REPUBLIC OF TÜRKİYE
KIRŞEHİR AHİ EVRAN UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ADVANCED TECHNOLOGIES

AUTOMATIC DETECTION OF INTRUSION ATTACKS IN IOT NETWORKS USING BI-LSTM-CNN NEURAL NETWORK

SINDIBAD ALI FAYYADH FAYYADH

MASTER'S THESIS

KIRŞEHİR
2023

REPUBLIC OF TÜRKİYE
KIRŞEHİR AHİ EVRAN UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ADVANCED TECHNOLOGIES

AUTOMATIC DETECTION OF INTRUSION ATTACKS IN IOT NETWORKS USING BI-LSTM-CNN NEURAL NETWORK

SINDIBAD ALI FAYYADH FAYYADH

MASTER'S THESIS

SUPERVISOR
ASSIST. PROF. DR. AYLA KAYABAŞ

KIRŞEHİR
2023

KIRŞEHİR AHİ EVRAN UNIVERSITY INSTITUTE OF NATURAL AND APPLIED SCIENCES MASTER'S THESIS ETHICS DECLARATION

In this thesis study, which I have read and understood the Kırşehir Ahi Evran University Scientific Research and Publication Ethics Directive and which I have prepared in accordance with the Kırşehir Ahi Evran University Institute of Science Thesis Writing Rules;

● I have obtained the data, information and documents I have presented in the thesis within the framework of academic and ethical rules,

● I present all information, documents, evaluations and results in accordance with scientific ethical rules,

● I have cited all the works I have benefited from in the thesis by making appropriate references,

● I have not made any changes in the data used and the results,

● This study, which I have presented as a thesis, is original,

Otherwise, I declare that I accept all legal actions to be taken against me in this regard and all loss of rights that may arise against me. …/…/20…

Sindibad Ali Fayyadh FAYYADH

LIST OF CONTENTS

Page No

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to:

- Assist. Prof. Ayla KAYABAŞ, my supervisor, for hThisuidance, support, valuable insights and suggestions, and encouragement throughout this research.

- My family for their love, support, and sacrifices. They have always been there for me, no matter what.

I am grateful for all the help and support I have received. This research would not have been possible without their support.

September, 2023 Sindibad Ali Fayyadh FAYYADH

ÖZET

YÜKSEK LİSANS TEZİ

IOT AĞLARINDA BI-LSTM-CNN SİNİR AĞI KULLANILARAK SIZMA SALDIRILARININ OTOMATİK ALGILANMASI

Sindibad Ali Fayyadh FAYYADH

KIRŞEHİR AHİ EVRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İLERİ TEKNOLOJİLER ANABİLİM DALI

Danışman: **Dr. Öğr. Üyesi Ayla KAYABAŞ**
Yıl: 2023, Sayfa: 95
Jüri: **Dr. Öğr. Üyesi Ayla KAYABAŞ**
Doç. Dr. Mustafa YAĞCI
Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN

Günümüzün hızla gelişen teknoloji ortamında, siber saldırılar ve olağandışı olayların artması, çeşitli sorunlara yol açabilecek bu düzensizliklerin ivedilikle tespit edilmesinin gerekliliğini vurgulamaktadır. Kötü niyetli saldırı sorunu, nesnelerin İnterneti (IoT) ağlarında kendini daha fazla göstermektedir, bu sorunun önemi de ilgi yoluyla oluşan ağ tarafından sunulan uygulamaların eleştirilmesiyle ortaya çıkmaktadır. Güç ağı, büyük imalat endüstrileri vb. Güç ağları, büyük imalat endüstrileri, vb. ağındaki herhangi bir düzensizlikten kaynaklı ciddi bir tehditle karşı karşıya kalacaktır. Bu saldırılar virüsler, saldırılar, kesintiler ve daha fazlası dahil olmak üzere birçok biçimde gerçekleşebilmektedir. Anomali tespiti tam da burada devreye girmektedir– veri akışlarında norm dışı olan bu sıra dışı olayları dikkatlice tanımlamanın bir yoludur. Bu zorluğun üstesinden gelmek için iki tekniği birleştiren yeni bir yaklaşım salık vermekteyiz: çift yönlü uzun kısa süreli bellek (Bi-LSTM) ve evrişimli sinir ağları (CNN). Bu eşleştirme, içerisinde anomalilerin daha hızlı ve daha doğru şekilde tanımlanmasını sağlayabilme potansiyelini bulundurmaktadır. Bu zorluğun üstesinden gelmek için iki tekniği birleştiren yeni bir yaklaşım öneriyoruz: çift yönlü uzun kısa süreli bellek (Bi-LSTM) ve evrişimli sinir ağları (CNN). Yöntemimiz, özel araçlar kullanan kontrollü bir siber ortamda derinlemesine test ettiğimiz köklü UNSW-NB15 veri kümesi üzerine tesis edilmiştir. Modelimizi daha da iyi hale getirmek için rektifiye edilmiş linear ağırlıkların kullanıldığı bir teknik ekledik. Bu ağırlıklar CNN kurulumlu Bi-LSTM için özel olarak tasarlanmıştır ve standart ağırlıkların üzerinde ilerleme sağlamaktadır. Yöntemimizi mevcut yöntemlerle karşılaştırdık ve etkileyici sonuçlarla karşı karşıya kaldık. Modelimiz %94.14'lük bir algılama doğruluğu elde ederek (Barrons, March,29,2021) 'i %10,18, (Yin et al., 2023).'yı %13,89 ve (Jain et al., 2022)'yi %14,40 geride bıraktı. Bu, yaklaşımımızın anomalileri hızlı ve doğru bir şekilde tespit etmede son derece etkili olduğunun ayrıca gelişmiş siber güvenliğe katkıda bulunduğu anlamına gelmektedir.

Anahtar Kelimeler: Veri Seti, Bi-LSTM, CNN, UNSW-NB15, RNN.

ABSTRACT

MASTER'S THESIS

AUTOMATIC DETECTION OF INTRUSION ATTACKS IN IOT NETWORKS USING BI-LSTM-CNN NEURAL NETWORK

Sindibad Ali Fayyadh FAYYADH

KIRŞEHİR AHİ EVRAN UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ADVANCED TECHNOLOGIES

Supervisor:	**Assist. Prof. Ayla KAYABAŞ**
	Year: 2023, Pages: 95
Juries:	**Assist. Prof. Ayla KAYABAŞ**
	Assoc. Prof. Mustafa YAĞCI
	Assist. Prof. Ali Osman ÇIBIKDİKEN

In today's fast-paced technological landscape, the rise in cyber attacks and unusual incidents highlights the urgent need to quickly spot these irregularities, which can lead to various problems. The problem attacks and anomalies in Internet of Things (IoT) networks are more paramount; the significance of it is manifested by the criticality of the applications served by the network through the interest. Power networks, large manufacturing industries, etc. Will face a severe threat to any irregulations in IoT network. These attacks can take many forms, including viruses, hacks, disruptions, etc. That's where anomaly detection comes in – it's a way to carefully identify these unusual occurrences in data streams that stand out from the norm. To tackle this challenge, we suggest a new approach that combines two techniques: bidirectional long short-term memory (Bi-LSTM) and convolutional neural networks (CNN). This pairing has the potential to make identifying anomalies faster and more accurate. Our method is built on the well-established UNSW-NB15 dataset, which we've tested extensively in a controlled cyber environment using specialized tools. To make our model even better, we've added a technique using rectified linear weights. These weights are specially designed for the Bi-LSTM with CNN setup and provide improvements over the standard weights. We have compared our method with existing ones, and our results are impressive. Our model achieves a detection accuracy of 94.14%, surpassing by 10.18,by 13.89%, and by 14.40%. This means our approach is highly effective in quickly and accurately identifying anomalies, contributing to enhanced cybersecurity.

Keywords: Dataset, Bi-LSTM, CNN, UNSW-NB15, RNN.

GENİŞLETİLMİŞ ÖZET

Sızma Tespit Sistemleri (IDS) ve uygulamaları, özellikle şifreleme alanındaki veri güvenliğini ilerletmede önemli rolü bulunmaktadır. Veri hacminin artması ve veri işleme teknolojilerindeki gelişmeler, bulut tabanlı uygulamaların artmasını ve bu yolla ağ trafiğinde muazzam artışa sebep olmuştur. Yine veri boyutunun artması ile birlikte hem bireysel hem de kurumsal veri depolama hizmet ihtiyacını da arttırmıştır. Verinin bulut ortamında depolanması aynı zamanda güvenlik riskini de arttırmıştır. Bu gelişmelerle birlikte aynı zamanda izinsiz verileri açığa çıkarma ve gizlilik ihlalleri gibi güvenlik ve gizlilik riskleri önemli ölçüde artmıştır.

Bilgisayar ve mobil elektronik cihazlar dışında olan neredeyse tüm cihaz ve eşyaların internete erişiminin gerçekleştiği günümüzde güvenlik ihlalleri de benzer düzeyde artmıştır. Nesnelerin İnterneti (IoT), sensörleri, yazılım ve çeşitli teknolojileri birbirine bağlayarak internet üzerinden bağlı cihazlar ve sistemler arasında veri alışverişini kolaylaştıran geniş bir uygulamadır. Finans, tarım, taşımacılık ve sağlık gibi sektörlerde uygulama alanları bulmaktadır. Veri güvenliğini veri şifrelemesi ve etkili güvenlik önlemleri aracılığıyla sağlamak, siber saldırılar ve izinsiz girişimlere karşı korunmada esas öneme sahiptir. IDS'ler, veriyi kablosuz ortamlarda ve farklı veri depolama senaryolarında korumak için hayati öneme sahiptir.

IOT sistemleri, veri güvenliğini artırmayı ve gelişmiş şifreleme teknikleri aracılığıyla kolay erişimi mümkün kılmayı amaçlayan sistemlerdir. Bununla birlikte, önemli verileri korumak ve izinsiz açığa çıkarmayı önlemek için güçlü güvenlik önlemlerine ihtiyaç duyulması gibi önemli bir mesele ile karşı karşıya kalınmaktadır. Kullanıcılar, veri erişimi konusunda dikkatli olmalı, potansiyel tehditleri tanımalı ve bunları etkisiz hale getirmek için proaktif önlemler almalıdır.

IOT alanında IDS'ler, gelişmiş gizlilik ve güvenlik önlemleri aracılığıyla çeşitli tehditleri tespit ederek ve bunların sıklığını azaltarak veri güvenliği konusunda güven sağlama konusunda kritik bir rol oynarlar.

Söz konusu araştırma, dünya genelindeki İnternet of Things (IOT) cihazlarının hızlı yayılması ve yaygın kullanımının yanı sıra pratik uygulamaların ortaya çıkması nedeniyle yapılmıştır. Bu IoT cihazlarının hızlı büyümesi aynı zamanda kötü niyetli faaliyetlerin ve güvenlik ihlallerinin artmasına neden olmuştur. Bilgi güvenliği uzmanları ve teknik araştırmacılar sürekli olarak siber suçlular tarafından istismar edilen güvenlik açıklarını tespit

etmekte ve bu durum güvenlik ihlallerine, gizlilik ihlallerine ve kullanıcı güvenliği risklerine yol açmaktadır.

IoT peyzajındaki güvenlik tehditleri daha yaygın ve çeşitli hale gelmiştir, bu da etkili bir Intrusion Detection System (IDS) uygulamanın aciliyetini vurgulamaktadır. IoT ağlarında dış ve iç kaynaklı birçok saldırı ve tehdit ortaya çıkabilmektedir. Bu saldırılar genellikle IoT ekosisteminin en savunmasız düğümlerini hedef alarak veri güvenliği ve gizliliği için büyük bir risk oluşturmaktadır.

Bu çalışma, IoT ağlarıyla ilişkili güvenlik ve gizlilik sorunlarını ele almayı amaçlamaktadır. Bu kapsamda etkili bir Intrusion Detection System (IDS) geliştirilmesi hedeflenmektedir. Amaç, IoT'de sızma ve anormallikleri tespit etmek için Long-short-term memory (BI-LSTM) ve convolutional neural networks (CNN) mimarilerini birleştiren bir hibrit çerçeve kullanarak karmaşık özellikleri hem etiketli hem de etiketlenmemiş verilerden çıkarabilen bir çerçeve geliştirmektir.

Ağa bağlı sistemler içindeki hacking, sızma ve veri ihlalleri endişelerinin artması özellikle de IoT ağlarına yapılan saldırıların artması bu çalışmanın önemini bir kez daha ortaya koymaktadır. Elden edilen sonuçlar IoT ağlarına yönelik saldırıları önemli ölçüde engelleyeceği düşünülmektedir. Artan siber saldırı sıklığı göz önüne alındığında, kritik verileri korumak ve bilgisayar ağlarının bütünlüğünü sürdürmek için güçlü güvenlik protokolleri oluşturmak son derece önemlidir. Bu araştırmanın temel amacı, etkili bir Intrusion Detection System (IDS) uygulayarak IoT ağlarında kullanıcı verilerinin güvenliğini ve gizliliğini artırmaktır.

Bu konu, finans gibi endüstrilerde önemli bir öneme sahiptir, çünkü güvenlik ihlalleri büyük finansal kayıplara yol açabilir. IoT ağlarında bir IDS kullanmak veri korumasının ötesine geçerek aynı zamanda ağı merkezileştirme ve düzenleme, güvenli ve gizlilik odaklı bir ekosistem oluşturmaya da katkı sağlamaktadır. Nesnelerin İnterneti (IoT) bağlamında Intrusion Detection Systems (IDS) kullanmanın dikkate değer bir başka yönü, geniş veri ve ağ trafiği hacmini etkili bir şekilde yönetme kabiliyetidir. Bu sayede izin verilen kullanıcıların güvenlik veya gizliliği tehlikeye atmadan verilere erişebilmeleri sağlanmaktadır.

Dijital dünyada, izinsiz veri erişimi hem organizasyonlar hem de bireyler için önemli riskler oluşturmaktadır. Bu riskleri ele almak için, İzinsiz Giriş Tespit Sistemleri (IDS'ler), çeşitli endüstrilerde geniş çapta benimsenen ağ güvenliği araçlarıdır. IDS'ler, bir ağ içindeki potansiyel zafiyetleri ve kötü niyetli faaliyetleri tanımak için tasarlanmıştır.

IDS'ler sürekli olarak ağ trafiğini analiz eder ve herhangi şüpheli davranışı sistem yöneticilerine bildirirler. Ancak IDS'lerin zafiyetleri önlemek veya ortadan kaldırmak gibi bir

yetenekleri olmadığını, ancak erken tehdit tespitinde önemli bir rol oynadıklarını belirtmek önemlidir.

Güvenlik duvarları yeni tehditlerin ağa girmesini engelleme ve önleme üzerine odaklanırken IDS'lerin temel amacı potansiyel tehditleri tanımaktır. IDS'ler potansiyel tehditleri tanır ve ayırt eder, bunları ortadan kaldırmak için ek önlemlere bağlı kalmaktadırlar. Hem donanım hem de yazılım IDS çözümlerini kullanmak, kuruluşların anormallikleri etkili bir şekilde tanımlamalarına ve izinsiz girişleri azaltmalarına yardımcı olur.

IoT sistemleri bağlamında, geleneksel Sızma Tespit Sistemleri (IDS), IoT cihazlarının sınırlı depolama ve hesaplama kapasiteleri nedeniyle etkisiz olabilir. IoT ortamlarına özgü uzmanlaşmış IDS çözümleri, kritik IoT verilerinin gizliliğini ve bütünlüğünü korumak için gereklidir. IDS'ler, hızlı bir şekilde hizmet reddi saldırıları ve port taramaları dahil olmak üzere çeşitli kötü niyetli davranışları tanımlayabilir ve potansiyel riskleri azaltabilirler.

Bu çalışma aynı zamanda IDS çerçevesi içinde denetimli ve denetimsiz modelleri birleştirmenin önemini de ortaya koymaktadır. Bu yaklaşım, sızma tespitinin doğruluğunu artırmayı ve yarı denetimli bir öğrenme metodolojisi aracılığıyla geniş IoT verilerini işlemenin zorluklarını ele almayı amaçlamaktadır. Genel olarak, bu çalışma, özellikle gizlilik, doğruluk ve gizlilik odaklı veri yönetimi alanlarında IoT ağları için güvenli çözümlerin ileri taşınmasına katkıda bulunmaktadır.

Önerilen metodoloji, Sızma Tespit Sistemlerindeki (IDS) doğruluk ile ilişkili sorunları ele almayı amaçlar ve bunu, birbirine bağlı uzun kısa vadeli bellek (Bi-LSTM) modeli ve evrişimsel sinir ağları (CNN) ile rektife edilmiş doğrusal ağırlıklar içeren bir hibrit mimari kullanarak yapmaktadır. Bu yaklaşım, anormallik tespiti ve sınıflandırma doğruluğunu artırmaktadır.

Yöntemin etkinliği, eksik verileri ele alma, aykırı değerleri ortadan kaldırma, özellikleri ölçeklendirme, kategorik verileri kodlama ve SMOTE örnekleme gibi çeşitli veri ön işleme teknikleri kullanılarak UNSW-NB15 veri kümesi ile gösterildi. Bu ağırlıklar CNN kurulumlu Bi-LSTM için özel olarak tasarlanmıştır ve standart ağırlıkların üzerinde ilerleme sağlamaktadır. Yöntemimizi mevcut yöntemlerle karşılaştırdık ve etkileyici sonuçlarla karşı karşıya kaldık. Modelimiz %94,14'lük bir algılama doğruluğu elde ederek 'i %10,18,'yı %13,89 ve 'yi %14,40 geride bıraktı. Bu, yaklaşımımızın anomalileri hızlı ve doğru bir şekilde tespit etmede son derece etkili olduğunun ayrıca gelişmiş siber güvenliğe katkıda bulunduğu anlamına gelmektedir.

LIST OF TABLES

Page No

LIST OF FIGURES

Page No

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviations		Described
AIDS	:	Anomaly Intrusion Detection System
AUC	:	Area Under the ROC Curve
Bi-LSTM	:	Bidirectional Long Short-Term Memory
CNN	:	Convolutional Neural Network
CPNN	:	Convolution and Pooling Neural Network
DDoS	:	Distributed Denial of Service
DL	:	Deep Learning
DoS	:	Denial of Service
DS	:	Data Security
DT	:	Decision Tree
EDA	:	Exploratory Data Analysis
FCL	:	Fully Connected Layer
FCNN	:	Full Connection Neural Network
FLN	:	False Negative Rate
FLP	:	False Positive Rate
HIDS	:	Host Based Intrusion Detection System
ICT	:	Information and Communication Technology
IDS	:	Intrusion Detection System
IoMT	:	Internet of Medical Things
IoT	:	Internet of Things
IPS	:	Intrusion Prevention System
LDA	:	Linear Discriminant Analysis
LSTM	:	Long Short-Term Memory
ML	:	Machine Learning
NIDS	:	Network Based Intrusion Detection System
NSL-KDD	:	Network Security Laboratory – Knowledge Discovery İn Database
PCA	:	Principal Component Analysis
QoS	:	Quality of Service
R2L	:	Root to Local
RNN	:	Recurrent Neural Network
ROC	:	Receiver Operating Characteristic Curve
SIDS	:	Signature Based Intrusion Detection System
SMOTE	:	Synthetic Minority Over Sampling
SVM	:	Support Vector Machine
TRN	:	True Negative Rate
TRP	:	True Positive Rate
VANET	:	Vehicular Ad Hoc Network

1. INTRODUCTION

1.1. Overview

The proliferation and advancement of the Internet of Things (IoT) have resulted in a high volume of data transmission, which presents opportunities for identifying and analyzing breaches within the IoT ecosystem. Although attempts have been made to annotate the traffic seen in the Internet of Things (IoT), the number of labeled records used in this process has decreased. This decrease in the number of labeled records contributes to heightened challenges and complexities in the identification of anomalies and the detection of unfamiliar threats within IoT data. Despite these limitations, there is a growing commitment to address the complexities associated with task recognition and intrusion detection challenges (Abdel-Basset et al., 2021). Currently, the global count of interconnected devices exceeds 25 billion, a figure that surpasses the global human population by a factor of three. The advancement of the Internet of Things (IoT) revolves around intelligent devices and diverse services that integrate intricate and varied functions into a single domain, thus facilitating the execution of tasks. This integration enables the consolidation of various tasks into a unified system, enhancing inter-device communication with greater efficiency and supporting decision-making processes through the use of sensor data. Additionally, it enables remote control. Clients, in their capacity as users, employ mobile applications and web services to gain access to data and exert control over their devices. Furthermore, The Internet of Things (IoT) infrastructure depends on many sensors to collect accurate and detailed data. In addition, the information and data are subjected to analysis by artificial intelligence algorithms limited (Alkahtani et al., 2021). Limited. The aforementioned constraint presents a significant difficulty in accurately identifying and mitigating breaches within the Internet of Things (IoT) network (Abd Elkhalik et al., 2023).

It is important to note that the primary objective of an Intrusion Detection System (IDS) is to protect information within a given system by upholding the principles of confidentiality; the proliferation of the Internet of Things (IoT) and its associated devices has resulted in an expanded attack surface for cybercriminals, making it easier for them to execute more powerful and devastating cyber-attacks. Consequently, there has been a significant increase in the number of cyber-attacks targeting computer systems and the data housed within cloud computing environments. These attacks often have malicious intent and employ novel and creative methods to infiltrate the data.

