes in the IoT environment and does not cause a burden to the system.

REFERENCES

[1] O. Novo, N. Beijar, and M. Ocak, “Capillary Networks - Bridging the Cellular and loT Worlds,” *IEEE World Forum on Internet of Things (WF-IoT)*, vol. 1, pp. 571–578, December 2015.

[2] F. Hussain, Internet of Things; Building Blocks and Business Modles. Springer, 2017.

[3] Vailshery, Lionel Sujay. “Topic: Internet of Things (IoT).” Statista, www.statista.com/topics/2637/internet-of-things/, accessed at 29 Jun 2021.

[4] M. Ammar, G. Russello, and B. Crispo, “Internet of things: A survey on the security of iot frameworks,” *Journal ofInformation Security and Applications*, vol. 38, pp. 8 – 27, 2018.

[5] I. Farris, T. Taleb, Y. Khettab, and J. Song, “A survey on emerging sdn and nfv security mechanisms for iot systems,” *IEEE Communications Surveys Tutorials*, vol. 21, pp. 812–837, Firstquarter 2019.

[6] A. Ferdowsi and W. Saad, “Deep learning for signal authentication and security in massive Internet-of-Things systems,” *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, Feb 2019.

[7] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine Learning in IoT Security: Current Solutions and Future Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020, doi: 10.1109/COMST.2020.2986444.

[8] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, “IoT security techniques based on machine learning,” *arXiv*, no. September, pp. 41–49, 2018.

[9] W. Li, W. Meng, and M. H. Au, “Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments,” *Journal of Network and Computer Applications*, vol. 161, p. 102631, 2020.

[10] A. Ferdowsi and W. Saad, “Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things,” *arXiv*, pp. 1–6, 2019.

[11] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, “When intrusion detection meets blockchain technology: A review,” *IEEE Access*, vol. 6, pp. 10179–10188, 2018, doi: 10.1109/ACCESS.2018.2799854.

[12] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, ‘‘A trust-aware, P2P-based overlay for intrusion detection,’’ *in Proc. DEXA Workshop*, 2006, pp. 692–697.

[13] J. Sotos and D. Houlding, ‘‘Blockchains for data sharing in clinical research: Trust in a trustless world,’’ *Intel, Santa Clara*, CA, USA, Blockchain Appl. Note 1, 2017

[14] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivanko, and M. Muhlhauser, ‘‘Towards blockchain-based collaborative intrusion detection systems,’’ *in Proc. Int. Conf. Critical Inf. Infrastruct. Secur.*, 2017, pp. 1–12.

[15] Denning DE. An intrusion-detection model. *IEEE Transactions on Software Engineering* 1987;SE-13:222–32.

[16] Lunt TF. A survey of intrusion detection techniques. *Computers & Security* 1993;12:405–18.

[17] Mukherjee B, Heberlein LT, Levitt KN. Network intrusion detection. *IEEE Network* 1994;8:26–41.

[18] Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion detection systems. Computer Networks 1999;31:805–22

[19] Axelsson S, Intrusion detection systems: a survey and taxonomy, Chalmers University of Technology, Sweden, Technical Report 99-15 (2000), pp. 1–27.

[20] Mishra A, Nadkarni K, Patcha A, Tech V. Intrusion detection in wireless ad-hoc networks. *IEEE Wireless Communications* 2004;11:48–60.

[21] Krugel C, Toth T, A survey on intrusion detection systems, Technical University of Vienna, Austria, Technical Report TUV-1841-00-11 (2000), pp. 22–33

[22] Jones AK, Sielken RS, Computer system intrusion detection: a survey, University of Virginia, Technical Report (2000).

[23] Delgado N, Gates Q, Roach S. A taxonomy and catalog of runtime software-fault monitoring tools. *IEEE Transactions on Software Engineering* 2004;30:859–72

[24] Estevez-Tapiador JM, Garcia-Teodoro P, Diaz-Verdejo JE. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications* 2004;27:1569–84.

[25] Anantvalee T, Wu J. A survey on intrusion detection in mobile ad hoc networks. In: Xiao Y, Shen X, Du D-Z, editors. *Wireless/mobile network security*. SpringerVerlag; 2007. p. 170–96.

[26] Mandala S, Ngadi MA, Abdullah AH. A survey on MANET intrusion detection. *International Journal of Computer Science and Security* 2008;2:1–11.

[27] Amer SH, Hamilton JA. Intrusion detection systems (IDS) taxonomy—a short review. *Journal of Software Technology* 2010;13.

[28] Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. *Computers & Security* 2009;28:18–28.

[29] Xie M, Han S, Tian B, Parvin S. Anomaly detection in wireless sensor networks: a survey. *Journal of Network and Computer Applications* 2011;34:1302–25.