Hence, the utilization of machine learning (ML) techniques in an anomaly-based Intrusion Detection System (IDS) is employed to identify and categorize threats within the Internet of Things (IoT) environment. In situations where there exist complex networks and associated technologies in diverse infiltration approaches, conventional machine learning techniques and algorithms may demonstrate limited efficiency or effectiveness in managing such data. In contrast, deep learning (DL) techniques exhibit enhanced efficiency and proficiency in precisely detecting anomalies. Currently, hackers frequently exploit IoT networks that lack necessary updates and encryption, thereby gaining unauthorized access to important data and compromising vulnerable IoT devices. Although there have been advancements in security systems in comparison to previous methods, many intrusion systems are now attempting to bypass security measures such as smart locks and even garage doors (Ashraf et al., 2021).

In terms of technology, the Internet of Things (IoT) integrates internet connectivity with tangible entities, facilitating the exchange of data and information among diverse items. Nevertheless, the heightened execution orders of the Internet of Things (IoT) and its corresponding networks have raised significant security concerns. The limited computational efficiency exhibited by Internet of Things (IoT) devices renders them susceptible to a multitude of security breaches and concerns. In light of these issues, the significance of Intrusion Detection Systems (IDS) has been recognized, leading to the deployment of various tools aimed at bolstering the security of information within interconnected networks. In recent decades, there has been a significant surge in the adoption of Internet of Things (IoT) technology, resulting in notable improvements in operational efficiency. The Internet of Things (IoT) is utilized to establish connectivity between various devices and the Internet, hence enabling the seamless exchange of information. The aforementioned interconnected devices can be remotely controlled and accessed through the internet, regardless of time or geographical location. The Internet of Things (IoT) can be broadly characterized as the amalgamation of the internet and tangible items. IoT is widely utilized in various domains such as healthcare, industrial automation, residential automation, and other sectors (Farahani, et al., 2018).

Moreover, intrusion detection refers to systematically examining intelligent devices and network resources to detect and identify unauthorized actions. These unauthorized access attempts might manifest in individual devices or across entire networks, undermining their overall security. The perpetuation of illicit operations by attackers and hackers might result in the impeding of authentic user responses. An

Intrusion Detection System (IDS) is a specialized security mechanism that is designed to safeguard system integrity by detecting and identifying illegal activities. An Intrusion Detection System (IDS) can efficiently circumvent traditional firewalls to facilitate accurate detection of unauthorized access attempts. Intrusions manifest when information security is compromised, resulting in losing its confidentiality, availability, and essential integrity. The preservation of data confidentiality is ensured when it is only available to authorized users (Sugi et al., 2020). In a similar vein, Machine learning (ML) plays a crucial part in the field of anomaly detection by aiding in the identification and detection of deviations from pre-established profiles. In the context of conventional supervised learning, which is frequently employed as a classification technique, it is necessary to have pre-existing training samples accompanied by accurate ground-truth labels. However, within the practical framework of the Internet of Things (IoT), examples that have been labeled are frequently scarce, but unlabelled data is more readily available globally. The rationale behind this phenomenon can be attributed to the substantial financial investment and extensive human resources involved in the process of data labeling, along with the indispensable requirement for specialized knowledge in the relevant field. To tackle these issues, using semi-supervised learning algorithms has emerged as a potentially effective option. The algorithms can utilize the vast amount of unlabelled data and autonomously assign labels to it, eliminating the requirement for human involvement (Li et al., 2020). With the pervasive integration of the Internet into various aspects of human endeavors, there has been a notable surge in the connectivity of numerous gadgets to this global network. IoT devices have attained widespread adoption across diverse domains of human existence. Nevertheless, the Internet of Things (IoT) sphere presents numerous intriguing issues with inadequately defined solutions. The resolution of security and technological issues associated with the Internet of Things (IoT) continues to provide a substantial obstacle, with specific approaches still being developed. Machine learning (ML) and deep learning (DL) methodologies, encompassing both unsupervised and supervised methods, are employed for monitoring and analyzing logs produced by Internet of Things (IoT) sensors. In the context of extensive frameworks, the number of records might potentially exceed millions daily, hence necessitating the utilization of machine learning (ML) and deep learning (DL) Equipment needed to conduct adequate analysis (Liu et al., 2022).

Adversarial assaults, a subject of substantial research in computer vision, present an area of ongoing investigation within network security. The proliferation of the Internet

of Things (IoT), 5G technology, along with artificial intelligence (AI), has led to a corresponding rise in the likelihood of security incidents and occurrences inside the IoT domain. As a result, there has been an increase in the utilization of deep learning techniques to identify and address security vulnerabilities. The expansion of the IoT ecosystem is accompanied by a corresponding increase in security threats and vulnerabilities. The circumstances above give rise to a significant demand for heightened security protocols and catalyze hackers to seek novel methods to exploit IoT devices. Hence, there is a pressing need to enhance current systems and methodologies to build effective security solutions that effectively mitigate the emerging security vulnerabilities inside the Internet of Things (IoT) domain (Ibitoye et al., 2019). Thus, the proliferation of technology has facilitated convenience in daily life, although it has also created many security vulnerabilities. The advent of the internet has witnessed a concomitant rise in cyber threats aimed at compromising the security of online resources. In the world of information security, Intrusion Detection Systems (IDS) are a valuable defense layer that creates a secure environment for organizations and protects against network intrusions. Nowadays, there has been a utilization of machine learning (ML) Equipment in intrusion detection systems (IDS) to augment their capacities in the identification and categorization of security threats (Da Costa et al., 2019).

1.2. Importance and types of IDS

The use of Intrusion Detection Systems (IDS) has become an integral aspect in the realm of computer utilization and system security. Several intrusion detection systems (IDS) provide business and industrial organizations with enhanced network visibility, hence assisting in the adherence to security laws. In addition, established industries can effectively incorporate IDS logs into their documentation, serving as a crucial means of showcasing their commitment to compliance obligations. Intrusion detection systems (IDS) utilization has increased. The potential to greatly bolster security measures through the promotion of reliability and adherence to established terms and conditions (Demirkan et al., 2020). An Intrusion Detection System (IDS) serves a vital function in the surveillance of both inbound and outbound network operations, as well as the detection of indicators of unauthorized access that might potentially jeopardize the integrity of the system or the confidentiality of the data. The primary purpose of this system is to provide an alert in response to the detection of harmful behavior, making it a passive monitoring system. An Intrusion Detection System (IDS), alternatively known as an Intrusion

Prevention System, can identify irregularities and proactively intervene to impede unwanted actions within the organization's network. Two primary methods for implementing these Intrusion Detection Systems (IDS): are both host-based and network-based intrusion systems. Solutions for network-based intrusion detection (IDS) include hardware components and sensors that are carefully positioned at critical locations inside the network of an organization or corporation (Chiba, et al., 2022). In addition to network-based IDS, individual systems can also be equipped with host-based IDS, capable of analyzing outbound traffic on the network. Intrusion Detection Systems (IDS) can identify and alter various types of files, servers, and critical files. The system employs a comprehensive database based on signatures, enabling it to generate alerts and dispatch relevant notifications. Using signature-based techniques helps these intrusion detection systems (IDS) reduce the likelihood of false positive alerts. Additionally, they contribute to the maintenance of regulatory compliance and enhance the level of visibility and control over the network. There are numerous advantages linked with the utilization of Intrusion Detection Systems (IDS).

- The task of monitoring and evaluating specific threats is undertaken.
- Measures are implemented to avoid Distributed Denial of Service (DDoS) and Denial of Service (DoS) assaults and intrusions on the network.
- The objective is to mitigate security breaches targeting the SSL protocol to find accessible and distinct network ports.
- The analysis and identification of efforts to fingerprint the operating system to initiate targeted attacks on the designated system have been examined (Chiba, et al., 2022).
- These Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) solutions do not perform their intended duties with sufficient skill despite proper configuration and deployment.

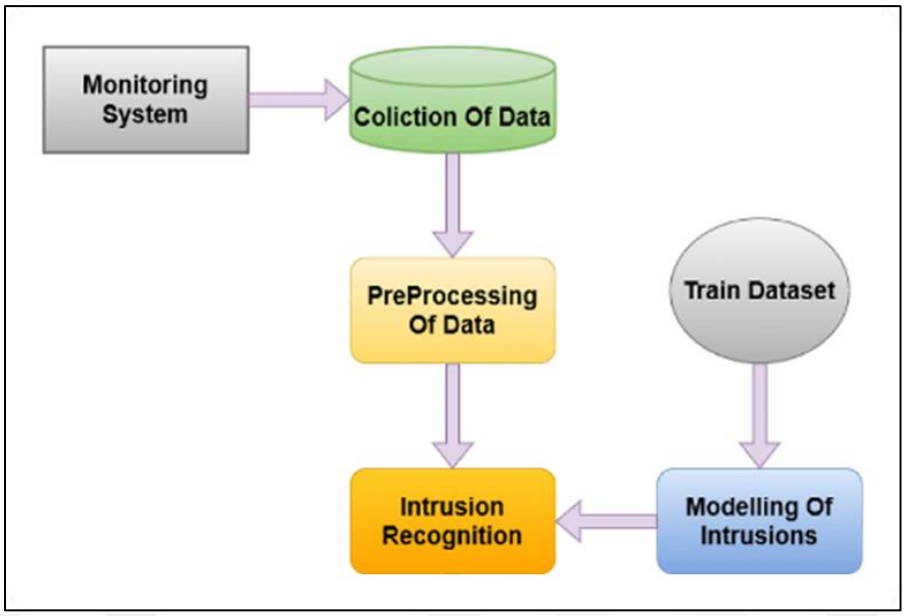

Figure 1.1. IDS / IPS system connectivity (Pratum, 2023)

Figure 1.1 depicts the existing cloud computing environment and its geographically distributed workforces, focusing on the security measures implemented through firewalls, which may exhibit deficiencies in their operational performance (Pratum, 2023).

An Intrusion Detection System (IDS) is a surveillance system utilized to identify and detect potentially malicious behaviors and actions, initiating an alert mechanism upon identifying deviations from normal patterns. Upon receiving this notification, the analyst or relevant individual in the Security Operation Center investigates the matter and takes necessary actions to address the identified threat. Adversaries or malicious actors may seek to breach a network by exploiting vulnerabilities present in devices and software. There are numerous benefits linked with the utilization of Intrusion Detection Systems (IDS). Centralized management is a key feature offered by these systems, facilitating the correlation of assaults and corresponding replies. To improve the effectiveness of IDS controls in an organization, managers can use IDS capabilities to analyze various types of attacks through pattern recognition (Hajj, et al., 2021). Anomaly-based detection, alternatively referred to as behavior-based detection, entails the alteration of user, network, and host system behaviors. An alarm is generated, and the administrator is promptly notified in the event of even a minor divergence in user behavior. In contrast, signature-based detection, alternatively referred to as knowledge-based detection, relies on a comprehensive database comprising records of previously seen threats. The approach employed in this methodology involves comparing the patterns seen in network traffic

with the signatures of documented attacks and their corresponding vulnerabilities (Kim et al., 2020). Hybrid-based intrusion detection represents a distinct category that integrates anomaly-based and signature-based detection methodologies. Numerous Intrusion Detection Systems (IDS) employ the abovementioned techniques to identify intrusions, as each technique has inherent limitations. Furthermore, Intrusion Detection Systems (IDS) can be categorized into two distinct categories according to their operational behavior: Active IDS and Passive IDS. The active intrusion detection system (IDS) not only performs the task of monitoring and analyzing network traffic, but it also proactively implements preventive measures by obstructing suspicious traffic within the cloud data environment. In contrast, Passive Intrusion Detection Systems (IDS) solely observe and evaluate network traffic, thereby notifying users of potential vulnerabilities in the early stages (Bhati et al., 2021).

In addition, there are five additional categories of Intrusion Detection Systems (IDS) (Vaigandla et al., 2022):

1- An intrusion detection system (IDS): is a network designed to monitor network traffic to detect and analyze potential intrusions.

2- Network: Node Intrusion Detection Systems (IDS) are designed to specifically target and identify intrusions aimed at individual network nodes or devices.

3-Host: Intrusion Detection Systems (IDS are created to observe and analyze the actions and conduct of individual host systems to identify potential intrusions.

4- Protocol-based: Intrusion Detection Systems (IDS) are designed to examine network protocols to detect and identify any irregularities or deviations that could signify unauthorized invasions.

5- The Application Protocol-based Intrusion Detection System (IDS) is designed to specifically target and detect intrusions that are aimed at particular application protocols.

1.3. Existing trends in IDS

The development of IDS systems has been driven by the quick progress of technologies and the evolution of network configurations. These systems have been designed to possess improved capabilities to effectively address the security demands of both conventional and emergent contexts. The detection technology utilized in Intrusion Detection Systems (IDS) has undergone significant advancements, encompassing network-based detection, wireless detection, and behavior-based anomaly detection.

These advancements are specifically designed to mitigate intrusion activities (Sharma et al., 2019). Hence, the proliferation and connectivity of devices have been greatly influenced by the Internet of Things (IoT), primarily driven by the constrained processing capabilities of these devices. According to estimates, the total number of internet-connected devices surpassed 50 billion by the conclusion of the year 2020.

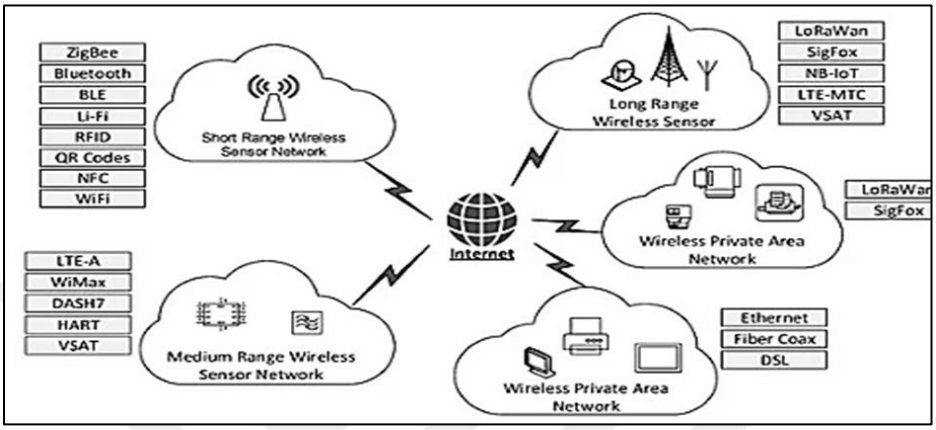

Figure 1.2. IoT-related access and protocols (Asharf et al., 2020)

Figure 1.2 depicts the system environment and the diverse array of practical applications that employ micro-computing devices featuring specialized protocols renowned for their energy efficiency. These applications encompass a wide range of fields and are facilitated using standardized communication and protocols. The figure shows various aspects of the Internet of Things (IoT), each depicting a different network access method or a set of diverse IoT communications and technologies. The architecture of the Internet of Things involves the integration of tangible objects with computing devices, hence facilitating the delivery of intelligent services to end-users. To facilitate the connecting of a vast number of heterogeneous devices throughout the Internet, it is important for the architecture to be structured in layers and possess adaptability (Asharf et al., 2020). One of the notable rising trends in Intrusion Detection Systems (IDS) pertains to the Network-based IDS, specifically developed to identify and thwart reconnaissance attacks before their potential infiltration into the internal network. Anomaly-based detection systems are beneficial because they can identify new attacks, especially when there are no established signatures or patterns. These detection algorithms have demonstrated their effectiveness in instances where they surpass the performance of ordinary traffic patterns (Alsoufi et al., 2021).

Utilizing network-based intrusion detection systems (IDS) presents several notable benefits, including achieving swifter reaction times than host-based IDS. Additionally, its implementation is reasonably straightforward as it does not necessitate modifying the current infrastructure or monitoring all aspects of an operating system on a target host. In contrast, Network Intrusion Detection Systems (NIDS) prioritize the surveillance of network segments to detect possible security breaches (Jyothsna et al., 2019). Furthermore, signature-based Intrusion Detection Systems (IDS) also provide significant benefits. These systems may be readily deployed and configured with minimal training time. SIDS exhibits the ability to efficiently handle recognized attacks at elevated velocities and exhibit reduced rates of false positives, hence leading to enhanced accuracy in detecting harmful events (Abhale, 2023). Additionally, all of these Intrusion Detection System (IDS) types can offer monitoring capabilities for the event logs of the entire network of host systems. All types of Intrusion Detection Systems (IDS) possess the ability to monitor event logs originating from the systems within the host network. This facilitates network managers in maintaining awareness of any modifications that transpire on the hosts. Through the surveillance of event logs, intrusion detection systems (IDS) provide administrators the ability to see the actions and occurrences taking place within the network, enabling them to efficiently identify and address any potentially dubious or illegal alterations. Implementing proactive monitoring allows administrators to uphold the network's security and integrity effectively while ensuring rapid responses to possible threats or breaches. IDS systems are commonly associated with a range of advantages, which include (Heidari et al., 2022):

1- The concept of high security refers to the implementation of measures and protocols aimed at minimizing the risk of unauthorized access, breaches, and threats to a particular system,

2- Adherence to legal regulations and broad societal acceptability

3- Anomaly detection is its ability to save both time and cost.

4- The utilization of automated processing in identifying and analyzing potentially harmful and dubious behavior and actions.

5- The encryption of data of significant importance and the corresponding management mechanisms have been extensively studied

In general, intrusion detection systems (IDS) offer many advantages regarding network security, regulatory adherence, operational effectiveness, and safeguarding of data, rendering them indispensable instruments for enterprises striving to protect their

networks and data resources. Another breakthrough linked to Intrusion Detection Systems (IDS) is the integration of Federated Learning (FL), a widely used technique for preventing data leaks and ensuring privacy in the Internet of Things (IoT). The utilization of Federated Learning (FL) enables IDS learning models to effectively tackle concerns related to the preservation of model parameters for safeguarding private data and maintaining the security of independent and identically distributed data. Federated learning (FL) also serves to alleviate the negative consequences of training on private data. Nevertheless, the practical implementation of the approach may face obstacles due to the amplified communication overhead resulting from the substantial sizes of the models. Furthermore, the utilization of FL has the potential to mitigate the occurrence of distillation failures resulting from inaccurate data projections (Agrawal et al., 2022).

Another rising trend in Intrusion Detection Systems (IDS) is their integration into In-vehicle networks (IVNs). Intrusion Detection Systems (IDS) possess the capability to identify and safeguard against cyber-attacks against Industrial Vehicle Networks (IVNs), thereby guaranteeing the integrity of their operational functionality and facilitating instantaneous communication. In this specific context, a comprehensive analysis is conducted on the attacks directed towards the external interfaces of cars. This analysis encompasses three distinct layers, namely the bus level, message level, and data flow level. Every vulnerability is subjected to individual analysis to build cutting-edge solutions. The implementation of Intrusion Detection Systems (IDS) in cars is categorized into three distinct sets of variations, which are bus level, message level, and data flow level. These variations are designed to cater to certain functional requirements, as stated in reference (Wu et al., 2019).

Within the domain of information security, Intrusion Detection Systems are frequently linked to the application of machine learning and deep learning methodologies to attain sophisticated findings and outcomes. These methodologies are well-suited for fulfilling the requirements of achieving high levels of accuracy and cost-effectiveness in the context of Network Intrusion Detection Systems (NIDS), thus effectively boosting the security of computer networks. The utilization of DL (Deep Learning) and ML (Machine Learning) methodologies in the domain of network security has experienced a surge in popularity, primarily attributed to the significant computational prowess offered by GPUs (Graphics Processing Units). Machine learning (ML) and deep learning (DL) methodologies have demonstrated efficacy in identifying and predicting abnormal activity using acquired patterns. Nevertheless, deep learning methodologies do not only

depend on the automated acquisition of intricate characteristics and unprocessed data, owing to the intricate architecture of the neural network. Utilizing deep learning techniques in network intrusion detection systems (NIDS) allows for identifying and detecting unauthorized individuals within the network's connection (Ahmad et al., 2021).

1.4. Problem Statement

The Intrusion Detection System (IDS) and its various applications have exhibited a diverse array of favorable characteristics in advancing and auguring data security systems, specifically in the realm of encrypted data. The proliferation of data and advances in data processing techniques have resulted in a surge in cloud-based traffic and the storing of an extensive quantity of data. Nevertheless, this advancement has also led to an increase in security and privacy risks, such as the unauthorized disclosure of data and violations of privacy. The Internet of Things is a prevalent application that establishes connectivity between sensors, software, and various technologies to enable data exchange among interconnected devices and systems via the Internet. The Internet of Things (IoT) is employed across many sectors like finance, agriculture, transportation, and healthcare. Implementing data encryption alongside effective security measures is imperative in several fields to safeguard against hacking and incursion endeavors. Intrusion Detection Systems (IDS) serve a crucial role in ensuring the security of data, even in wireless environments and across diverse data storage capacities. Using Internet of Things (IoT) solutions has become increasingly prevalent in administering high-value data using sophisticated management devices. Implementing robust intrusion detection and prevention systems is vital in safeguarding data against unethical activities and privacy infringements, hence guaranteeing the security and accessibility of data.

Moreover, Internet of Things (IoT) Systems aim to improve data security and enable convenient access when necessary by employing advanced encryption techniques. Nevertheless, a significant obstacle that must be addressed is the requirement to implement robust security measures to safeguard and protect valuable data, thereby mitigating the potential for unauthorized disclosure of information. This highlights the importance of users maintaining a state of vigilance regarding data accessibility, identifying potential dangers and attacks, and effectively neutralizing them. In the Internet of Things (IoT) realm, Intrusion Detection Systems (IDS) perform a crucial role by providing confidence in data safety, detecting a wide range of threats, and mitigating their frequency through implementing advanced data privacy and security measures.