[30] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019.

[31] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[32] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J Netw Comput Appl*, vol. 60, pp. 19–31, 1// 2016

[33] Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. *Springer International Publishing*, Cham, pp 149–155

[34] Kreibich C, Crowcroft J (2004) Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput Commun Rev* 34(1):51–56

[35] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys &amp*; Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[36] J. Cannady, “Artificial neural networks for misuse detection,” *in Proc. 1998 Nat. Inf. Syst. Secur. Conf., Arlington*, VA, USA, 1998, pp. 443–456.

[37] R. P. Lippmann and R. K. Cunningham, “Improving intrusion detection performance using keyword selection and neural networks,” *Comput. Netw*., vol. 34, pp. 597–603, 2000.

[38] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, and M. Embrechts, “Network-based intrusion detection using neural networks,” *Intell. Eng. Syst. Artif. Neural Netw*., vol. 12, no. 1, pp. 579–584, 2002.

[39] H. Brahmi, B. Imen, and B. Sadok, “OMC-IDS: At the cross-roads of OLAP mining and intrusion detection,” *in Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 2012, pp. 13–24.

[40] D. Apiletti, E. Baralis, T. Cerquitelli, and V. D’Elia, “Characterizing network traffic by means of the NetMine framework,” *Comput. Netw*., vol. 53, no. 6, pp. 774–789, Apr. 2009.

[41] W. Chimphlee, A. H. Abdullah, M. Noor Md Sap, S. Srinoy, and S. Chimphlee, “Anomaly-based intrusion detection using fuzzy rough clustering,” *2006 International Conference on Hybrid Information Technology*, 2006.

[42] F. Jemili, M. Zaghdoud, and A. Ben, “A framework for an adaptive intrusion detection system using Bayesian network,” *in Proc. IEEE Intell. Secur. Informat*., 2007, pp. 66–70

[43] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, “Bayesian event classification for intrusion detection,” *in Proc. IEEE 19th Annu. Comput. Secur. Appl. Conf*., 2003, pp. 14–23.

[44] R. Hendry and S. J. Yang, “Intrusion signature creation via clustering anomalies,” *in Proc. SPIE Defense Secur. Symp. Int. Soc. Opt. Photonics*, 2008, pp. 69730C–69730C.

[45] C. Kruegel and T. Toth, "Using decision trees to improve signaturebased intrusion detection," *in Proc. 6th Int. Workshop Recent Adv. Intrusion Detect*., West Lafayette, IN, USA, 2003, pp. 173–191

[46] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "2014 Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, Apr. 2014.

[47] J. Zhang, M. Zulkernine, and 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.

[48] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting botnet command and control servers through large-scale netflow analysis," *in Proc. 28th Annu. Comput. Secur. Appl. Conf. (ACSAC'12)*, Orlando, FL, USA, Dec. 3–7, 2012, pp. 129–138

[49] D. Ariu, R. Tronci, and G. Giacinto, "HMMPayl: An intrusion detection system based on hidden Markov models," *Comput. Secur.,* vol. 30, no. 4, pp. 221–241, 2011.

[50] S. S. Joshi and V. V. Phoha, "Investigating hidden Markov models capabilities in anomaly detection," *in Proc. ACM 43rd Annu. Southeast Reg. Conf*., 2005, vol. 1, pp. 98–103

[51] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. decision trees in intrusion detection systems," in Proc ACM Symp. Appl. Comput., 2004, pp. 420–424.

[52] R. Agrawal and R. Srikant, "Mining sequential patterns," *in Proc. IEEE 11th Int. Conf. Data Eng.*, 1995, pp. 3–14.

[53] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, 2012.

[54] F. Amiri, M. Mahdi, R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for IDSs," J. Netw. Comput. Appl., vol. 34, no. 4, pp. 1184–1199, 2011.

[55] W. J. Hu, Y. H. Liao, and V. R. Vemuri, "Robust support vector machines for anomaly detection in computer security," *in Proc. 20th Int. Conf. Mach. Learn*., 2003, pp. 282–289.

[56] C. Wagner, F. Jérôme, and E. Thomas, “Machine learning approach for IP-flow record anomaly detection,” *in Networking 2011*. New York, NY, USA: Springer, 2011, pp. 28–39.

[57] W. Li, “Using genetic algorithms for network intrusion detection,” *in Proc. U.S. Dept. Energy Cyber Secur. Group 2004 Train*. Conf., 2004, pp. 1–8.

[58] S. Khan, “Rule-based network intrusion detection using genetic algorithms,” *Int. J. Comput. Appl.,* vol. 18, no. 8, pp. 26–29, Mar. 2011.