1.5. Motivations

The impetus for this research arises from the exponential expansion and escalating utilization of Internet of Things (IoT) devices on a global scale, together with the emergence of pragmatic applications that heavily depend on instantaneous data and operations. The proliferation of Internet of Things devices has resulted in a concomitant rise in occurrences of malevolent activities and breaches of security. Information security professionals and technical researchers continuously identify vulnerabilities exploited by cybercriminals, resulting in system security breaches, privacy violations, and risks to user protection. The prevalence and diversity of security threats have experienced a substantial surge, hence emphasizing the criticality of performing an efficient Intrusion Detection System (IDS). Within the realm of Internet of Things (IoT) networks, a multitude of incursions and threats can arise from both external and internal origins. These attacks frequently focus on the most vulnerable nodes in the Internet of Things (IoT) ecosystem, thereby presenting a substantial threat to the security and privacy of data within the network. Hence, The main goal of this research is to investigate and effectively manage the security challenges and associated hazards of Internet of Things (IoT) networks. Researchers aim to analyze different types of threats in the IoT domain and develop effective IDS solutions. These solutions are intended to proficiently identify and avert assaults, thereby ensuring the preservation of data integrity and fortification of user privacy.

1.6. Aim and objectives

This study aims to address the security and privacy challenges associated with IoT networks by developing an effective Intrusion Detection System (IDS). The objective is to detect intrusions and anomalies in IoT using a hybrid framework that combines Long-short-term memory (BI-LSTM) with convolutional neural networks (CNN) architectures, which are capable of extracting complex features from both labeled and unlabeled data.

The specific objectives of this study are as follows:

To detect intrusions and anomalies in IoT by employing a hybrid framework of BI-LSTM and CNN, which can effectively extract complex features from the data.

1. To minimize the loss rate by integrating a Recurrent Neural Network (RNN). This integration allows for efficient data reconstruction and retention of the original data.

2. To evaluate the proposed framework's efficiency using performance indicators such as accuracy, precision, recall, and F1-score to recognize various intrusion attacks. The UNSW_NB 15 dataset will be utilized for this evaluation.

By achieving these objectives, the study aims to provide a secure IoT environment and effectively address the cybersecurity threats and challenges associated with IoT networks.

1.7. Significance of the study

The study's importance lies in addressing hacking, infiltration, and data breaches within network-connected systems, specifically focusing on IoT networks. Considering the escalating frequency of cyberattacks, it is imperative to establish resilient security protocols to safeguard critical data and uphold the integrity of computer networks. The objective of this study is to promote security and privacy for user data in IoT networks through the implementation of an effective Intrusion Detection System. The significance of this matter is particularly pronounced in industries such as finance because the ramifications of security breaches can lead to substantial financial deficits. Using an Intrusion Detection System in networks connected to the Internet of Things holds a broader relevance that extends beyond the mere protection of data. Furthermore, it plays a role in fostering the centralization and organization of the network, thereby establishing a safe and privacy-centric ecosystem. IoT networks can achieve the desired security results by strategically investing in the right design and technologies for IDS.

Furthermore, one notable aspect of employing Intrusion Detection Systems (IDS) in the Internet of Things (IoT) context is its capacity to effectively manage substantial data and network traffic volumes. This measure guarantees that the intended user can efficiently access and oversee data without compromising the integrity of security or privacy. The paper acknowledges the significance of integrating supervised and unsupervised models within the Intrusion Detection System (IDS) framework. This work aims to enhance the accuracy of intrusion detection and tackle the difficulties related to evaluating large volumes of IoT data using a semi-supervised learning methodology. The study's value primarily resides in its contribution to the advancement of secure solutions for IoT networks, specifically in the areas of secrecy, accuracy, and privacy-focused data management.

1.8. Scope of the study

This study addresses a wide range of data security and protection factors, including considerations within cloud-based and local network contexts. The primary focus is continuously monitoring and examining data to optimize security methods and guarantee effective data administration. Intrusion Detection Systems (IDS) are used to analyze diverse categories and volumes of attacks, rendering them suitable for implementation across various companies as a fundamental security mechanism. These tools properly manage data and aid enterprises in identifying and resolving network device configuration issues. The Intrusion Discovery System (IDS) is a cybersecurity monitoring system providing ongoing surveillance to identify and flag potentially harmful or questionable activity. It generates alerts to facilitate the discovery of anomalies. These warnings enable security management and operators to implement measures to safeguard the data promptly. Intrusion Detection Systems (IDS) are responsible for maintaining activity records and providing alerts to administrators in the event of a security breach. Proactive network maintenance is an additional facet of intrusion detection systems (IDS), wherein the cleaning and blocking of the cloud network is undertaken to safeguard network functionality. Implementing Intrusion Detection Systems (IDS) is strategically planned to optimize operational efficiency while minimizing any potential negative impact on network performance.

Additionally, the study emphasizes the preparation of data to address cyberattacks and criminal activities efficiently. The collection and analysis of data from many systems and network sources are employed to detect security breaches and assure preparedness for their mitigation. The main goal of this research is to augment the security of systems and data in Internet of Things (IoT) systems, providing comprehensive safeguarding measures throughout the entire data transmission process.

1.9. Thesis organization

The thesis is organized in a manner that aligns with the structure and content of the study. The following chapters are included:

Chapter 1: Introduction

This chapter provides an overview of IDS and IoT, discussing the current trends in IDS methods and their advantages in ensuring data security in IoT. The study's scope, significance, aim, and objectives are also presented in this chapter.

Chapter 2: Literature Review

This chapter explores existing methodologies and methods used in IoT and other relevant domains of IDS implementation. It discusses conventional algorithms employed as primitive security systems across various domains.

Chapter 3: Proposed Methodology

Chapter 3 focuses on the proposed methodology and the criteria used in the study, specifically utilizing the UNSW_NB 15 dataset. It describes the approach and techniques employed in the research.

Chapter 4: Results and Analysis

Chapter 4 demonstrates the outcomes of the suggested methods and provides an analysis of the overall performance. A comparison with existing methods and methodologies is also included to assess the suggested strategy's efficiency.

Chapter 5: Conclusion and Future Work

In this final chapter, the conclusion of the study is presented, summarizing the findings and their implications. Additionally, recommendations for future work and further research are provided, highlighting potential areas for improvement and exploration.

By organizing the thesis in this manner, he research makes sure the material is presented logically, from the introduction to the conclusion, enabling readers to understand the background, methodology, results, and implications of the research.

1.10. Summary

In Chapter 1, a thorough examination is presented of the aims associated with the implementation of Intrusion Detection Systems (IDS) inside the realm of the Internet of Things (IoT) and other sectors; in the realm of data privacy and security, A crucial role is played by intrusion detection systems (IDS). as a sophisticated and indispensable solution. The chapter elucidates the patterns and benefits associated with integrating IDS in real-time applications. This statement underscores the significance of implementing advanced security systems to facilitate efficient data retrieval and consumption while prioritizing better security and privacy protocols. Moreover, the chapter provides an in-depth analysis of the scope, significance, aim, and objectives of the study, establishing the fundamental framework for the subsequent research undertaken the following chapters.

2. LITERATURE REVIEW

2.1. Introduction

To identify and stop intrusions, an Intrusion Detection System (IDS)) is used to detect and identify anomalies in a network that can potentially cause damage. IDS can be classified as either network-based IDS or host-based IDS. Network-based IDS is typically installed on client computers, while host-based IDS resides on the network itself. IDS is commonly implemented as a software application on customer hardware, and cloud-based IDS is sometimes used to protect systems and data in cloud deployments. Employing IDS offers several benefits, including the detection of malicious activities, improved network performance through issue detection, and valuable insights for enhancing network security (Ahanger et al., 2022).

There are various types of IDS, such as NIDS, HIDS, SIDS, and AIDS. NIDS is deployed strategically to keep an eye on all devices' incoming and outgoing traffic inside the network. NIDS analyzes network threats, quickly detects and identifies malware and unknown threats, and can shut down processes and alert users upon intrusion detection. NIDS continuously monitors network traffic, detecting suspicious activity and blocking it before hackers can gain access to the system (Albasheer et al., 2022).

Another common type of IDS is HIDS, which can run on any computer or device having immediate access to both the workplace network and the internet. One advantage of HIDS over NIDS is its ability to identify anomalous network packets originating within the organization, which NIDS may fail to detect. HIDS monitors traffic to and from specific systems where the IDS software is installed and can oversee critical system files, alerting any attempts to overwrite these files. Studies have suggested that HIDS is more versatile and adaptable than NIDS (Yasmeen et al., 2022).

Signature-based IDS (SIDS) identify attacks based on specific patterns, such as the number of bytes and the presence of 1s and 0s in the network traffic. SIDS can also detect previously known malicious instruction sequences used by malware. These patterns detected by IDS are referred to as signatures. Therefore, SIDS can quickly identify signatures that already exist in the system, but it is challenging to see malware attacks with unknown patterns (Spadaccino et al., 2020).

Anomaly-based IDS, commonly known as AIDS, relies on rule-based detection rather than signatures. Although AIDS is accurate and effective, they are more suitable for detecting attacks (Khraisat et al., 2019). They can detect abnormalities at different

levels. AIDS is typically used to identify unknown malware attacks as new malware rapidly evolves. In AIDS, a trustworthy activity model is created using machine learning classifiers, and deviations from the model are considered suspicious, while adherence to the model is deemed unsuspicious. Machine learning-based approaches offer better adaptability compared to signature-based IDS, as anomaly-based IDS can be trained based on hardware configurations (Hamolia et al., 2020).

The Internet of Things (IoT) has witnessed significant growth and has found applications in various sectors, such as intelligent systems, smart agriculture, and smart transportation. These IoT systems consist of interconnected devices, including sensors, actuators, and other network-enabled devices. It is estimated that there will be around 75.3 billion actively connected IoT devices by 2025 (Gyamfi et al., 2022). Unlike traditional internet technology, IoT systems operate autonomously without human intervention, enhancing data processing capabilities.

The rapid growth of IoT devices has increased the demand for network bandwidth. However, IoT devices often have resource constraints, posing challenges in implementing conventional security methods to protect them from cyber-attacks. Security concerns become critical when sensitive information is transmitted within the IoT network. IoT is a network of devices that communicate with each other via the Internet. The three main components of IoT are the gateway, device, and cloud. Three levels make up the conventional Internet of Things architecture: the application, network, and perception layers (Jabraeil Jamali et al., 2020).

The application layer, located at the top of the system, receives data from the network layer and uses it to provide services to users. As the topmost layer, the application layer should ensure data integrity, reliability, and customer information protection. However, the security requirements for the data vary depending on the application context (Gupta et al., 2020).

The network layer in the IoT system is responsible for facilitating device interactions with the processing center. Positioned as the middle layer, the network layer plays a crucial role in coordinating information. However, the network layer is more susceptible to attacks compared to other layers, making it essential to implement robust security measures and adopt diverse technologies with varied features. Intrusion Detection Systems (IDS) have also focused extensively on the network layer (Ahmad et al., 2021).

The perception layer, also known as the sensor layer, is situated at the bottom of the IoT architecture. It is responsible for data collection through sensors. Ensuring the security of communication devices within the perception layer is essential. Traditional internet security standards are challenging to apply to IoT devices, particularly wireless ones that require more sophisticated and robust security measures (Khattak et al., 2019). Despite the layers offering robust security, attacks are still prevalent in IoT applications. These attacks Comprise user-to-root attacks, remote-to-local attacks, denial of service (DoS) attacks, wormhole attacks, Sybil attacks, hello flood attacks, sinkhole attacks, and so on (Hajiheidari et al., 2019).

DoS attacks frequently render computers or other devices unavailable to their intended users. These attacks typically involve external attempts by hackers or attackers to disrupt the legitimate use of a service. Another type of attack is the user-to-root attack, where the attacker gains access to a user's regular account and then expands their access to root privileges by exploiting vulnerabilities in the system (Bala et al.).

Wormhole attacks are also frequently observed in IoT systems. In the suggested study, two protocols, RPL and 6LoWPAN, were designed specifically for constrained devices in IoT to address the issue of wormhole attacks (Zainel et al., 2022). These protocols aim to mitigate the impact of such attacks on the IoT network. Furthermore, an Intrusion Detection System (IDS) was implemented in the suggested study to counterattack the attackers using the Contiki OS simulator and Cooja simulator. The IDS helps identify and respond to potential security breaches in the IoT system (Deshmukh-Bhosale et al., 2020).

Hence, implementing Intrusion Detection Systems (IDS) is crucial in overcoming security problems in IoT applications. IDS can provide enhanced security by detecting and responding to potential threats in the network. To achieve effective results, IDS can be combined with other IDS techniques. IDS are commonly classified based on their placement strategy, validation strategy, and detection method (Khraisat et al., 2021).

2.2. Applications of IDS

One of the significant concerns in utilizing IDS across various fields, particularly in IoT, is the protection against cyber threats. Cybersecurity is a highly researched topic and is essential for dealing with the challenges posed by IoT applications in areas such as home automation, healthcare, and agriculture. One notable IDS application is Passban, an intelligent process designed to protect directly connected IoT devices. Passban is cost-

effective and can be easily deployed on IoT gateways. It leverages edge computing to detect threats closer to the data source. Passban has demonstrated the capability to identify numerous malicious traffic types, including port scanning, HTTP, and SYN flood assaults, with acceptable levels of accuracy and minimal false positive rates (Eskandari et al., 2020).

Another application is the use of discrete probability for analyzing the propagation of attacks and cascading effects. This approach has proven effective in threat analysis. Additionally, the Cybersecurity Supply Chain (CSC) plays a crucial role in providing facilities for a safe network for organizations to meet their Goals for the company. Combining many technologies in the supply chain has streamlined company procedures, accelerated production, and cut distribution expenses. However, it has also resulted in increased interdependencies among stakeholders and the absence of methods involving other parties for auditing and addressing cyber threats. This can lead to manipulations in design specifications and alterations during the distribution process. By addressing these challenges and implementing advanced IDS technologies, organizations can enhance the security and resilience of their IoT systems and protect against online dangers that may compromise the integrity and functionality of the network and connected devices (Yeboah-Ofori et al., 2019).

Another application of IDS in business is Cyber Threat Attribution (CTA), which involves identifying the source responsible for initiating malicious activities. CTA provides valuable information to cybersecurity teams for mitigation and strategic planning. This is particularly crucial for detecting future attacks, especially in sectors such as finance and critical infrastructure. CTA indicators have proven to be more effective than traditional indicators like trace-back and firewalls. However, CTA can be deceptive and biased, as low-level indicators of compromise are rarely used and easily modified. Despite this, CTA achieves an accuracy rate of 83% and a precision rate of 33% (Noor et al., 2019).

In the field of agriculture, IDS has also found applications by integrating advanced technologies into existing farming operations to improve the quality and productivity of agricultural products. IoT has being frequently utilized in numerous agricultural applications to achieve the concept of Agriculture 4.0. However, the agriculture sector is also vulnerable to cyber-attacks and threats such as DDoS attacks, which can render services unavailable or inject false data even into functional agricultural equipment. IDS with deep learning (DL) approaches have been introduced to address these challenges,

utilizing datasets like CIC-DDoS2019 that contain various DDoS attacks (Ferrag Noor et al., 2021).

Another application of IDS in Farming involves utilizing Unmanned Aerial Vehicles (UAVs) equipped with remote sensing technology, which has rapidly developed for monitoring farmlands. UAVs enable real-time data access and analysis of crop growth and farmland dynamics information. Techniques such as the DDQN algorithm and GPI (Greedy Progressive Incremental) strategy are employed to optimize the positioning of UAVs and obtain innovative methods for data gathering. Additionally, Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) are employed in conjunction with IoT in agriculture. LSTM is used for data transmission, while CNN is used for building model networks. By leveraging IDS in these diverse applications, businesses can enhance their cybersecurity measures, protect critical sectors like agriculture, and develop effective strategies for threat detection, attribution, and mitigation (Fu et al., 2023).

In agricultural practices, protecting crops from animal interference poses a significant challenge. IoT devices such as Passive Infrared Sensors (PIR) are widely used for detection purposes. Additionally, Thin Field Transistors (TFT) can be used as visual alarms to protect farmlands when farmers are not present. The pixel control feature of TFT displays ensures clear imaging for easy monitoring. An Arduino board based on the Uno system can be developed to automatically detect intruders on farmland, providing a customizable and adaptable security solution (Sudarshan et al., 2019).

The mode and duration of irrigation in agricultural fields significantly impact crop production. Crop irrigation is pivotal in ensuring water supply and predicting yield values. Smart irrigation and precision farming techniques are viable solutions for optimizing expenses, forecasting, and enhancing efficiency in farming. In this context, the NSL KDD dataset is used in combination with the machine learning (ML) approach in IoT applications. Feature extraction techniques, such as Principal Component Analysis (PCA), are employed for performance analysis and comparison using different performance metrics (Raghuvanshi et al., 2022).

In the medical field, the Internet of Medical Things (IoMT) is utilized for health intelligence in smart hospitals to assist medical staff and improve patient care. However, implementing IoMT in a high-speed, real-time environment poses vulnerabilities and implementation challenges. Cybersecurity (CS) plays a crucial role in achieving a secure healthcare environment. IDS serves as a protective layer for communication within the

CS and network systems. Ensemble-based learning methods are employed in classifier models to predict intrusion attacks on medical networks, ensuring the security of sensitive healthcare data. IoT applications offer valuable solutions in various domains, including agriculture and healthcare, by leveraging technologies to enhance productivity, efficiency, and security. IDS plays a vital role in ensuring the protection and integrity of IoT systems and networks (Mokbal et al., 2022).

IoT has found extensive application in the development of smart cities, smart hospitals, and smart factories. These applications rely on real-time data to improve existing procedures and enhance capabilities. Ensuring information protection and detecting attacks are critical aspects of maintaining the security of smart infrastructure. An example of such an attack is the rank attack, where an intruder node attempts to attract legitimate traffic and steal data, such as patient information. Anomaly-based detection systems are commonly used to detect and mitigate such attacks in IoT environments. Simulations using tools like Contiki Cooja have shown higher accuracy rates and lower false positive rates for these systems (Said et al., 2020).

In critical IoT domains such as the defense industry and healthcare, false alarm rates and disruptions to emergency services can have significantly adverse effects. Intrusions in IoT networks can impact the accuracy of defect detection. To address these challenges, alert systems can be deployed to notify stakeholders in case of intrusion or attack detection. Anomaly Detection Systems (ADS) are often proposed for implementation in smart hospitals as part of IoT systems to accurately detect events of interest and adapt patient care environments accordingly. By processing data close to the data source, low latency rates can be ensured. Evaluations of ADS implementations in E-health event detection, based on realistic data analysis, have demonstrated higher accuracy rates in detecting both E-related events and intrusions in IoT. Overall, using IoT and IDS technologies in various sectors, including smart cities, healthcare, and critical IoT, enables enhanced capabilities, real-time monitoring, and efficient detection and mitigation of security threats. These advancements contribute to developing more secure and intelligent environments (Said et al., 2021).

IoT has brought significant advancements to healthcare, offering transformative technical, social, and economic changes, leading to a healthier future. Many medical devices now incorporate wireless communication capabilities, enabling remote monitoring and connectivity to the internet. Real-time monitoring and improved diagnostic accuracy contribute to effective patient treatment. However, an attack on the

Internet of Medical Things (IoMT) can pose physical harm and jeopardize patient safety. A mobile agent-based IDS is employed to secure the hierarchical network, which utilizes Regression and machine learning algorithms. This IDS detects intrusions at different network levels, achieving high accuracy rates while minimizing resource consumption (Thamilarasu et al., 2020).

Data security is paramount in healthcare monitoring systems, especially when patient data is transmitted over wireless devices such as optical or radio channels and fiber-based transport networks. IDS methods are commonly used to encrypt patient data using blockchain technology. The primary objective is to safeguard patient records and enhance healthcare information security. Wireless networks, including WAN, LAN, and WPAN, form the basis of these systems (Mishra et al., 2019).

Intelligent Transportation Systems (ITS) have rapidly developed as a novel application of IoT, particularly in the transportation sector for developing smart, sustainable cities. Emerging technologies like vehicular ad hoc networks (VANETs) have a big influence on how cities manage traffic and how safe roads are. They facilitate efficient data exchange among vehicles, enhancing security and passenger/driver safety. By integrating VANETs with uncrewed aerial vehicles (UAVs), where UAVs act as monitoring systems and assist vehicles in extending network connectivity, obstacles can be efficiently avoided, resulting in higher data delivery rates. Deep learning and advanced machine learning techniques are used to protect VANET and UAV communication from cyber-attacks, ensuring the integrity and confidentiality of vehicular data and providing enhanced security for ITS.

In conclusion, using IoT and IDS technologies in healthcare, data security, and intelligent transportation systems can revolutionize these fields, improving patient care and enhancing safety and transportation efficiency (Ahmad et al.,2021).

VANET is one of the emerging areas that continues to gain interest due to the increased diversity among available applications. It is an underlying system in Intelligent Transportation Systems (ITS), where its applications provide enhanced security and comfort for drivers and passengers. However, the characteristics and size of VANET pose challenges in terms of security. Cryptography technology is employed in this study to address these challenges, including Sybil attacks, DDoS, and black holes Gonçalves et al., 2019).

VANET is a subsystem of published ITS, enabling Automobiles to interact via wireless infrastructure. VANET has several uses, including enhancing driving safety and

preventing collisions. Machine Learning (ML) methods are used in VANET to improve security levels. Extensive research is based on the NSL-KDD dataset. In this study, a realistic dataset called ToN-IoT is employed, which is produced from a massive, diverse Internet of Things network. The suggested study utilizes the chi2 approach for feature selection and class balance, and the synthetic minority oversampling technique (SMOTE). The XGBoost algorithm therefore performed better than previous ML methods (Gad et al., 2021).

Shortly, the transitional period will be crucial in reshaping the industry, evolving beyond recognition. The emergence of intelligent vehicles has significantly contributed to availability of vehicular cloud services in smart cities, which is becoming increasingly important due to subscriber demands for improved vehicular management and architecture. Automated, secure, and continuous cloud services are utilized to meet he standards for quality of service and quality of experience. By grouping smart cars into service-specific clusters, which facilitate contact with dependable third parties that serve as intermediaries between service demands and suppliers, continuous service availability is made possible. Intrusion Detection Systems (IDS) are implemented employing three-phase data for categorization, reduction, and traffic analysis techniques to identify genuine and trusted service requests while guarding against false requests resulting from intrusion attacks. This suggested study employs Deep Learning (DL), Decision Trees (DT), and Machine Learning (ML)

Approaches for data reduction and false identification, enhancing the solution's effectiveness with satisfactory detection accuracy rates and higher false favorable detection rates (Aloqaily et al., 2019). Both privacy and trust are crucial values in the transportation field. Data security plays a significant role in automation systems as vehicle user data is transmitted over the internet and wireless connections like fiber and radio channels. In IoT-based environments and Intelligent Transportation Systems (ITS), Blockchain technology has found numerous applications to ensure data security (Mohan Krishna et al., 2020).

The emergence of Information and Communication Technology (ICT) has enabled target users to access electronic devices through open channels like the Internet. However, this digital communication opportunity also poses security risks that need to be addressed to safeguard privacy and integrity. To resolve these issues, a single-window-based cloud service delivery model utilizes smart cards as a single interface for accessing multifaceted electronic sectors such as banking, employment, and healthcare. Cloud

banking transitions are widely employed to enhance the security of intrusion detection systems in these fields (Ahmad et al., 2021).

2.3. Machine learning based IDS in IOT

Internet of Things (IoT) security is gaining increasing both the academic and industrial groups have shown interest. Therefore, the suggested paper has employed a automated security framework based on machine learning enhances security features in the IoT domain. This framework influences both Software-Defined Networking (SDN) and Network Function Virtualization (NFV) to mitigate various threats. The proposed AI framework combines a monitoring agent and an ML-based reaction agent, employing ML models divided into network pattern analysis and various anomaly-based Intrusion Detection Systems (IDS) in IoT systems. The proposed study demonstrates that utilizing a data mining approach to distribute attacks is extremely effective in identifying attacks that function well and are inexpensive the suggested paper employed a one-class Support Vector Machine (SVM) model. Experimental results showed an accuracy of approximately 99.71%. Lastly, a feasibility research was done to determine the current possible clarifications and endorse further research toward addressing open challenges (Bagaa et al., 2020).

The internet has integrated seamlessly into everyday life, with the rising amount of internet-connected gadgets rapidly. Socially, IoT gadgets have developed commonplace in daily life. However, difficulties are arising, and responses to these issues not clearly defined. Hence, the suggested paper adopts an improved IoT network using Machine Learning (ML) techniques. The proposed study utilizes ML and Deep Learning (DL) techniques, employing a standard dataset to enhance IoT security performance. Several ML methods like Support Vector Machine (SVM), Random Forest Multilayer Perceptron, and Convolutional Neural Network (CNN), were employed in the suggested paper. However, from the experimental results, it was found that CNN provided better results than the ML algorithms. A larger batch size might expedite the calculation process, demonstrating that CNN performed better and more efficiently than ML algorithms (Susilo et al., 2020).

Applications of the Internet of things offer various advantages and real-life applications that make life easier and more accessible. One of the most catastrophic attacks against IoT is Denial-of-Service (DoS). Therefore, the suggested paper used a Machine Learning (ML) classification algorithm to secure IoT against numerous DoS

attacks. The proposed study utilizes anomaly-based Intrusion Detection Systems (IDS). The datasets used in the suggested paper include CIDDS-001, UNSW-NB15, and NSK-KDD datasets. CIDDS-001 and UNSW-NB15 datasets were used as they contain real data traffic, which assists in generating precise IDS for monitoring and detecting new types of DoS attacks. The NSK-KDD dataset was used for classifier validation. ML classifiers used in the suggested paper are categorized as ensemble classifiers and single classifiers. Ensemble classifiers include Random Forest (RF), Adaboost, Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGB), and Extremely Randomized Trees (ETC). In contrast, single classifiers include Classification and Regression Tree (CART) and Multilayer Perceptron (MLP). The experimental results identified that the Classification and Regression Tree and XGB classifier provided the best trade-off regarding performance metrics and response time. The performance metrics used in the suggested paper were accuracy, specificity, sensitivity, False Rejection Probability (FRP), and Area Under the Curve (AUC) (Verma et al., 2020).

The Data flow and high dimensionality have significantly increased inside the IoT ecosystem. IDSs are crucial self-defense weapons against various cyber-attacks. However, the functional and physical variety of IoT IDS systems presents difficulties, making them more complex and unrealistic. Therefore, the proposed paper employs feature selection, feature extraction, and a gain percentage approach to choose and take out appropriate characteristics with different ratios. Ensemble-based and single classifiers are made use of to improve performance of the anomaly-based IDS model. These classifiers employ a hybrid feature selection approach to produce a final set of selected features. Ensemble classifiers include Artificial Neural Networks (ANN), k-nearest Neighbors (KNN), C4.5, and Bagging. The datasets used in the suggested paper are IoT-IDS2020 and NSL-KDD datasets. Finally, performance metrics such as accuracy, ROC area, precision rate, recall rate, and F1 Score are implemented in the suggested paper (Albulayhi et al., 2022).

A significant number of network security breaches occur in IoT networks, which can render current Network Intrusion Detection Systems (NIDS) unreliable. Therefore, the suggested paper employs six models: DFF, CNN, RNN, LR, NB, and DT. Three feature extraction algorithms are used: PCA, autoencoder, and Linear Discriminant Analysis (LDA). The datasets implemented in the suggested paper are ToN-IoT, UNSW-NB15, and CSE-CIC-IDS2018. The optimal number of features for each dataset has been identified, revealing that LDA degrades the performance of the ML models on two

datasets. Variance analysis is also used to analyze the performance, specifically the correlation between the number of dimensions and detection accuracy. It was found from the experimental findings that the combination of feature extraction and ML algorithms performed well across a wide variety of datasets and has the potential to be applied in application-oriented scenarios (Sarhan et al., 2022).

The internet has revolutionized communication by connecting the entire world and enabling information sharing on a single platform. Since data is considered one of the most valuable assets in every organization, people must protect their valuable assets by implementing necessary security measures, such as firewalls and antivirus software. Despite various security mechanisms in place, hackers can still exploit vulnerabilities. Therefore, the suggested paper emphasizes using Intrusion Detection Systems (IDS) in organizations to detect malicious activity and mitigate cyber-attacks. Several ML algorithms, including KNN, MLP, NB, SVM, and DT, are employed in the proposed study to classify network connections as normal or malicious. The effectiveness of these algorithms is evaluated using performance measures include sensitivity, F-score, and the accuracy and precision metrics. From the experimental results, it was found that the decision tree algorithm performed better and delivered superior results compared to the existing classifiers (Manhas et al., 2021).

Investing in reliable security devices is important as digital transformation faces remarkable security challenges. The growing number of cyber-attacks targeting IoT systems highlights the need for dependable and consistent detection of malicious network activity. Therefore, the suggested paper used supervised, unsupervised, and reinforcement learning approaches on nine malware captures. The dataset implemented in the study is the IoT-23 dataset, which considers both binary classification and multi-classification scenarios. The ML models employed in the suggested paper include SVM, XGBoost, Light GBM (Light Gradient Boosting Machine), iForest (Isolation Forest), LOF (Local Outlier Forest), and DRL (Deep Reinforcement Learning). DRL is based on the DDQN model, which was adapted to the Intrusion Detection (ID) context. The experimental results showed that Light GBM delivered more reliable and satisfactory results than the existing models, while the iForest classifier showed good anomaly detection performance. In conclusion, the proposed paper highlights that the techniques used in the study are well-suited for IoT intrusion detection (Vitorino et al., 2021).

In cybersecurity, IoT plays an important role in protecting the privacy and data of individuals and organizations. As a result of the growing dependence on computerized

services and the inadequacy of general-purpose IDS for handling persistent network threats, several lightweight protocols, including the MQTT protocol, have been implemented for IoT device communication. However, existing datasets do not include the MQTT protocol, resulting in a lack of IDS results for this protocol. The suggested study employs six different ML techniques to detect and identify MQTT-based attacks. Three abstraction levels of features, namely uni- and bidirectional flow and packet-based flow, are assessed and analyzed. The MQTT-based dataset is used in the suggested paper, and after preprocessing, the data is split into 75% for training and 25% for testing. Six ML techniques are employed for classification, and performance metrics such as precision, recall, and F1 score are used to evaluate the efficacy of the suggested models. The results demonstrate that flow-based features are particularly suitable for differentiating between MQTT-based attacks and benign traffic, as both exhibit similar characteristics (Hindy et al., 2020).

IoT devices are known for their connectivity and ability to generate and consume large amounts of data, facilitating communication between different devices. As IoT devices are typically low-power devices with limited computational capabilities, Network Intrusion Detection Systems (NIDS) are implemented to detect and eliminate malicious packets from these devices. The suggested paper focuses on employing MQTT, TCP, and cluster-based flow features using the UNSW-NB15 dataset. The article addresses issues such as overfitting and dataset imbalance. Several supervised ML algorithms, including Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN), were implemented on the clusters. The experimental results showed that RF achieved 98.9% accuracy, outperforming existing algorithms. The proposed paper also demonstrated that feature clustering improved accuracy and reduced running time compared to existing ML-based techniques (Ahmad et al., 2021).

In recent years, IoT devices have evolved rapidly, and their usage has significantly increased to enhance convenience in our lives. However, these devices are susceptible to various vulnerabilities, making them common attack targets. Therefore, the suggested study explores different ML techniques that can be applied to identify defects and attacks in IoT devices. The ML algorithms employed in the paper are Random Forest (RF) and XGBoost (XGB). RF is used for feature selection, while XGB is implemented to detect various types of attacks in the IoT environment. The dataset used in the study is the N-Balo T dataset, which contains hazardous botnet attacks. The experimental results showed that the proposed XGBoost-Random Forest model detected 99.94% of the attacks in the

dataset. Performance metrics such as sensitivity, F1 score, specificity, and balanced accuracy were used to assess the suggested model's performance. A comparative analysis between the existing and proposed approaches revealed that the XGB-RF model outperformed the existing models, achieving the highest sensitivity, specificity, F1 score, and accuracy (Faysal et al., 2022).

Due to its capacity to function in various contexts, IoT has considerably increased in relevance. However, these devices are vulnerable to security threats due to the lack of robust security protocols. Therefore, the suggested study utilizes various classifier techniques from ML ensemble methods to detect and mitigate malicious traffic, thus preventing attacks on IoT networks. One of the classifiers employed in the study is GBM (Gradient Boosting Machine), which trains a binary classifier that recognizes patterns in pre-processed recorded data packets anomalies and prevent network attacks. The dataset used in the study is the CSER-CIC-IDS2018 dataset, which offers extensive coverage and a high ratio of benign to attack traffic. The proposed model achieved an accuracy rate of 98.27%, outperforming existing methods (Verma et al., 2021).

The suggested paper focuses on analyzing botnet traffic in an IoT environment using three ML classifiers: SVM, RF, and LR. The dataset is divided into ten attacks carried out by two botnets (safety and miari) and involves nine different IoT devices. The performance of the classifiers is evaluated using metrics such as accuracy, recall, precision, and F1 score. From the experimental results, it was observed that the RF algorithm performed better than the others, while SVM showed the lowest performance. The high F1 score demonstrates the robustness of the three existing algorithms (Bagui et al., 2021).

Detecting and identifying irregular and malicious traffic in an IoT network is crucial for IoT security. The suggested paper addresses this by employing ML algorithms for intrusion detection. The datasets used in the study are NSL-KDD, UNSW-NB15, and CCIDS2017. The data collection and feature extraction process is described, involving steps such as raw packet input, preprocessing, feature set division, and entropy estimation. XGBoost classifier is then employed to distinguish between attacks and normal traffic. The performance of the proposed model is evaluated using metrics such as accuracy, precision, recall, and F-measure. The experimental results show that the UNSW-NB15 dataset outperforms the NSL-KDD and CCIDS2017 datasets (Diwan et al., 2021).

IoT represents the interconnectedness of devices that constantly share and analyze data to make decisions. However, this architecture also introduces potential failures and

attacks on the IoT system. Therefore, developing a device capable of automatically detecting defects and attacks in IoT networks is crucial. The suggested study employs an IDS using the BoT-IoT dataset and focuses on ML algorithms to enhance IoT security. ML algorithms such as KNN, LR, SVM, MLP, DT, and RF are employed. The efficiency of these algorithms is evaluated using performance metrics such as RoC, accuracy, precision, recall, and F1 score (Tyagi et al., 2021).

With the expansion of IoT in healthcare, telecommunications, and industries, security has become a critical aspect of technology. IDS has been introduced in various fields to improve the security of IoT devices and environments. The objective of the IDS in the suggested paper is to detect intrusions in real time and make reliable decisions. The proposed study employs hybrid IDS for edge-based IoT security, specifically the PK-IDS model, which combines snort IDS-based malicious detection and anomaly-based detection using the KNN classifier and PCA technique. ML classifiers utilized in the study are the KNN classifier and PCA technique. KNN enhances detection accuracy and leads to effective decisions, while PCA is employed for feature engineering and training processes. The steps involved in the suggested paper include misuse detection, pre-processing and normalization, feature engineering, training and validation, and classification. The datasets used in the study are NSL-KDD and BoT-IoT. The performance and effectiveness of the model are evaluated using metrics such as ACC, DR, and FAR. The experimental results show that the combination of these two datasets provides incredible and reliable results compared to other models (Guezzaz et al., 2022).

As IoT becomes increasingly prominent, the number of security breaches associated with IoT devices also rises. Therefore, employing IDS techniques to mitigate attacks exploiting IoT vulnerabilities is crucial. However, due to the limitations of IoT devices and protocols, traditional IDS may not work effectively in specific IoT environments. The suggested study addresses this issue by employing ML algorithms to detect cyber-attacks and anomalies. The datasets used in the study are CICIDS2017 and NSL-KDD. The study applies multicollinearity, sampling, and dimensionality reduction standards to achieve significant intrusion detection. Classifiers employed in the study include KNN, RF, and XGBoost. The experimental results demonstrate high detection rates and low false alarm rates with the employed datasets. The proposed model is also shown to be efficient and suitable for deployment on IoT nodes with restricted power and storage capabilities (Roy et al., 2022).

IoT integrates billions of self-organized and heterogeneous nodes that communicate with each other without human intervention. It has become a major technological advancement, finding applications in various fields such as healthcare, agriculture, and education. However, node heterogeneity poses complex challenges in IoT environments. The suggested paper employs various ML techniques, including RF, SVM, and DL techniques, such as DNN, LSTM, Bi-LSTM, and DBN, in the IoT environment. The datasets used in the study are KSL-KDD, IoTDevNet, DS2OS, IoTID20, and IoT botnet datasets. The process in the paper involves data collection, pre-processing (cleaning, visualization, numericalization, and normalization), training and testing (80% training, 20% testing), loading values into learning algorithms (DT, RF, DBN, SVM, LSTM, DNN), evaluating the detection rate, comparing the suggested model with existing models, and selecting an optimal model. From the experimental results, it is observed that Bi-LSTM outperforms the existing classifiers (Islam et al., 2021).

IoT devices are susceptible to numerous attacks due to the proliferation of small computing devices. A robust IDS should be developed to enhance the security of IoT devices. The suggested paper employs various ML techniques, including feature selection methods and feature classification methods. The datasets used in the study include the UNSW-NB15, KDD-CUP 1999, and NSL-KDD datasets. The process involves data collection, training, and testing; pre-processing with data transformation and normalization techniques; feature selection using a filter, wrapper, and embedded methods; and classification using LR, KNN, DT, SVM, RF, and MLP models. TP, TN, FP, and FN rates evaluate the model's performance. The experimental results indicate that KNN and DT outperform existing classifiers (Fenanir et al., 2019).

With the rapid development of IoT devices, the number of cyber-attacks has also increased. The suggested paper focuses on Botnet-based attacks, which are prevalent in IoT environments. Conventional security methods may not effectively mitigate these attacks. Hence, the paper proposes an ML-based botnet attack detection framework. The framework incorporates an efficient feature selection method to develop a lightweight detection system with promising performance. The algorithms employed in the study are Naive Bayes, J48, DT, and ANN. The suggested model achieves a detection rate of around 99% for botnet attacks. The experimental findings show that the suggested model works better than current methods for identifying botnet assaults (Soe et al., 2020).

2.4 Deep learning based IDS in IoT

The exponential growth of IoT has attracted significant attention from cybercriminals, posing a major security threat to IoT devices and protecting identity and information. To mitigate these risks, reliable security measures, such as IDS, safeguard against unauthorized access. The suggested paper focuses on an IoT-based IDS system for IoT devices. The proposed model uses a four-layer Fully Connected (FC) network architecture to detect and identify various attacks, including DoS, black hole, opportunistic service, and wormhole attacks. The performance of the IDS system is evaluated using metrics such as precision, recall, and F1 score. The experimental results show that the DL-based IDS achieves an average accuracy of 93.21%, providing satisfactory results for enhancing the security of IoT networks (Awajan, 2023).

In the era of communication and IT, the vast amount of data traffic generated by IoT enables the analysis of abnormal network attacks. DL techniques have been widely applied in IDS models to detect unusual access in the network. However, these DL models have certain limitations that can be overcome by the suggested model, which enhances network security and complexity. The presented study introduces the DL-based RK-CNN-MMBO model for detecting network attacks. The steps in the proposed research include data pre-processing with min-max normalization and using the EBRO (Enhanced Battle Royal Optimization) algorithm for optimal feature detection. The selected features are then categorized using DL classifiers (RK-CNN), and the performance of the classifiers is enhanced using the MMBO classifiers to detect attacks precisely. The N-BaIoT and CICIDS2017 datasets are utilized in the study. The N-BaIoT dataset achieves an accuracy rate of 99.959%, while the CICIDS2017 dataset, when employed with the RKCNN-MMBO model, achieves an accuracy rate of 99.94%. Comparing the proposed model to existing approaches, the experimental findings show that it detects incursions with high accuracy (Om Kumar et al., 2023).

According to the suggested paper, IoT devices are extensively used in various sectors, such as smart homes, healthcare centers, and smart cities. However, ensuring the security of these devices is critical as protecting information becomes increasingly important. To address this issue, the suggested study proposes a deep learning-based approach for detecting threats and attacks in the IoT environment. The motivation behind this model is the lack of optimal feature learning and dataset management in existing models, which the proposed model aims to improve upon. The algorithms employed in building the model include SMO (Spider Monkey Optimization) for feature selection and

SDPN (Stacked Deep Polynomial Network) for classification, distinguishing between normal and abnormal data. The model identifies various types of attacks, such as DoS attacks, U2R attacks, and R2L attacks. The performance of the suggested model is evaluated using accuracy, recall, precision, and F1 score as performance metrics. The proposed DL-IDS model performs better than existing models (Otoum et al., 2022).

As the prevalence of cyber-attacks continues to rise, detecting and mitigating network defects and attacks becomes crucial. Undetected attacks can cause significant problems and disrupt critical systems for end users. In this context, the suggested study employs deep learning techniques to detect attacks in the network. The chosen deep learning algorithm in the paper is DNN (Deep Neural Network) due to its ability to identify abnormal behavior in IoT networks. Implementing the existing model comprises three stages: data collection, data pre-processing, and model performance evaluation. The study discusses various types of attacks, such as black hole attacks, DDoS attacks, and wormhole attacks. The performance of the existing models is evaluated using precision, recall, F1 score, and accuracy as performance metrics. The experimental results demonstrate that the proposed model achieves better overall detection performance compared to existing models (Otoum et al., 2022).

The development of IoT created a revolution for cybercriminals. Due to this, the security sector has observed an exponential rise in cyberattacks, which result in various ways for intruders to enter the network and steal information. Hence, the suggested paper employed anomaly-based IDS, which employed CNN methods to protect the network from unauthorized access. The suggested study used 4 CNN blocks such as CNN1D, CNN2D, and CNN3D. These CNN blocks comprise a convolutional layer, an average pooling layer. Spatial dropout layer and layer normalization. The entire model contained four layers: the input layer, FCL, Output layer, and Convolutional layer. The dataset employed in the suggested study includes BoT-Iot datasets, IoT intrusion detection datasets, MQTT-IoT-IDS 2020, and IoT-23 datasets. The steps involved in building an improved model are- dataset collection, feature processing, pre-processing dataset, feature selection, and evaluating the existing models' performance. the detection rate obtained by CNN1D model includes 99.74%, CNN2D model 99.43%, CNN3D model 99.029% (Ullah et al., 2021).

IoT has developed to improve and enhance people's living standards by delivering various devices and applications in different domains. However, theft of information is considered to be expected in IoT environments. High-end security providers such as IDS

were employed to protect the IoT devices. Therefore, the suggested paper employed deep learning, which assists in identifying the threats possess the network. The proposed model provided a seamless choice for anomaly-based detection. The presented article used a CNN-based technique for anomaly-based IDS, which takes advantage of IoT power. The suggested model provided the quality to investigate the traffic effectively and efficiently across IoT. The proposed model has the capability and ability to find any possible intrusions and abnormal traffic behavior. The experimental results concluded that the NID dataset attained an accuracy of 99.51%, whereas the BoT-IoT dataset attained an accuracy of 92.85% (Saba et al., 2022).