[59] S. Mukkamala, A. H. Sung, and A. Abraham, “Modeling intrusion detection systems using linear genetic programming approach,” *Innovations in Applied Artificial Intelligence*, pp. 633–642, 2004.

[60] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” *Proceedings of the 23rd international conference on Machine learning* - ICML '06, 2006.

[61] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in internet of things,” *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[62] C. Liu, J. Yang, Y. Zhang, R. Chen, J. Zeng, Research on immunity-based intrusion detection technology for the Internet of Things, *in: Natural Computation (ICNC), 2011 Seventh International Conference on*, Vol. 1, 2011, pp. 212–216.

[63] P. Kasinathan, C. Pastrone, M. Spirito, M. Vinkovits, Denial-of-service detection in 6LoWPAN based Internet of Things, in: Wireless and Mobile Computing, Networking and Communications (WiMob), *2013 IEEE 9th International Conference on*, 2013, pp. 600–607

[64] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, M. A. Spirito, DEMO: An IDS framework for internet of things empowered by 6LoWPAN, *in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS ’13, ACM, New York, NY, USA, 2013, pp. 1337–1340.

[65] Ioulianou, Philokypros, Vasilakis, Vasileios, Moscholios, Ioannis and Logothetis, Michael (Accepted: 2018) A Signature-based Intrusion Detection System for the Internet of Things. *In: Information and Communication Technology Form*, 11-13 Jul 2018. (In Press)

[66] K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (IDPS), Tech. rep., *National Institute of Standards and Technology*, special Publication 800-94 (2007).

[67] E. Cho, J. Kim, C. Hong, Attack model and detection scheme for botnet on 6LoWPAN, *in: C. Hong, T. Tonouchi, Y. Ma, C.-S. Chao (Eds.), Management Enabling the Future Internet for Changing Business and New Computing Services*, Vol. 5787 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 515–518.

[68] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, M.-C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN, in: Y.-M. Huang, H.-C. Chao, D.-J. Deng, J. J. J. H. Park (Eds.*), Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, Vol. 260 of Lecture Notes in Electrical Engineering, Springer Netherlands, 2014, pp. 1205– 1213.

[69] D. H. Summerville, K. M. Zach, Y. Chen, Ultra-lightweight deep packet anomaly detection for Internet of Things devices, in: *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, IEEE, 2015, pp. 1–8.

[70] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in Internet of Things, *International Journal of Computer Applications* 121 (9) (2015) 1–9.

[71] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy and H. Ming, "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0305-0310, doi: 10.1109/CCWC.2019.8666450.

[72] Y. Intrator, G. Katz, and A. Shabtai, “MDGAN: boosting anomaly detection using multi-discriminator generative adversarial networks,” *CoRR*, vol. abs/1810.05221, 2018.

[73] A. Farooqi, F. Khan, Intrusion detection systems for wireless sensor networks: A survey, in: D. Sl˛ezak, T.-h. Kim, A.-C. Chang, T. Vasilakos, M. Li, K. Sakurai ´ (Eds.), *Communication and Networking*, Vol. 56 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2009, pp. 234–241.

[74] S. Raza, L. Wallgren, T. Voigt, SVELTE: Real-time intrusion detection in the Internet of Things, *Ad Hoc Networks* 11 (8) (2013) 2661 – 2674.

[75] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the RPLbased Internet of Things, *International Journal of Distributed Sensor Networks* 2013.

[76] D. Oh, D. Kim, W. W. Ro, A malicious pattern detection engine for embedded security systems in the Internet of Things, *Sensors* 14 (12) (2014) 24188–24211.

[77] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things, *in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM),* 2015, pp. 606–611.

[78] A. Le, J. Loo, Y. Luo, A. Lasebae, Specification-based IDS for securing RPL from topology attacks, *in: Wireless Days (WD)*, 2011 IFIP, 2011, pp. 1–3.

[79] A. Le, J. Loo, K. K. Chai, M. Aiash, A specification-based IDS for detecting attacks on RPL-based network topology, Information 7 (2) (2016) 25.

[80] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, J. Isoaho, Distributed internal anomaly detection system for Internet-of-Things, *in: 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, pp. 319–320.

[81] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, “Towards blockchain-based Collaborative intrusion detection systems,” *Critical Information Infrastructures Security*, pp. 107–118, 2018.

[82] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, “A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, 2020, doi: 10.1109/jiot.2020.2996590.