IoT devices paved the way for various smart devices and produced various benefits for people. However, IoT is a promising target for criminals. Hence, the suggested study employed (DL) methods to provide additional security and immensely detect emerging and unknown attacks. Different DL models used in the proposed study include RNN, CNN-RNN hybrid, Boltzmann, DBN, and GAN models. Datasets employed in the presented paper include KDD99, NSL-KDD, ICSX-2012, UNSW-Nb15, CIDDS-001, and finally CICIDS-2017 dataset. The performance of these models was estimated using performance metrics, including the models' accuracy. The suggested paper revealed that DL methods are much more preferred and ideal for IDS than other shallow models, producing various challenges (Tsimenidis et al., 2022).

The IoT industry has experienced significant growth in recent years, but IoT devices remain highly vulnerable to cyber-attacks due to their small size and heterogeneity. To enhance the security of IoT networks, the suggested paper proposes using an IDS (Intrusion Detection System) device developed using deep learning methods. This IDS device focuses on detecting IoT DDoS botnet attacks in the network. The BoT-IoT dataset is employed for the study, as it includes IoT-generated traffic and incorporates new features specific to IoT. The deep learning models used in the proposed research include decision trees (DT), C4.5, ARM, Naive Bayes (NB), and artificial neural networks (ANN). The experimental results show that the decision tree classifier outperforms the others with an accuracy rate of 93%. Performance metrics such as accuracy, recall, precision, and F1 score are used to evaluate the IDS model's performance (Shareena et al., 2021).

While IoT devices have improved people's lives, they are also susceptible to cyber risks. Traditional ID technologies are often insufficient for meeting the network threat detection requirements in IoT environments. Therefore, the suggested study proposes a

near-end optimization strategy for industrial IoT IDS using deep learning algorithms, explicitly combining Deep Reinforcement Learning (DRL) and IDS. The IDS model in the proposed study utilizes the LightGBM algorithm for effective feature selection and employs the Proximal Policy Optimization (PPO2) algorithm and Rectified Linear Unit (ReLU) activation function for the network structure. The flow of the study involves dataset preprocessing, employing the IDS agent, training the agent module using the deep learning network, and finally deploying the IDS model using the PPO2 algorithm and ReLU activation function. The proposed IDS model achieves an accuracy rate of 99% in tests conducted using publicly accessible datasets. A comparative analysis is performed against existing algorithms such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Q-Network (DQN), with the suggested algorithm demonstrating superior performance (Tharewal et al., 2022).

In the context of smart cities and their reliance on IoT technologies, deploying IoT devices introduces posing security risks and implementing smart initiatives susceptible to attacks. The suggested study focuses on employing IDS (Intrusion Detection System) techniques to enhance security in smart cities. Conventional IDS methods may struggle to achieve high detection accuracy, so the proposed research introduces a hybrid optimization and deep learning-based IDS approach specifically designed for smart cities. The NSL-KDD dataset is utilized for the study. The dataset undergoes preprocessing to obtain an effective and precise IDS, and then the Hybrid Chicken Swarm Genetic Algorithm (HCSGA) is employed for feature selection. Additionally, the Min K-means algorithm is used for clustering. Finally, the data is classified and categorized using the DLHNN (Deep Learning-based Hybrid Neural Network) classifier. Experimental results demonstrate that the proposed IDS model outperforms existing models in accuracy and efficiency (Gupta et al., 2022).

IoT devices are growing in number and provide various benefits, but they also face security issues and are vulnerable to different attacks. The suggested paper proposes a hybrid convolutional neural network (HCNN) model for intrusion detection in IoT environments to address this. The HCNN model combines LSTM (Long Short-Term Memory) and RNN (Recurrent Neural Network) architectures to achieve enhanced performance. The proposed model consists of four stages: data collection, data preprocessing, network training, and attack detection. The data preprocessing stage involves feature extraction, feature encoding, and the creation of a feature matrix. The

data is then trained using the CNN (Convolutional Neural Network) architecture with weight values set accordingly. Finally, intrusion detection is performed using the CNN process. Performance evaluation of the suggested model includes various metrics such as precision, recall, F1 score, miscalculation rate, detection time, and accuracy rate. Experimental results demonstrate that the proposed HCNN model outperforms existing RNN models, and the overall results are deemed satisfactory. The study concludes that the HCNN model is well-suited for various IoT environments (Smys et al., 2020).

The suggested study aims to improve the robustness of IDS (Intrusion Detection System) in IoT networks, as existing studies have shown vulnerabilities and the potential for attacks on IDS. The proposed approach enhances IDS robustness by employing the Fast Gradient Sign Method (FGSM) to generate adversarial examples. The study utilizes three DL techniques: CNN, LSTM, and GRU models. Three training models are employed: training with normal examples, training with adversarial examples, and retraining with normal examples. The performance of the models is evaluated using performance metrics, with CNN identified as the most robust model (Rashid et al., 2022).

IoT has brought numerous benefits to daily life, but its vulnerabilities make it susceptible to cyber-attacks. To address this, the proposed study utilizes DL techniques for attack detection. The process includes feature extraction, feature pre-processing, training the dataset, and classifying the data using classifiers. The study employs the BoT-IoT dataset and implements a feed-forward neural network model. Performance evaluation metrics such as accuracy, F1 score, recall, and precision are used (Ge et al., 2021).

To improve the performance of DL models, the proposed study employs DL methods focusing on the GRU algorithm. The process involves pre-processing techniques, implementing the GRU classifier, using an MLP layer for decision-making, and applying the softmax activation function for output. The KDD99-DOS dataset is utilized for the study, and performance metrics, including precision, recall, F1 score, and accuracy, are used for evaluation (Zhong et al., 2021). Table 2.1 provides a summary of the key findings and results from the existing studies.

Table 2.1. Summary of the Existing studies

Sl. No	Objective	Summary	Reference
1	To combine monitoring agents and ML-based reaction agents, which employed ML Models divided into network pattern analysis and various anomaly-based IDS in IoT systems to detect the unknown attacks.	In order to select the assaults with the best performance and lowest cost, the distribution of attacks using a data mining technique is incredibly successful the suggested paper employed a class SVM model. From the experimental results, it was determined that the accuracy obtained was around 99.71%.	Bagaa et al., 2020).
2	To implement ML-based techniques to enhance the security performance of IoT devices	Several ML algorithms, such as random forest, SVM, Multilayer perceptron, and CNN, were employed in the suggested paper, and CNN and DL algorithm is also used in the proposed article. However, from the experimental results, it was identified that CNN provided better results than ML algorithms.	(Susilo et al., 2020).
3	To secure the information and data present in the network of IoT using ML classifiers	The datasets employed in the suggested paper were CIDDS-001, UNSW-NB15, and NSK-KDD datasets, in which CIDDS- 001 and UNSW-NB15 are used since the datasets contained traffic of accurate data, which assist in generating precise IDS for monitoring and detection of a new type of DoS. From various ML algorithms, the XGBoost classifier delivered the best results.	Verma et al., 2020).
4	To detect malicious attacks in the IoT networks by employing ML classifiers.	The suggested paper employed supervised, unsupervised, and reinforcement learning approaches on nine malware captures. ML models used in the presented paper include- SVM, XGBoost, LightGBM (Light Gradient Boosting Machine), iForest (Isolation Forest), LOF (Local Outlier Forest), and DRL (Deep Reinforcement Learning). DRL is based on the DDQN model, which was adapted to the ID context. The experimental results showed that LightGBM delivered more reliable and satisfactory results than the existing ones. Good anomaly detection results were displayed by the iForest classifier.	Vitorino et al., 2021).
5	To implement a lightweight protocol for communication of IoT devices.	The MQTT-based dataset is employed in the suggested paper. The dataset implemented in the presented paper is pre-processed; after pre-processing of the dataset, data is split into 75% of training and 25% of testing. 6 ML techniques were employed for classification purposes, and finally, performance metrics, precision, recall, and F1 score were used to find the suggested models' efficacy.	Hindy et al., 2020).
6	To detect any defects, attacks, or outbreaks in the network of IoT.	ML algorithms employed in the suggested paper were RF and XGBoost. RF was used for the feature section, and XGB was implemented to identify various kinds of attacks present in the IoT environment. The experimental results determined that the suggested model XGBoost and Random Forest detected 99.94% of the attacks in the dataset.	Faysal et al., 2022).

Sl. No	Objective	Summary	Reference
7	To discover dangerous threats in the IoT network using DL classifiers.	The suggested model helps expose IoT DDoS botnet attacks in the network. DL models employed in the proposed study include DT, C4.5, ARM, NB, and ANN. From the experimental results, it was identified that the Decision tree performed better than other classifiers.	(Shareena et al., 2021).
8	To improve the robustness of the IDS by DL classifier such as FGSM (Fast Gradient Sign Method) classifier.	FGSM (Fast Gradient Sign Method) helps generate adversarial examples—the suggested model employed 3 DL techniques: CNN, LSTM, and GRU. From the experimental results, it was detected that CNN is the most robust and vigorous one.	(Rashid et al., 2022).

Based on the review of existing studies, several future directions and areas of improvement can be identified:

1. Increasing profiling accuracy and collecting suitable features: Future work can focus on improving the accuracy of profiling patterns adopted by malicious traffic in IoT networks. This can be achieved by collecting relevant parts related to IoT protocols and employing the proposed model to identify the accuracy of known and unknown attacks. The analysis and examination of various datasets using these techniques will be valuable (Ahmad et al., 2021).

2. Higher accuracy and reduced detection time: Future work should aim to maintain high accuracy while reducing the detection time in busy IoT systems. Implementing high-performance ML classifiers for detecting unknown attacks in the IoT environment will be crucial to focus on (Faysal et al., 2022).

3. Detection of location-dependent attacks: Future research can explore detecting attacks specific to IoT devices based on location. This includes identifying cloning devices, spoofing, Sybil attacks, isolation attacks, misappropriation attacks, etc. Tracking device IDs and authenticating journal entries using the DODAG (Destination Oriented Directed Acyclic Graph) table can be explored to detect these attacks (Thamilarasu et al., 2019).

4. Efficient detection of known and unknown botnet attacks: Future work can involve analyzing normal traffic data from various types of upcoming IoT devices. The goal would be to enhance the efficiency and effectiveness of detecting known and unknown botnet attacks in real-time IoT environments (Shareena et al., 2021).

5. Integration of feature selection methods with ML algorithms: Future research can explore integrating different feature selection methods with various ML algorithms

specifically designed for real-time IoT devices. This integration can enhance the performance and effectiveness of IoT security solutions (Fenanir et al., 2019).

These future directions aim to address the challenges and improve the security capabilities of IoT networks, considering factors such as accuracy, detection time, location-based attacks, and botnet attack detection.

3. METHODOLOGY

3.1. Preface

Detecting anomalies in network traffic is crucial for maintaining the security of IoT devices and networks. The proposed framework you mentioned addresses this by implementing a practical model that can detect and categorize the presence or absence of attacks in network traffic. To achieve this, the framework utilizes the UNSW NB 15 dataset, commonly used for evaluating intrusion detection systems. In the proposed methodology, rectified linear weights are used. These weights feed the best weights, as opposed to standard weights, into a combination of the Bidirectional Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN) model. Using this model, the framework can efficiently and effectively detect attacks in the network traffic. The Bi-LSTM component can capture temporal dependencies and patterns in the traffic data, while the CNN component helps extract relevant features from the data. Rectified linear weights enhance the model's performance by providing improved weight initialization. Overall, the proposed framework aims to enhance the accuracy and effectiveness of anomaly detection in network traffic by utilizing a combination of Bi-LSTM, CNN and rectified linear weights.

3.2. Proposed design

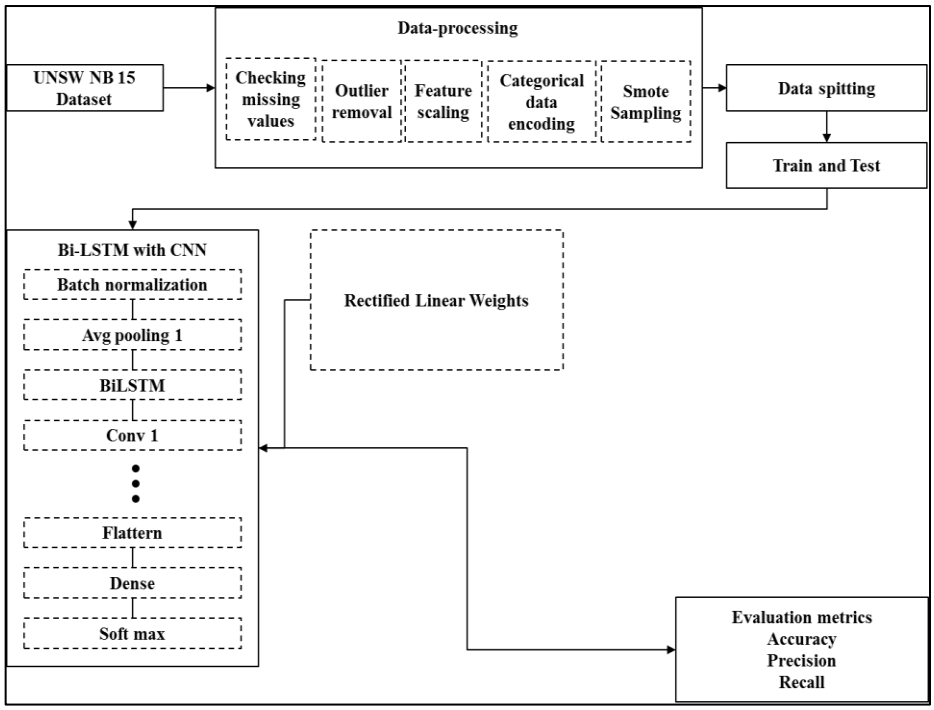

Figure 3.1. Overall proposed framework

Figure 3.1 illustrates the overall methodology employed in the proposed project, which utilizes the UNSW-NB 15 dataset. The steps involved in implementing the proposed model are outlined in the figure.

The first step is data preprocessing, which prepares the dataset before training or learning the model. This involves checking missing values, scaling features to specific dimensions, and encoding categorical data. These preprocessing steps aim to ensure the best outcomes in the subsequent stages.

Subsequently, the dataset is split into training and testing subsets. This division evaluates the model's performance on unseen data and helps assess its generalization capabilities.

The proposed study employs a hybrid model that combines Bi-LSTM with CNN network architecture. This hybridization addresses challenges such as the exploding gradient problem, significant feature extraction, noise reduction, and accelerated learning. Bi-LSTM, with its recurrent nature, can capture temporal dependencies and preserve learned information over time. On the other hand, CNN is effective in extracting complex and critical feature representations from preprocessed data.

In addition to Bi-LSTM and CNN, a rectified linear weights framework based on the RNN (Recurrent Neural Network) is integrated into the model. This framework compresses the trained data features using the encoder and performs reconstruction of the original data using the decoder. The rectified linear weights process aims to enhance data prediction and make it more secure. The variational method is employed to estimate the parameters in the RNN.

The proposed model aims to attain low error rates and improve detection performance according to valuation metrics by integrating these components.

Finally, performance metrics are employed to evaluate the effectiveness of the proposed framework. These metrics assess the model's performance in detecting anomalies and provide insights into its accuracy and overall performance.

Overall, the proposed methodology involves data preprocessing, dataset splitting, the hybridization of Bi-LSTM with CNN, integrating a rectified linear weights framework, and performance evaluation using appropriate metrics. These steps aim to enhance the detection performance and accuracy of the proposed model for anomaly detection in IoT network traffic.

Among all the existing methods, CNN is combined with the Bi-LSTM model because the network packets are presented in a 1D format. Therefore, CNN, along with

the Bi-LSTM model, is employed. The main motive for utilizing the CNN-BiLSTM model is that this model aids in automatically extracting features from the raw data and facilitates better extraction of coarse-grained features at the start of the network. CNN and the max-pooling layers are employed together for parameter sharing and spatial invariance characteristics. Parameter sharing helps constrain the number of parameters in the initial layers, resulting in feature extraction with fewer computational resources, while spatial invariance enables recognition of correlations between features. ReLU is employed as the activation function for faster convergence.

Furthermore, batch normalization is applied between the intermediate layers to minimize the effects of covariance shifts between layers during the training process. This helps avoid slower learning speeds and makes weight fluctuations more robust in the network. The model's final layer is the FCL (Fully Connected Layer) model, which serves as the output layer with a softmax activation function for binary classification, aiding in identifying the presence of attacks in the network.

Significant Method: Integrating a Convolutional Neural Network (CNN) with the Bidirectional Long Short-Term Memory (Bi-LSTM) model has emerged as a promising approach to improving current techniques. This fusion arises from the intrinsic one-dimensional representation of network packets, making it well-suited for applying Convolutional Neural Networks (CNNs) in conjunction with the Bidirectional Long Short-Term Memory (Bi-LSTM) architecture. The primary justification for implementing the CNN-BiLSTM model lies in its ability to extract significant features from unprocessed data independently. Particularly noteworthy is its capacity to effectively remove macroscopic properties during the early stages of network processing, a characteristic that significantly enhances data representation.

Strategic utilization of parameter sharing is evident in the fusion of CNN and max-pooling layers. This approach not only maximizes the use of computational resources by imposing constraints on parameters in the initial layers but also imparts the crucial attribute of spatial invariance. The network's capability to preserve spatial invariance enables it to identify relationships between features, enhancing interpretability and analytical capabilities. The use of the Rectified Linear Unit (ReLU) activation function is strategically implemented to enhance the model's convergence rate and expedite the training process.

To bolster the model's robustness, the integration of batch normalization is seamlessly implemented within the intermediate layers. The astute inclusion discussed in

this statement addresses the issue of covariance shifts that occur across layers during the training process. By tackling this challenge, this inclusion aims to maintain a consistent learning rate and improve the resilience of weight modifications. The deployment of this strategic approach mitigates the occurrence of poor learning rates and promotes network stability.

After the model, the Fully Connected Layer (FCL) assumes a prominent role as the output layer, effectively connecting with a softmax activation function to facilitate binary classification. Utilizing this strategic design enables precise detection of network threats, a critical objective within the cybersecurity domain.

In conclusion, the CNN-BiLSTM model demonstrates a high level of sophistication by effectively using the underlying structure of network packets. This study proposes a novel methodology that combines feature extraction, parameter sharing, spatial invariance, and effective activation functions to detect and categorize network abnormalities. The proposed strategy aims to advance the current state-of-the-art anomaly detection methodologies.

3.3. Dataset

The UNSW-NB15 dataset https://research.unsw.edu.au/projects/unsw-nb15-dataset is in a prominent position as a highly recognized and extensively employed benchmark dataset within network intrusion detection and cybersecurity research. The primary objective of its creation was to assess and enhance intrusion detection systems, with a specific focus on those designed to detect network-based attacks. The dataset was produced inside a controlled setting, rendering it a significant asset for evaluating the efficacy of diverse machine learning and data mining methodologies in detecting and categorizing distinct network assaults and abnormalities.

The UNSW-NB15 dataset encompasses several significant characteristics and specificities:

1. The origin and source of the information can be traced back to its original context. The Cyber Range Lab at the University of New South Wales (UNSW) in Australia produced the dataset. The IXIA PerfectStorm program was utilized to produce the above output. This tool is a traffic generator capable of replicating authentic network traffic and controlled attacks.

2. The dataset known as UNSW-NB15 consists of network traffic data gathered from a range of attack instances and regular activities. This classification encompasses

nine distinct categories of attacks: Fuzzers, Analysis, Backdoors, DoS (Denial-of-Service), Exploits, Generic, Reconnaissance, Shellcode, and Worms. In addition, regular (non-malicious) network traffic is included in this dataset.

3. The dataset provides comprehensive details about individual network connections, including features such as source and destination IP addresses, source and destination ports, protocol, bytes transmitted, packets exchanged, and flags. The data is supplied in a format organized and designed for compatibility with machine learning analysis.

4. The UNSW-NB15 dataset is of considerable size and variability, encompassing a substantial number of network connections amounting to tens of thousands. This heterogeneity makes it suitable for assessing the resilience and generalizability of intrusion detection methods.

5. The dataset aims to replicate genuine network traffic and attacks, albeit being generated within a controlled setting. It encompasses both benign and malicious cases, providing an accurate portrayal of network behavior. Ground truth labels are assigned to each link, indicating whether it represents regular behavior or an attack.

6. The dataset has been extensively utilized in scholarly research and industry applications to assess the effectiveness of intrusion detection systems. Researchers use the dataset to evaluate the accuracy, precision, recall, and other evaluation metrics of various machine learning algorithms and methodologies.

7. The UNSW-NB15 dataset serves as a valuable resource for addressing many difficulties in network security, particularly in identifying and mitigating cyber threats and attacks. This facilitates the development of intrusion detection systems capable of efficiently classifying network traffic and distinguishing between authorized operations and malicious conduct.

In conclusion, it can be asserted that the UNSW-NB15 dataset holds significant value as a resource for scholars and professionals engaged in the domain of cybersecurity. This tool's broad and realistic composition and the presence of labeled data make it indispensable for assessing and enhancing intrusion detection approaches and models.

3.3.1. Data pre-processing

Data pre-processing is indeed a crucial step in data preparation, as it transforms raw data into a format that is understandable and suitable for analysis. Here are some key points highlighting the importance of implementing data pre-processing:

• Improved Accuracy and Reliability: Data pre-processing helps enhance the accuracy and reliability of the data by handling inconsistent and missing values. By removing or imputing missing data and addressing inconsistencies, the overall quality of the dataset improves, making it more reliable for analysis.

• Consistency of Data: During data collection, it is common to encounter duplicate or redundant data. These duplicates can introduce biases and inconsistencies in the analysis. Data pre-processing involves identifying and removing duplicate data, ensuring that the dataset remains consistent and free from redundant information.

• Enhanced Readability for Algorithms: Data pre-processing plays a vital role in maximizing the readability and interpretability of machine learning and deep learning algorithms. By transforming the data into a standardized format, normalizing or scaling features, and handling outliers, the algorithms can more effectively understand and interpret the data, leading to better performance and results.

3.3.1.1. Check missing values

Missing values are considered to be one of the most common phenomena in datasets. Missing data values can also be referred to as "Not Available" (NA). These missing data can result from errors in the data entry process, collection using incorrect methods, leaving particular fields blank at times, or when specific values are not applicable. Therefore, missing values should be handled and considered carefully, as ignoring them during dataset pre-processing can lead to incorrect results. Missing values can either be deleted or replaced with an appropriate statistic. Hence, properly detecting missing values is the first step in dealing with them.

Issues caused by missing values include:

• Missing data can lead to a lack of precision in statistical analysis.

• Bias can arise due to distortion in the data distribution.

• Computational complications occur when holes exist in the dataset.

There are various ways to handle the missing values:

• Removal or deletion of missing values

• Deleting / Eliminating Rows

• Implementation of Prediction model (regression and classification)

• Replacing the missing value with any of the statistical methods such as mean/median/mode.

• Imputation using DL library.

• By employing sklearn impute models such as the SimpleImputer, iterativeImputer, and KNNImputer models.

3.3.1.2. Removal of outliers

Removing outliers is considered a crucial process because these outliers can cause significant issues. They often represent measurement, processing, poor sampling, or data entry errors. These outliers can escalate the variability of the data, thereby reducing statistical power. Hence, eliminating outliers can lead to more statistically significant results.

The advantages of removing outliers include:

• Enhancing the stability and accuracy of statistical and machine learning models.

• Improving data visualization by providing more transparent and more precise data representations.

• Enhancing the robustness of statistical analysis, ensuring more reliable results.

```
"OUTLIERS REMOVAL"

# Using Z-Score Analysis
From Scipy import stats

# Calculate Z-score for each column
z_scores = np.abs(stats.zscore(X))

# Set the threshold for outlier detection
threshold = 3

# Identify outliers
outliers = np.where(z_scores > threshold)

# Remove outliers
clean_df = pd.DataFrame(X).drop(pd.DataFrame(X).index[outliers[0]])

clean_y = pd.DataFrame(y).drop(pd.DataFrame(y).index[outliers[0]])

# Print the number of outliers detected and the cleaned dataset
print("Number of outliers detected: ", len(outliers[0]))
```

```
Number of outliers detected: 133478
```

3.3.1.3. Categorical data encoding

Categorical data encoding is a process of converting categorical data into an integer format, which allows the data with transformed categorical values to be used in various approaches.

In certain cases, it is necessary to transform categorical variables into numbers to ensure the model understands and utilizes the relevant information.

Some advantages of encoding data are:

• It speeds up data entry, reducing the time required for inputting categorical data.

• It improves the accuracy of data entry by removing ambiguity associated with categorical representations.

• It enables faster data searching, as numerical representations can be indexed and sorted more efficiently. Categorical data encoding is a process of converting categorical data (attack and nonattack) into an integer format (0 and 1), which covers attack as (1,0) and nonattack as (0,1) in the separate columns that mention as (246636 rows, two columns).

```
#Encode labels from text to integers.
le = preprocessing.LabelEncoder()
y = le.fit_transform(y_resampled)
# convert labels to one-hot encoding
Y = to_categorical(y)
Y.shape
(246636, 2)
```

3.3.1.4. Feature scaling technique

Feature scaling is a method used to standardize the range of features or independent variables in data. It is also known as data normalization and is typically performed during the pre-processing step. Feature scaling steps help to normalize the independent features, ensuring they are in a consistent range. Some advantages of implementing feature scaling techniques include:

• Faster and more effective training of models through feature selection.

• Improved optimization process, as feature scaling helps algorithms converge more efficiently.

• Without feature scaling, algorithms may be biased towards features with higher Magnitude values, leading to imbalanced results. Feature scaling is the operation in which data are converted from 0 to 1 for improved computation and will be passed to the next step.

```
"FEATURE SCALING"
from sklearn.preprocessing import StandardScaler
## scaling
scaler = StandardScaler()
X_Scale = scaler.fit_transform(clean_df)
```

3.3.1.5. Smote sampling

SMOTE (Synthetic Minority Oversampling Technique) is a statistical technique used to effectively and equally increase the number of cases in a dataset, particularly in cases where the minority class is underrepresented. SMOTE works by generating new instances for existing minority cases.

One of the advantages of employing SMOTE is that it does not generate exact duplicate data points. Instead, it generates synthetic data points that are slightly different from the original data points. This helps introduce diversity into the dataset and can improve the robustness of the model when dealing with imbalanced classes. The class distribution before SMOTE for attacks is 123,318; for non-attacks, it is 63,483. After applying SMOTE sampling to attacks and non-attacks, the class distribution remains at 123,318.

```
Class distribution before SMOTE:
 1    123318
0     63483
dtype: int64
Class distribution after SMOTE:
 0    123318
1    123318
dtype: int64
```

3.4. CNN

CNN is a type of DL employed for image recognition and classification of images. It consists of several layers, including the convolutional layer, pooling layer, and FCL (Fully Connected Layer). The basic functions of these layers are as follows Liu et al., 2021):

Convolutional layer: This layer is responsible for extracting features from input images by applying filters. The filters scan the input image and detect patterns and features, such as edges, textures, or shapes.

Pooling layer: The pooling layer is used to downsample the image or feature map. It helps reduce the computational complexity by reducing the dimensionality of the data.

The most common pooling techniques are max pooling and average pooling, where a region's maximum or average value is selected as the representative value.

Fully Connected Layer (FCL): The fully connected layer is the final layer in a neural network. It takes the features learned from the previous layers and produces the final predictions. Each neuron in the FCL is connected to every neuron in the previous layer, allowing for complex interactions and transformations.

In general, a neural network is classified into three main layers:

1. Input layer: The input layer receives the input data for the model. It represents the dimensionality of the input vector, defining the number of input features.

2. Hidden layers: The hidden layers are the intermediate layers between the input and output layers. They perform computations and transformations on the input data to extract meaningful features and patterns. The number of hidden layers and their sizes depend on the complexity of the data and the model architecture. Neural networks can have multiple hidden layers.

3. Output layer: The output layer receives the transformed data from the hidden layers and produces the neural network's final output. It typically uses a logistic function, such as softmax or sigmoid, to convert the work for each class into a probability score representing the likelihood of belonging to that class. The output layer represents the final predictions of the neural network.

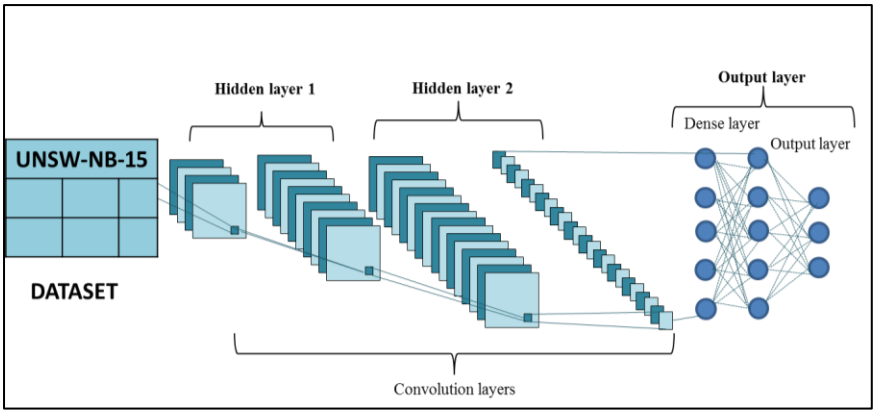

Figure 3.2. Architecture of CNN

Figure 3.2 shows the architecture involved in CNN. Hidden, Convolutional, and Output layers are some of the layers incorporated in CNN. The proposed framework employed CNN with Bi-LSTM as the CL will study by extracting the complex features (attributes in the dataset (columns)) representations from pre-processed data. It will show a significantly low error rate in performance improvement of evaluation metrics.

The pseudocode for CNN is listed in algorithm 1.

Algorithm 1: Algorithm of CNN (Liu et al., 2021).
$Input$:
: : $train_x and\ train_y$: $the\ labels and\ features\ of\ the\ training\ set$
: $test_x and\ test_y$: $the\ labels and\ features\ of\ the\ test\ set$
max_time: $the\ maximum\ number\ of\ iterations\ to\ train\ the\ model$
$target\_error$: $the\ desired\ level\ of\ training\ error\ at\ which\ to\ stop\ training$
Lr_CPNN: $the\ learning\ rate\ for\ the\ CPNN$
$Initialization$: $The\ weights\ and\ scaling\ parameters\ for\ the\ CPNN\ and\ FCNN\ c$
$Output$:
: $w\ and\ b'$: $the\ weights\ and\ bias\ of\ the\ CPNN$
$W_{jk}\ and\ b_{jk}$: $the\ weights\ and\ bias\ of\ the\ FCNN$
$Steps$:
$step\ 1.\ Set\ the\ required\ parameters\ and\ initialize\ the\ weights\ and\ scaling\ par$
$step\ 2.\ While\ the\ current\ simulation\ time\ (t) is\ less\ than\ the\ maximum\ time\ a$
$the\ mean\ square\ error\ (L(t))\ is\ greater\ than\ the\ target\ error$:
$step\ 3.\ For\ each\ in\ the\ training\ set$:
$step\ 4.\ Calculate\ the\ predicted\ label\ (train\_p)\ using\ the\ forward\ calculation\ f$
$step\ 5.\ Calculate\ the\ mean\ square\ error\ (L(t))\ over\ the\ entire\ training\ set.$
$step\ 6.\ Update\ the\ weight\ and\ bias\ changes\ for\ the\ CPNN\ and\ FCNN$
$step\ 7.\ Adjust\ the\ weights\ and\ biases\ for\ the\ CPNN\ and\ FCNN$
$step\ 8.\ Increment\ the\ simulation\ time\ (t).$
$step\ 9.\ End\ while.$
$step\ 10.\ Output\ the\ weights\ and\ biases\ for\ the\ CPNN\ and\ FCNN.$

3.5. Bi-Lstm model

A Bidirectional LSTM (Bi-LSTM) consists of two LSTM models and is a type of sequence processing model (Imrana et al., 2021). The Bi-LSTM model takes input simultaneously from both the forward and backward directions. By incorporating information from both directions, Bi-LSTM effectively captures sequential dependencies in the input data and enhances the algorithm's capabilities.

Bi-LSTM is commonly used in Natural Language Processing (NLP) tasks. It differs from the standard LSTM model in that it processes the input in both directions.

This enables the model to leverage information from both past and future contexts, allowing it to capture more comprehensive dependencies between words and phrases in a sequence. Bidirectional LSTM is considered a powerful and versatile tool in NLP because it can effectively model sequential dependencies in dual directions.

Since Bi-LSTM uses 2 LSTM models, it has its advantages, which include (Imrana et al., 2021):

• Generating more meaningful output as it combines LSTM layers from both directions.

• Bi-LSTM is helpful in natural language processing since it produces diverse output for each sequence component.

• BI-LSTM can be used for various models such as classification, forecasting models, etc.

• Compared to LSTM, Bi-LSTM is more robust and efficient.

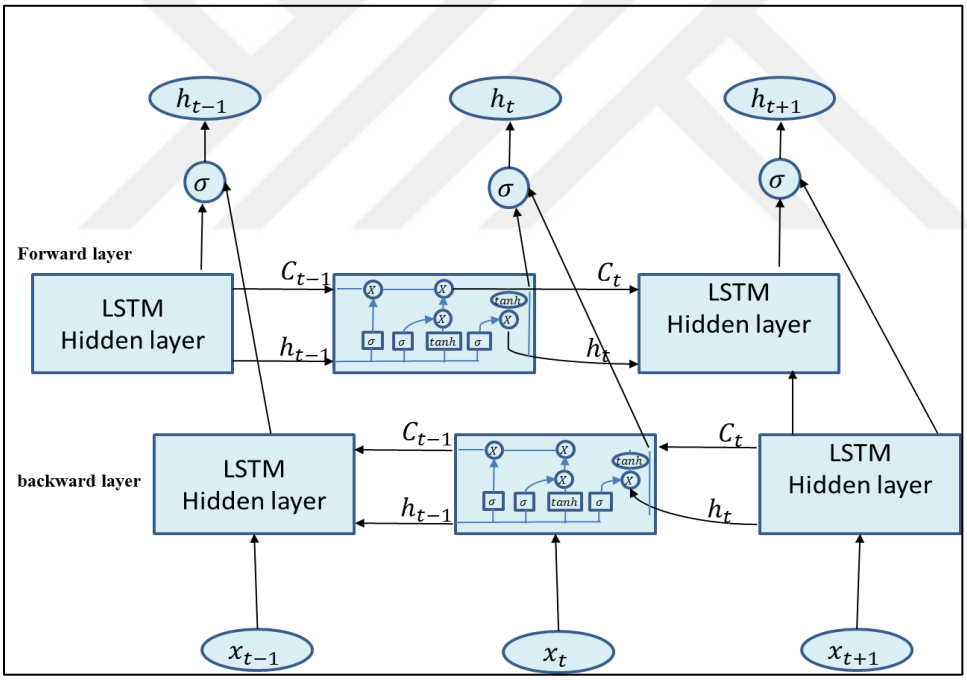

Figure 3.3. Architecture of Bi-LSTM

Figure 3.3 illustrates the working procedure of the Bi-LSTM model. An additional layer is incorporated alongside the existing LSTM layer, where the direction of information flow is reversed. This means the input sequence progresses in the additional LSTM layer in the opposite direction. Finally, the outputs from both LSTM layers are combined using various methods such as summation, concatenation, subtraction, etc.

Pseudocode for Bi-LSTM training is listed in algorithm 2

$Algorithm\ 2$: $BiLSTM\ Training$ (Imrana et al., 2021).
$Input$:
$Features$: $A\ set\ of\ features\ for\ each\ data\ point.$
$Labels$: $A\ set\ of\ labels\ for\ each\ data\ point.$
K: $The\ number\ of\ folds\ in\ the\ K-fold\ cross-validation.$
$Output$:
$A\ trained\ BiLSTM\ model.$
$Classifications\ for\ the\ test\ set.$
$Steps$:
$Step\ 1.\ Load\ the\ dataset.$
$Step\ 2.\ For\ each\ data\ point\ in\ the\ training\ and\ test\ sets$:
$Step\ 3.\ Extract\ the\ features\ (x).$
Step 4. $Extract\ the\ label\ (y).$
$Step\ 5.\ For\ each\ feature\ in\ x$:
$Step\ 6.\ If\ the\ feature\ is\ numerical, encode\ it\ using\ the\ Keras\ library.$
$available Step\ 7.\ Scale\ the\ features\ using\ Z' = (X - Xmin)/(Xmax - Xmin).$
$Step\ 8.\ For\ i\ from\ 1\ to\ n$:
$Step\ 9.\ Start\ with\ K = 10.$
$Step\ 10.\ Split\ the\ training\ set\ into\ K-groups.$
$Step\ 11.\ Load\ the\ BiLSTM\ model.$
$Step\ 12.\ Fit\ the\ model\ using\ K-1\ groups.$
$Step\ 13.\ Validate\ the\ model\ using\ the\ remaining\ Kth\ group.$
$Step\ 14.\ Repeat\ until\ all\ K-groups\ are\ used\ as\ validation\ sets.$
Step 15. $Test\ the\ model\ on\ the\ test\ set\ (UNSW\ NB\ Test).$
$Step\ 16.\ Output\ the\ trained\ BiLSTM\ model\ and\ the\ classifications\ for\ the\ test$

3.6. Bi-Lstm with CNN Rectified Linear

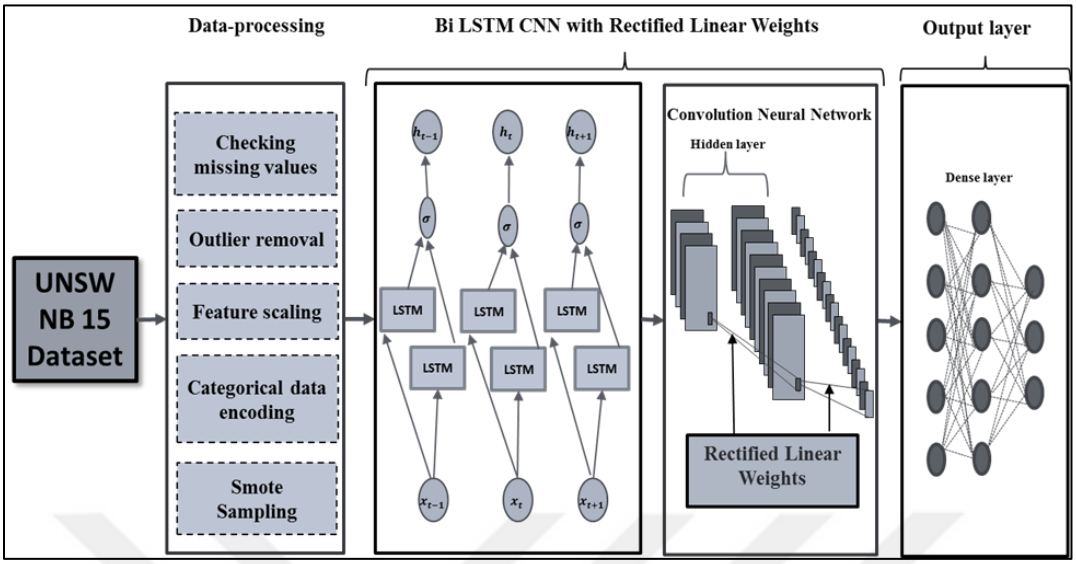

Figure 3.4. The architecture of Bi-LSTM with CNN rectified linear weights on the proposed model

Figure 3.4 illustrates the step-by-step architecture of the Bi-LSTM model with rectified linear weights applied to CNN. The initial step involves implementing the UNSW NB 15 dataset. The dataset values are pre-processed using various reprocessing techniques. The best weights obtained from the rectified linear weight are fed into the Bi-LSTM model with CNN. Finally, the output layer is obtained.

The architecture of the BiLSTM network helps address the issues of vanishing gradients or exploding gradients that occur in traditional networks. The BiLSTM's recurrent behavior allows the model to retain learned information over time, while the CNN effectively extracts data features. Since the model employs deep learning techniques for detecting network attacks, a manual approach is used for initializing the parameters. Parameters are selected to enhance the model's interpretability and improve its generalization. Therefore, it is essential to choose the parameters wisely, as an inappropriate selection can result in suboptimal efficacy outcomes. Table 3.1 displays the parameters used in the proposed model.

Table 3.1. Parameters of proposed model

Conv1D	Filters = 256 Kernel_size=3 Activation=’relu’
MaxPooling1D	Pool_size = 2
Conv1D	Filters = 128 Kernel_size=3 Activation=’softmax’

MaxPooling1D	Pool_size = 2
Conv1D	Filters = 256 Kernel_size=3 Activation='softmax'
MaxPooling1D	Pool_size = 2
Bidirectional(LSTM)	Units=64
Dropout	Value= 0.2
Dense	Units =2 Activation = 'softmax'

The algorithm presented below outlines the Bi-LSTM model with CNN using rectified linear weights. A well-known CNN model is utilized for the convolutional process, where a single-layer CNN model is trained for text classification. The CL layer is employed to generate phrase vectors. The convolution relationship between word vectors and kernels is explained in equation 3.1 below.

$$f_{ab} = tanh\ (W_{j:j+conv-1} \otimes\ K + b) \tag{3.1}$$

In which, f_{ab} denotes the a-th word vector and b is denoted as bias, and W is denoted as an amalgamation of all word vectors. Finally, the symbol $\otimes$ is represented as a Hadamard product. In the proposed model average pooling function is employed. This pooling function is used to obtain a single feature that denotes the original text word vector, represented in the below function 3.2.

$$S_{i,k} = f_a^b \tag{3.2}$$

3 main components are employed in the proposed framework. In which i, is represented as the input gate, which helps to calculate the information which needs to be saved. f_t is meant as forget gate, which helps identify the information before the information is; finally, ot is represented as an output gate, which helps generate the output state.

$$f_t\ =\ \sigma(W_f\ *\ [hid_{t-1}, x_t]\ +\ b_f) \tag{3.3}$$

$$i_t\ =\ o(W_i\ *\ [hid_{t-1}, x_t]\ +\ b_i) \tag{3.4}$$

$$\widetilde{conv}_t =\ tanh\ (W_{conv}\ *\ [hid_{t-1}, x_t]\ +\ b_{conv}) \tag{3.5}$$

$$conv_t\ =\ f_t\ *\ conv_{t-1} + i_t\ *\ \widetilde{conv}_t \tag{3.6}$$

$$o_t = \sigma(W_0 * [hid_{t-1}, x_t] + b_o) \quad (3.7)$$

$$hid_t = o_t * tanh\ tanh\ (conv_t) \quad (3.8)$$

$sigma\ is$ denoted as sigmoid function. The weight matrices of each gate are represented as W_f, W_i,W_c, and W_o. Hidden unit numbers are denoted as n, and at last, the output of the Bi-LSTM is denoted as h_t.