[83] H. Liang, J. Wu, S. Mumtaz, J. Li, X. Lin, and M. Wen, “SECURE WIRELESS COMMUNICATIONS FOR VEHICLE-TO-EVERYTHING MBID: Micro-Blockchain-Based Geographical Dynamic Intrusion Detection for V2X,” *IEEE Commun. Mag.*, vol. 57, no. October, pp. 77–83, 2019, doi: 10.1109/MCOM.001.1900143.

[84] B. Hu, C. Zhou, Y. Tian, Y. Qin, and X. Junping, “Using Blockchain for Multimicrogrid Systems,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 8, pp. 1720–1730, 2019, doi: 10.1109/TSMC.2019.2911548.

[85] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, “Federated Learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2020, doi: 10.2200/S00960ED2V01Y201910AIM043.

[86] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *in Proc. Advances Neural Information Processing Systems Conf.*, 2014, pp. 2672–2680.

[87] Karpathy, A. (2015). Convolutional Neural Networks for Visual Recognition. Lecture notes CS231n. Stanford University. Stanford, CA. Retrieved from http://cs231n.github.io/neural-networks-1/

[88] V. Kazak “UNSUPERVISED FEATURE EXTRACTION WITH AUTOENCODER FOR THE REPRESENTATION OF PARKINSON'S DISEASE PATIENTS,” M.S. thesis, NOVA Information Management School, Lisbon, 2018. Accessed on: Des 19, 2021. [Online]. Available: http://hdl.handle.net/10362/71589

[89] V. K. Ojha, P. Dutta, H. Saha, and S. Ghosh, “Detection of Proportion of Different Gas Components Present in Manhole Gas Mixture Using Backpropagation Neural Network,” *Technology*, vol. 37, no. Icint, pp. 11–15, 2012.

[90] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Deep learning for visual understanding: Part 2 generative adversarial networks,” *IEEE Signal Process. Mag.*, no. January, pp. 53–65, 2018.

[91] V. K. Ojha, P. Dutta, H. Saha, and S. Ghosh, “Detection of Proportion of Different Gas Components Present in Manhole Gas Mixture Using Backpropagation Neural Network,” *Technology*, vol. 37, no. Icint, pp. 11–15, 2012.

[92] Schlegl, T., Seebock, P., Waldstein, S. M., Schmidt-Erfurth, ¨U., and Langs, G. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. abs/1703.05921, 2017. URL http://arxiv.org/abs/1703.05921.

[93] Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. Efficient GAN-Based Anomaly Detection. abs/1802.06222, 2018. URL http://arxiv.org/abs/1802.06222.

[94] Akcay, S., Abarghouei, A. A., and Breckon, T. P. GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. abs/1805.06725, 2018. URL http://arxiv.org/abs/1805.06725.

[95] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, “A Survey on GANs for Anomaly Detection,” 2019, [Online]. Available: http://arxiv.org/abs/1906.11632.

[96] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, “Autoencoders,” *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 193–208, 2019, doi: 10.1016/B978-0-12-815739-8.00011-0.

[97] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. *In NIPS’11 Proceedings of the 24th International Conference on Neural Information Processing Systems* (pp. 2546–2554). Granada, Spain. Retrieved from https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf.

[98] B. A. Tama, B. J. Kweka, Y. Park and K. -H. Rhee, "A critical review of blockchain and its current applications," *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 2017, pp. 109-113, doi: 10.1109/ICECOS.2017.8167115.

[99] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, “Decentralized Applications: The Blockchain-Empowered Software System,” *IEEE Access*, vol. 6, no. September, pp. 53019–53033, 2018, doi: 10.1109/ACCESS.2018.2870644.

[100] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” URL: http://www.bitcoin.org/bitcoin.pdf, 2008.

[101] H. Kim, J. Park, M. Bennis, and S. L. Kim, “Blockchained on-device federated learning,” *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, 2020, doi: 10.1109/LCOMM.2019.2921755.

[102] U. Majeed and C. S. Hong, “FLchain: Federated Learning via MEC-enabled Blockchain Network,” *2019 20th Asia-Pacific Netw. Oper. Manag. Symp. Manag. a Cyber-Physical World, APNOMS 2019*, no. September, 2019, doi: 10.23919/APNOMS.2019.8892848.

CURRICULUM VITAE

Name Surname: Nouha Hejazi

B.Sc.: Information Systems, King Khalid University (2015 – 2019) first class honor and 4.99/5 GPA.

Professional Experience:

- Freelancer Tutor for high school students (2016-2019)
- Freelancer Software Developer (2019-2021)

Rewards:

- King Khalid University Honors Prize in Information System Bs. Degree.
- Ministry of Education Cup for Secondary Schools, First place, Science Section.

List of Publications and Patents:

- Blue Brain: The Virtual Brain Technology, IJSTE, Volume 5, 2018, 68-71pp.