Input phrases are converted to a vector with a fixed length using an encoder. The decoder assists in restoring the vector to an equal-length output text phrase sequence. The equations are represented as follows:

$$y1 = f\ (conv) \quad (3.9)$$

$$y2 = f\ (conv, y1) \quad (3.10)$$

$$y3\ = f\ (conv, y1, y2) \quad (3.11)$$

$$C = f(x_1, x_2, \ldots, x_m) \quad (3.12)$$

In which, C is a semantic vector, which is expressed by a non-linear function of input phrases of a given specific text. However, the traditional sequence-to-sequence approach failed to record the context of the phrases, since C has a fixed value.

Hence, this problem can be overcome by adding semantic vectors like C1, C2, and C3. These vectors are called soft attention to capture the semantic differences and information of the context. Output is calculated using the formula given below 3.13.

$$y_i = f\ (conv_i, y_1, y_2, \ldots, y_{i-1}) \quad (3.13)$$

$$conv_i = \sum_j^{Tx} \quad a_{ij} hid_j \quad (3.14)$$

The value of C_i is calculated using equation 3.15,

$$yi = f\ (conv_i, y1, y2, \ldots, y_{i-1}) \quad (3.15)$$

In which, h_j is represented as annotation and a_{ij} are used as weight employed in the encoder.

However, best weights are fed into the Bi-LSTM model by employing equation 3.16,

$$conv_i = \sum_{j=1}^{Tx} a_{ij} h_j * \left(\frac{a_{x,y}^{i}}{(K + \propto \sum_{j=\left(0, i-\frac{n}{2}\right)}^{\left(N-1, i+\frac{n}{2}\right)} (a_{x,y}^{j})^2)}\right) \quad (3.16)$$

$a_{x,y}^{i}$ is denoted as the activity of the neuron which is calculated by using kernel i, at the position of (x, y). Further, ReLU non-linearity is applied, which generates the response normalized activity as given in equation 3.16.

Feature extraction using Bİ-LSTM model with CNN: Bi-LSTM with CNN employed various functions such as batch normalization, average pooling, Bi-LSTM, Conv 1, Flatten, Dense, and Softmax Function.

3.6.1. Batch normalization

Batch normalization is a technique used to accelerate and stabilize the training of artificial neural networks (ANNs) by normalizing the inputs of each layer through rescaling or recentering. By applying batch normalization techniques, higher learning rates can be used, resulting in faster network training.

Some advantages of batch normalization include:

- Accelerating the training process by reducing the number of iterations needed to reach convergence.
- Addressing the problem of internal covariate shift, which occurs when the distribution of input values to a layer change during training, leading to slower convergence.
- Smoothing the loss function aids in optimizing the model's parameters and further enhances the speed and efficiency of the model.

3.6.2. Average pooling

The pooling layer, a downsampling layer, is a component in neural networks that performs a similar operation to the convolutional layer. However, the pooling layer differs in that it uses filters without any weights. Instead, it applies aggregation functions within a receptive field, resulting in a condensed output array. Two commonly used types of pooling layers are max pooling and average pooling.

Using a pooling layer in the current study offers several advantages, including:

- Improvement in efficiency by reducing the dimensionality of the data.
- Reduction in model complexity, which can help prevent overfitting.

- Limiting the risk of overfitting by generalizing the learned features.
- Reducing the number of parameters to learn in CNN architecture as the pooling layer reduces the dimensions of the feature map.
- Summarizing features from a feature map region allows operations to be performed on summarized features instead of precisely positioned features. This enhances the model's robustness to variations in the input.

When using the average pooling method, there is a possibility that a sharp image may not be effectively detected. This is because the average pooling operation smooths out the image by taking the average value of all the pixels within the pooling region. One of the main reasons for implementing average pooling is to reduce the feature map size. By reducing the dimensionality of the data, the computation becomes more efficient and faster, particularly as the number of trainable parameters decreases. However, it is important to consider that the trade-off for this efficiency is the potential loss of detailed information, especially in the case of sharp or fine-grained features.

3.6.3. Softmax function

The softmax function is commonly used as the final activation function in a neural network. It is employed to regularize the network's output into a probability distribution rather than just projected output classes.

The softmax function originated in statistical mechanics and was further formalized and applied in various significant contexts. John S. Bridle is credited with using the term "softmax" in two conference papers, playing a vital role in naming this concept in machine learning. The softmax function possesses two key properties: each value ranges between 0 and 1, and the sum of all values is always 1. These properties are characteristic of a softmax layer.

The softmax activation function transforms a vector of K real values into a vector of K real values that add up to 1. It ensures that even if the input values are 0, greater than 1, or positive/negative, the softmax function maps them to values between zero and one, which can be interpreted as probabilities. Softmax is sometimes referred to as Multi-Class Logistic Regression (MCLR) or softmax function. It provides a logistic regression overview and can be used for multiclass classification. The formula for softmax closely resembles the sigmoid function used in logistic regression.

The name "softmax" comes from the fact that it is a softer version of the argmax function. It is also called "softmax" because it represents a smoother version of all existing

activation functions. In softmax, the output will be +1 for the largest input value, and the smallest input value will have an output of 0. Hence, softmax can be seen as a softened and more flexible version of the argmax function, which returns the index of the largest value. Softmax is often implemented as the final layer in a neural network. It takes the network's output and produces a classification prediction, providing a probability for each class (summed up as 1).

4. RESULTS AND DISCUSSION

The results and discussion of the proposed methodology are presented in the following chapter. The UNSW-NB 15 dataset is used in the proposed framework, and a comparative analysis is conducted along with the proposed framework. The chapter provides an in-depth examination of the outcomes achieved and their implications.

4.1. Dataset description

The UNSW-NB15 dataset was generated in the cyber range lab of UNSW Canberra using the IXIA PerfectStorm tool. This dataset consists of 9 types of attacks, including worms, exploits, generic, DoS, fuzzes, backdoors, shellcode, analysis, and reconnaissance.

The dataset was created to provide a combination of real-world normal network traffic and synthetic malicious activities. A total of 100GB of raw traffic data was captured using the tcpdump tool. To generate the dataset, 12 algorithms were developed, and tools such as Argus IDS and Bro-IDS were utilized. The dataset includes 49 features along with class labels. Some of the features of the dataset include:

• The total number of records is 2,540,044, which are stored in four different CSV files.

• The dataset was partitioned into training and testing sets, namely UNSW_NB15_training_set and UNSW_NB15_testing_set. The training set contains 175,341 records, while the testing set contains 82,332 records, comprising various types of attacks and normal data.

For more information and access to the dataset, you can visit the following link: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

4.2. Performance metrics

Performance metrics have become a part of every machine learning (ML) task. These metrics are used to analyze the classification models for the given balanced datasets. The performance of the proposed framework design is evaluated using metrics such as F1-score, precision, accuracy, and recall.

a) Precision (Pc)

The term precision is defined as the ratio of true positive cases (TP) to the total number of cases that are correctly classified (TP+FP). It represents the accuracy and reliability of the predictions. It is calculated using equation (4.1).

$$Pc = \frac{TRP}{FLP+TRP} \tag{4.1}$$

In equation (4.1), TRP refers to True-positive, and FLP refers to False-positive, respectively.

b) Recall (Rc)

The term recall is signified as the reclusive of the production metric that estimates the total of accurate positive categories made out of all the optimistic categories. It is calculated with the following equation (4.2).

$$Rc = \frac{TRP}{FLN+TRP} \tag{4.2}$$

In equation (4.2), FLN refers to False-negative.

c) Accuracy (Acc)

The term accuracy is denoted as the system organization rate that is conveyed through the quantity of correctly classified cases (TRN+TRP) to the total cases of the dataset (TRP+FLP+TRN+FLN). The accuracy range is calculated with the following equation (4.3).

$$Acc = \frac{TRN+TRP}{TRP+FLP+TRN+FLN} \tag{4.3}$$

Where TRN refers to True-negative and FLN refers to False-negative.

4.3. EDA (Exploratory Data Analysis)

EDA, or Exploratory Data Analysis, is a crucial procedure that involves conducting initial data analysis to reveal patterns, identify anomalies, test hypotheses, and validate assumptions through graphical representations and summary statistics.

The primary objective of EDA is to examine the data before making any assumptions. It helps identify noticeable errors, better understand the data, detect outliers and anomalous events, and uncover interesting relationships between variables. This is

achieved by visualizing the data using various plots such as scatter plots, box plots, and histograms.

The performance of the proposed system is assessed using a confusion matrix. The confusion matrix is a table that describes the performance of a classifier. It provides information about the errors made by the classifier and the prevalent types of errors. It also helps understand how the classifier is confused or disoriented when making predictions. Figure 4.1 displays the confusion matrix of the proposed method, where the classes represented in the black box indicate misclassifications, while the classes in the remaining boxes represent correct classifications. Additionally, Figure 4.2 illustrates the proposed model's ROC (Receiver Operating Characteristic) curve, where the orange line represents the ROC curve. The ROC curve obtained by the proposed model is 0.94.14.

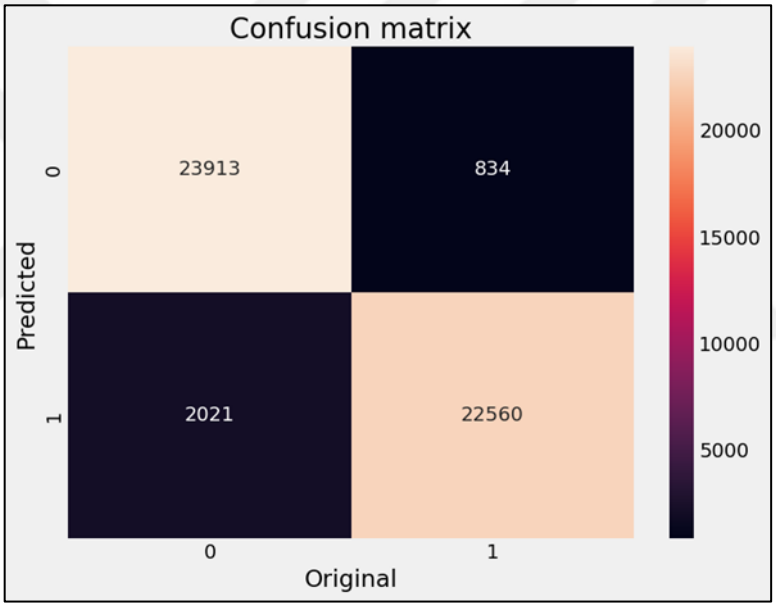

Figure 4.1. Confusion matrix

Based on the findings presented in Figure 4.1, several observations can be made:

- Accuracy: The model achieves an accuracy of approximately 96.53%, indicating that it made correct predictions for approximately 96.53% of all cases within the dataset.

- The high accuracy level suggests that the model effectively differentiates between the "zero" and "one" classes. Possible reasons for this outstanding performance may include utilizing a highly optimized model architecture, selecting relevant features, and using a well-balanced dataset.

- Precision: Precision refers to the level of accuracy or exactness in measurement or calculation.

- The model demonstrates a precision of approximately 91.76%, indicating that its positive predictions, specifically those labeled as "one" in this context, are correct approximately 91.76% of the time.

- A high precision score signifies that the model's predictions of "one" are highly accurate. This is particularly advantageous in situations where false positives, wherein a prediction of "one" is made when the actual value is "zero," have significant costs, or are undesirable.

The recall score, approximately 96.46%, signifies the model's capability to correctly identify around 96.46% of instances classified as "one" among the real examples. A high recall score indicates the model's effectiveness in accurately identifying and capturing a significant portion of the "one" class. This is particularly important in scenarios where failure to identify individual occurrences could have substantial repercussions. Specificity refers to the level of detail or precision in a given context. The model's accuracy in identifying the "zero" class is roughly 92.21%. A high level of specificity indicates the model's proficiency in accurately recognizing instances belonging to the "zero" class as truly representative of the "zero" class. This feature is particularly beneficial in roles that require precise identification of the absence of a specific condition. Furthermore, the F1-Score is a metric commonly used in academic research and evaluation to assess the balance between precision and recall. An F1 score of approximately 94.06% signifies a harmonious equilibrium between precision and recall. Moreover, the F1 score takes into account both false positives and false negatives. A high F1 score indicates the model's strong performance in accurately recognizing positive cases and effectively reducing the occurrence of false positives. The false positive rate (FPR) refers to the proportion of negative instances that are incorrectly classified as positive.

The false positive rate (FPR) of approximately 7.88% indicates that roughly 7.88% of instances that were labeled as "zero" were erroneously classified as "one." A low false positive rate (FPR) suggests that the model adopts a cautious approach in its predictions of the positive class ("one"), thereby reducing the occurrence of false positive errors.

This section will comprehensively analyze and provide a well-founded rationale for our arguments. The findings derived from the confusion matrix demonstrate that the model performs proficiently in accurately distinguishing between the "zero" and "one" classes. The model demonstrates robustness and effectiveness, as evidenced by its high

accuracy, precision, recall, and F1 score. It effectively strikes a balance between accurately forecasting individual events and reducing the occurrence of false positives. The model's cautious approach to making positive predictions is evident through its relatively low false positive rate. However, it is imperative to consider additional contextual information to evaluate these findings comprehensively. Various factors, such as class distribution, potential imbalances in class representation, and the specific domain or application of the model, may impact the importance of these metrics. Careful evaluation of the trade-offs between precision and recall is crucial, as it allows consideration of the specific objectives and demands of the given challenge.

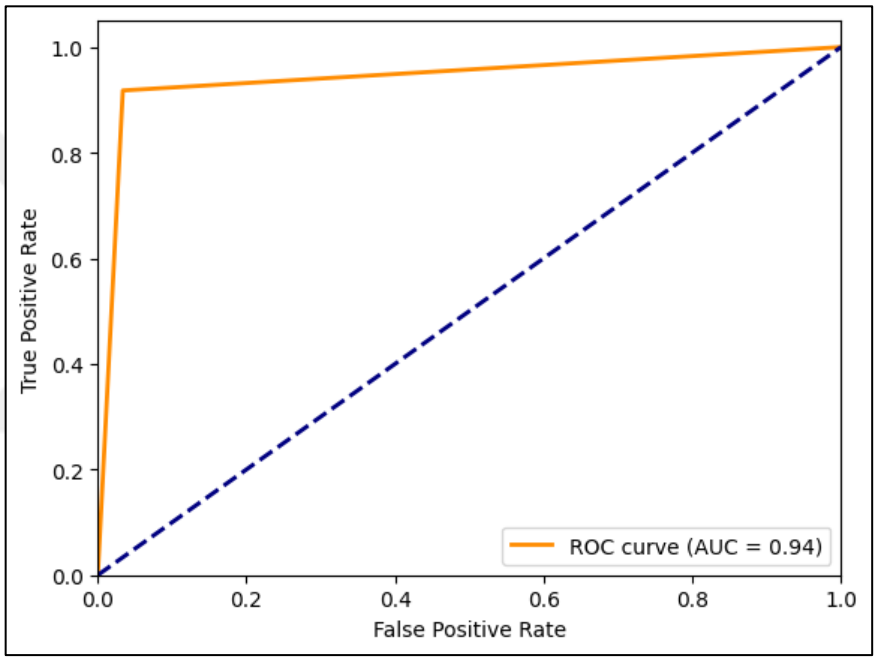

Figure 4.2. ROC curve

The Receiver Operating Characteristic (ROC) curve, as depicted in Figure 4.2, serves as an important tool for evaluating and illustrating the effectiveness of binary classification models at various classification thresholds. This graphical representation showcases the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) as the threshold for categorizing positive events is adjusted. However, in this scenario, where only one confusion matrix is provided, the ROC curve is represented by a single data point. Receiver Operating Characteristic (ROC) curve, taking into consideration the provided values within the confusion matrix. Furthermore, the true positive rate, also known as sensitivity, refers to the proportion of actual positive cases correctly identified as positive by a diagnostic test or model. The true positive rate (TPR)

is calculated by dividing the number of true positives (TP) by the sum of true positives and false negatives (TP + FN). The true positive rate (TPR) is a metric that quantifies the proportion of actual positive cases (specifically, instances belonging to class "one") that are accurately identified as positive by the model. In the above situation, the true positive rate (TPR) is estimated to be approximately 0.9646. A high true positive rate (TPR) signifies the model's ability to identify a significant proportion of true positive events accurately. The observed positive outcome indicates that the model demonstrates sensitivity toward the "one" class, which is crucial in situations where detecting "one" instances is of utmost importance. The false positive rate (FPR) can be calculated by dividing the number of false positives (FP) by the sum of false positives and true negatives (FP + TN). The False Positive Rate (FPR) denotes the ratio of true negative cases (classified as class "zero") that the model inaccurately identifies as positive. The false positive rate (FPR) is estimated to be roughly 0.0788 in the given scenario. A low false positive rate (FPR) signifies the model's effectiveness in reducing the occurrence of false positive errors. This holds significant value, particularly in scenarios where false positives carry substantial repercussions or where precisely detecting the absence of the specific class is of utmost importance. The rationale for utilizing a single point on the Receiver Operating Characteristic (ROC) curve is that the ROC curve comprises various points corresponding to different classification thresholds. Among these points, a single point holds significance in evaluating the model's performance. This point is determined by the True Positive Rate (TPR) and False Positive Rate (FPR) values derived from the confusion matrix. Analyzing this specific point offers valuable insights into the model's effectiveness at a particular classification threshold. In this scenario, the primary focus pertains to the model's ability to achieve a notable True Positive Rate (also known as sensitivity) while simultaneously maintaining a comparatively low False Positive Rate. The observed balance indicates that the model demonstrates proficiency in accurately distinguishing between the "zero" and "one" categories while also exhibiting sensitivity to the presence of the "one" category. It's important to note that a typical receiver operating characteristic (ROC) curve comprises multiple data points, each representing a distinct threshold. This graphical tool facilitates the examination of inherent trade-offs between sensitivity and specificity. However, in this particular scenario, the single data point continues to provide significant insights into the model's capacity to accurately classify instances belonging to the "one" class while effectively managing the occurrence of false positives.

4.3.1. Count plot visualization of data set

A count plot is primarily used to represent the presence of observations within a categorical variable. It serves as a visual representation of data.

A count plot using bars displays the number of observations in each categorical bin. It can also be described as a histogram across a categorical variable, as opposed to a quantitative one. The count plot of "dttl," as illustrated in Figure 4.3, demonstrates that the count decreases as the value of "dttl" fluctuates.

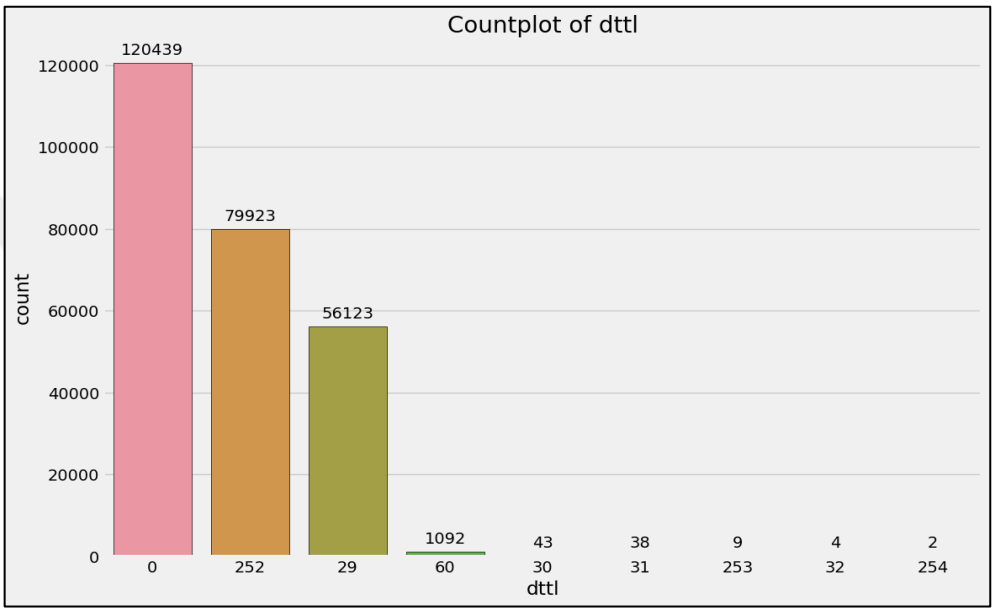

Figure 4.3. Countplot of dttl

The count plot graphically illustrates the relative frequency of various "dttl" values. The value "dttl" 0 exhibits the highest frequency among the plotted values, totaling 120,439 occurrences. It is closely followed by the value 252, which has a frequency of 79,923 occurrences. The other values (29, 60, 30, 31, 253, 32, and 254) display lower frequencies, with occurrences ranging from 56,123 down to 2. This plot analyzes the distribution of "dttl" values, highlighting the prevalence of specific values and the relative scarcity of others within the dataset.

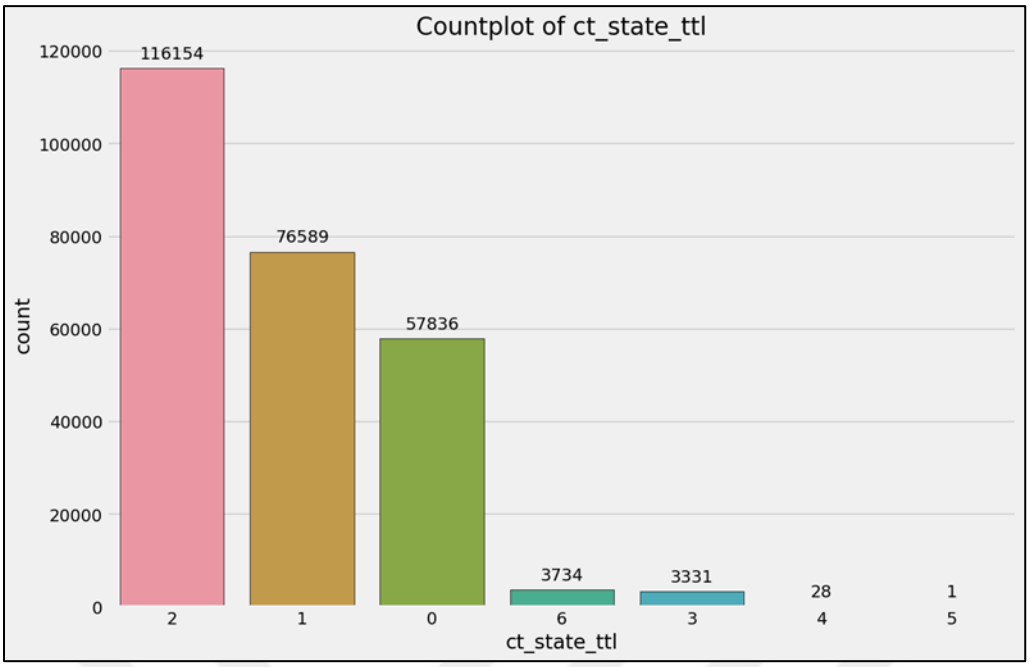

Figure 4.4. Countplot of ct_state_ttl

• The count plot visually represents the distribution of several categories inside the "CT_state_ttl" variable. Each bar depicted on the plot represents a distinct "CT_state_ttl" value, with the vertical dimension of the bar indicating the frequency of occurrences for that unique value. In this instance, the x-axis represents the values of "CT_state_ttl," which are 2, 1, 0, 6, 3, 4, and 5. The graph's vertical axis reflects the frequency of each "CT_state_ttl" value.

• The following is an analysis of the frequency distribution for each value of "CT_state_ttl":

• Regarding the variable "CT_state_ttl = 2", it is observed that the count is 116,145.

• The count for the variable "CT_state_ttl" is 76,589.

• Regarding the value assigned to the variable "CT_state_ttl" being equal to zero, it is observed that the count is 57,836.

• The count for the variable "CT_state_ttl" being equal to 6 is 3,734.

• Regarding the variable "CT_state_ttl = 3", it is observed that the count is 3,331.

• The value of "CT_state_ttl" is 4, resulting in a count of 28.

• Regarding the variable "CT_state_ttl = 5", the count is equal to 1.

• Analyzing the Narrative:

• The bar with the greatest height on the figure represents the category "CT_state_ttl = 2," signifying that this particular value has the highest frequency compared to all other categories of "CT_state_ttl."

• The bar with the second highest number represents "CT_state_ttl = 1," which is a frequently occurring value.

• The variable "CT_state_ttl" also has a significant count.

• The counts for "CT_state_ttl = 6" and "CT_state_ttl = 3" decrease compared to the three highest values. "CT_state_ttl = 4" and "CT_state_ttl = 5" have notably low counts, with only a limited number of instances.

• In essence, the count plot offers a lucid graphical depiction of the frequency distribution of "CT_state_ttl" values and their respective counts. This feature facilitates the rapid identification of the most frequently occurring values and the overall distribution pattern of a variable within a given dataset.

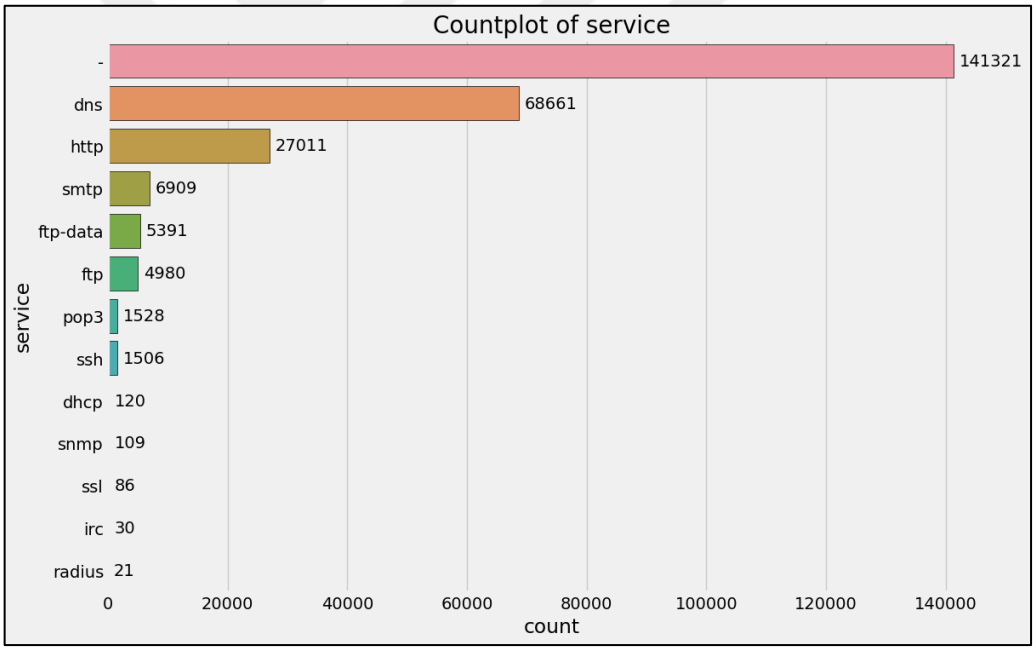

Figure 4.5. Countplot of service

The count plot provides a visual representation of the distribution of different network services based on their frequencies of occurrence within the dataset. Each service is associated with a unique numerical value, representing the observed number of instances or occurrences of that service that were observed.

The Domain Name System (DNS) is the most frequently occurring service in the dataset, with a count of 68,661. This suggests that DNS-related tasks, such as domain lookups and resolution, are widespread and prominent.

1. The Hypertext Transfer Protocol (HTTP) has a total count of 27,011. This high number suggests a notable prevalence of online interactions, indicating that people frequently interact with websites or online resources.

2. The Simple Mail Transfer Protocol (SMTP) has been seen 6,909 times, indicating a notable prevalence of email communication involving the transmission of messages.

3. The FTP-Data (File Transfer Protocol - Data) category has a count of 5,391, indicating that the use of FTP for data transfers is significant and plays a significant role in the overall composition of the dataset.

4. The File Transfer Protocol (FTP) has been observed 4,980 times, signifying the frequency of file transfers conducted using FTP connections.

5. POP3 (Post Office Protocol version 3) occurs 1,528 times, indicating the use of this protocol to retrieve emails from mail servers.

6. The term "SSH" (Secure Shell) is seen 1,506 times, indicating its importance in facilitating secure remote access and command execution operations.

7. The Dynamic Host Configuration Protocol (DHCP) has a count of 120, suggesting that the allocation of dynamic IP addresses in the dataset is relatively infrequent.

8. The Simple Network Management Protocol (SNMP) was observed a total of 109 times, indicating its use in network administration and monitoring activities.

9. The acronym SSL, which stands for Secure Sockets Layer, is used a total of 86 times in the text. This indicates the presence of secure communication sessions and the transport of encrypted data.

10. Internet Relay Chat (IRC) is a communication protocol with relatively lower frequency, as indicated by a count of 30, suggesting occasional use of IRC protocols.

The term "radius" is observed 21 times, suggesting a relatively low frequency of interactions involving Remote Authentication Dial-In User Service (RADIUS).

In brief, the count plot provides valuable insights into the distribution and prevalence of different network services within the dataset. The data presented highlights the prevalence of DNS and HTTP operations, as well as notable figures for email-related protocols such as SMTP and POP3 and file transfer protocols like FTP and FTP-Data. Less frequent operations include using secure protocols such as SSH and SSL, network administration through SNMP, and various other specialized services. The narrative deftly portrays the breadth of network activity and its various frequencies.

Figure 4.4 and figure 4.5 show the count plot of the ct_state_ttl and countplot of service.

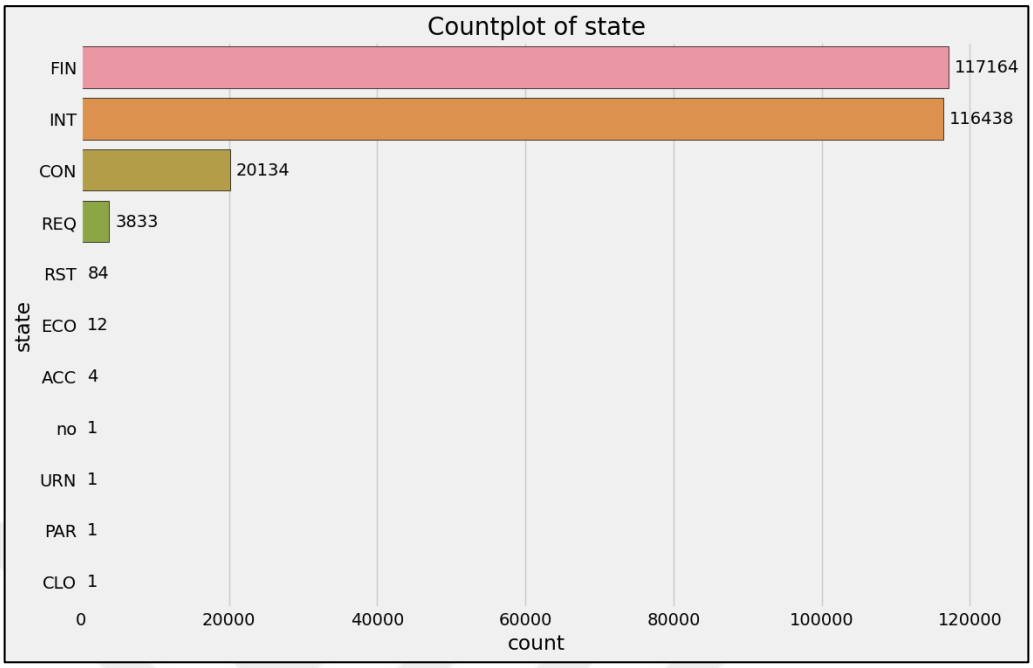

Figure 4.6. Count plot of state

The count plot is a graphical representation that visually depicts the distribution of service categories by their corresponding count values. Each distinct service category is associated with a specific label, and the count number represents the frequency of occurrence of each category within the dataset. The provided dataset includes the following service categories:

The service category of Financial Services denoted as FIN, is seen to have a total count of 117,164 instances. This finding suggests that the financial services category is the most common occurrence in the dataset.

The category of Internet Services consists of a total of 116,428 instances. This observation indicates that the dataset contains a significant presence of online services.

According to the data, there is a total of 20,134 occurrences of consultancy services. Compared to the two leading categories, this particular category exhibits a moderate level of representation.

According to the data, there have been a total of 3,833 occurrences of request services. This particular category appears to be rather infrequent, maybe suggesting the presence of distinct user inquiries.

The dataset contains a total of 84 occurrences of restaurant services. This particular category constitutes a relatively minor fraction of the dataset.

According to the data, there are a total of 12 occurrences of services that are considered eco-friendly. This particular category exhibits a relatively low prevalence in comparison to other categories.

There are four instances of accounting services within the organization denoted as ACC. The representation of this category is significantly low.

There is one case that falls under the category of "Other Services." This implies the existence of services that do not fit into the primary classifications.

There is only one case for each of the categories URN (Urgent Services), PAR (Parenting Services), and CLO (Clothing Services), suggesting that they are infrequent occurrences within the dataset. In general, the count plot effectively presents a visually coherent representation of the distribution of service categories within the dataset. The data indicates that financial and internet services exhibit the highest degree of prevalence, whereas consultancy, requests, and restaurant services demonstrate various amounts of representation. The graphic serves as a valuable tool for comprehending the relative frequencies of various service types and discerning potential trends or patterns within the data.

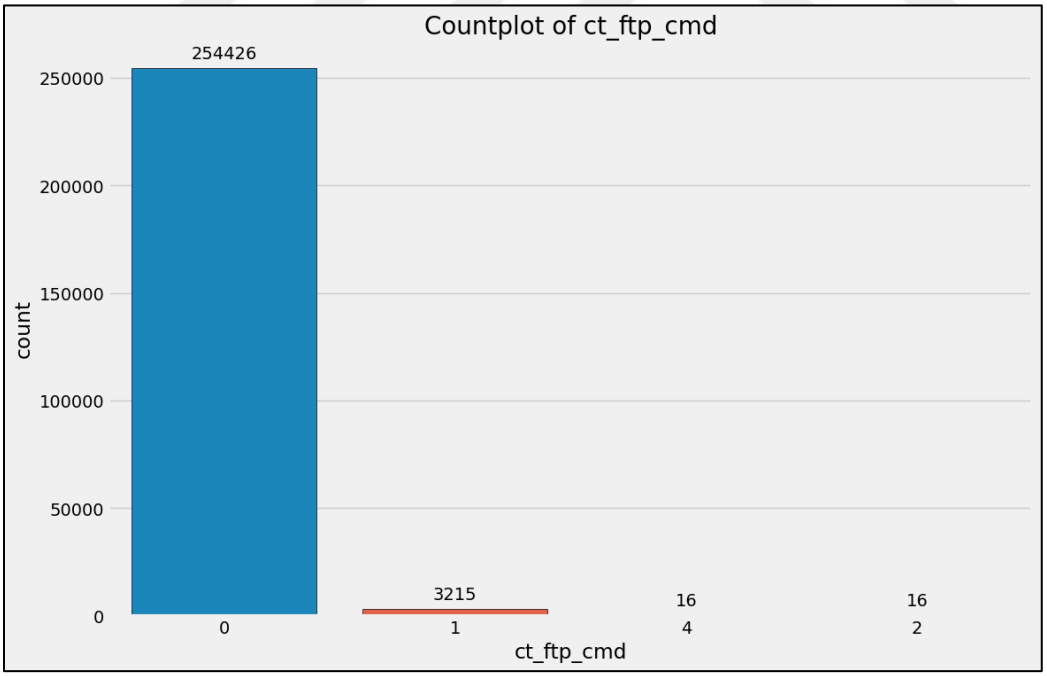

Figure 4.7. Countplot of ct_ftp_cmd

Figure 4.7 shows a count plot illustrating the distribution of different categories in the CT_FTP_CMD variable. The variable has four distinct categories: zero, one, four, and two. The frequencies of each category are as follows:

- The term "zero" has been observed 254,426 times, while the term "one" has been recorded 3,215 times.

- The terms "four" and "two" have been documented 16 times.

The count plot graphically represents the frequencies of these counts. The length of each bar is directly proportional to the number of cases that belong to each respective category. In this case, the category labeled as "zero" exhibits the highest frequency, whereas the category labeled as "one" demonstrates a comparatively lower occurrence. The frequencies of the categories "four" and "two" exhibit significant counts in comparison.

The count plot presented offers a comprehensive depiction of the distribution of CT_FTP_CMD categories, emphasizing the predominance of occurrences categorized as "zero" and the comparatively lower frequencies of cases labeled as "one," "four," and "two." Utilizing this visualization aids in quickly understanding the distribution of different categories in the variable, facilitating informed decision-making about data analysis or modeling.

Correlation Plot: When a dataset contains a large number of columns, one of the best ways to assess the correlation among the columns is by visualizing it as a correlation matrix plot.

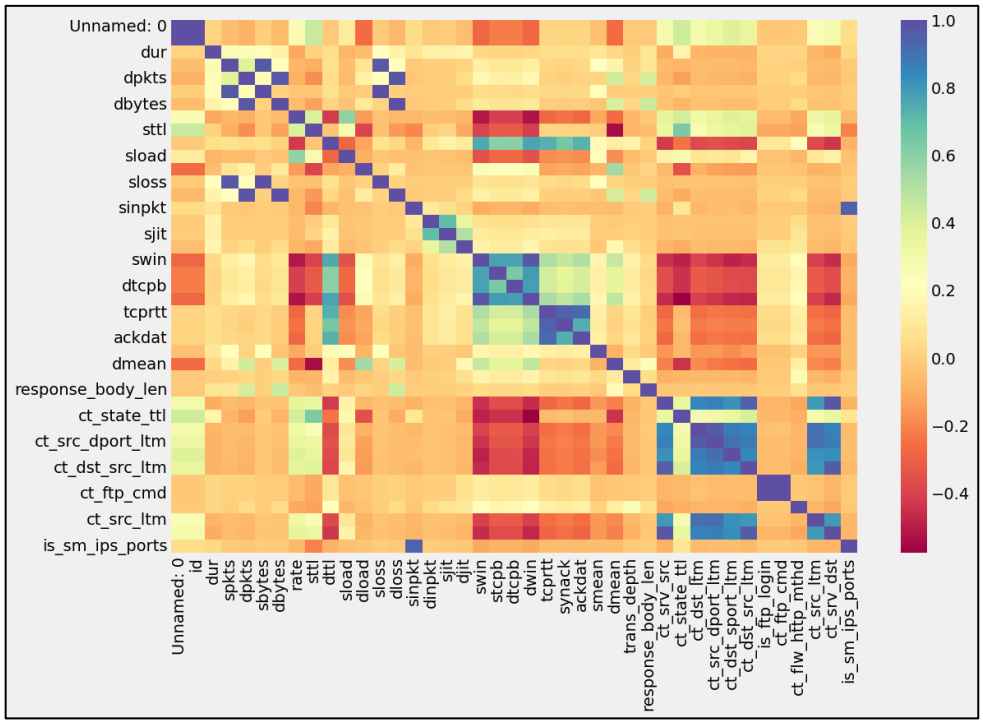

Figure 4.8. Correlation plot

Figure 4.8 shows the correlation plot for a proposed model for the UNSW-NB 15 dataset using different variables.

A correlation plot, sometimes represented as a heatmap, describes the relationships between variables in a given dataset. Each cell in the matrix has a correlation coefficient, quantifying the magnitude and direction of linear associations between two given variables. The heatmap visualization uses color gradients to depict the magnitudes of correlation, where different colors are used to signify different levels of correlation.

Positive correlations near +1 indicate that the other variable tends to increase as one variable increases.

Negative correlations, which are close to -1, indicate an inverse relationship between the variables.

Correlations close to 0 suggest a lack of substantial linear relationships. The aforementioned above holds significant value in the context of pattern recognition and the detection of dependencies. For example, identifying a strong positive correlation between two variables indicates a simultaneous increase or decrease in their values. Still, a large negative correlation suggests an inverse relationship where one variable increases as the other falls. Through the revelation of these links, data analysts acquire valuable insights into the interactions between features and their potential for predictive capability. In addition, the correlation plot serves as a valuable tool for identifying multicollinearity, a condition characterized by strong correlations between variables, which can adversely affect a model's stability and interpretability. The utilization of visualization plays a vital role in the process of exploratory data analysis, as it aids in the selection of features and the building of models. This capability enables analysts to make well-informed decisions regarding the inclusion or exclusion of features, hence improving their data analysis and modeling efforts

4.3.2. Box plot and histogram visualization of data set

A box plot, also known as a box and whisker plot, displays the five-number summary of a dataset. It includes the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum values. The interquartile range (IQR) is also shown, and outliers may be identified. Box plots are useful for visualizing the distribution and key characteristics of data.

Histograms, on the other hand, are bar plots used to represent numerical data that is divided into bins. They provide a convenient way to examine the distribution of data.

Histograms are particularly useful when working with large datasets as they can reveal patterns, outliers, and gaps in the data.

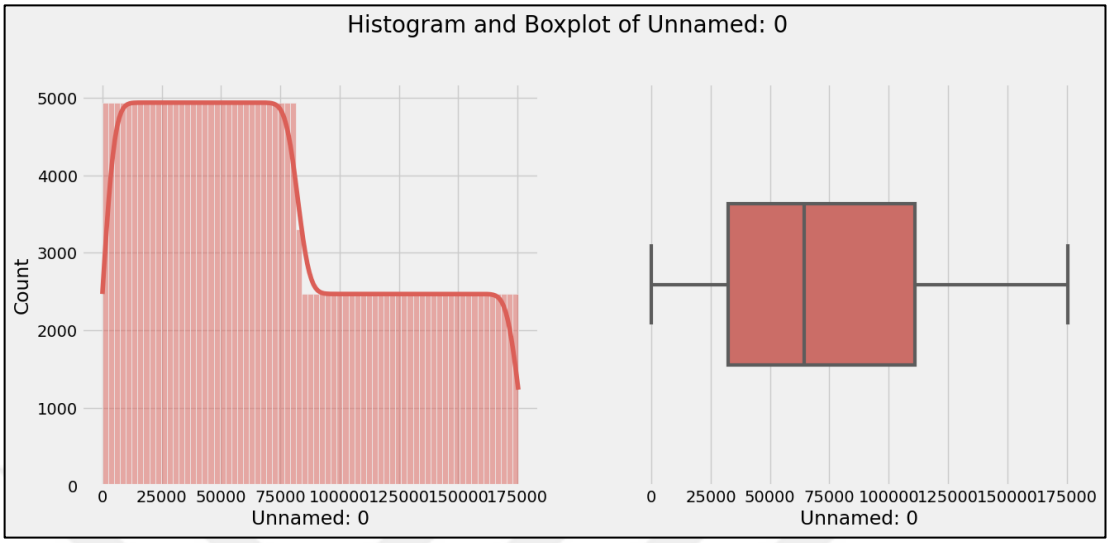

Figure 4.9. Histogram and boxplot of unnamed

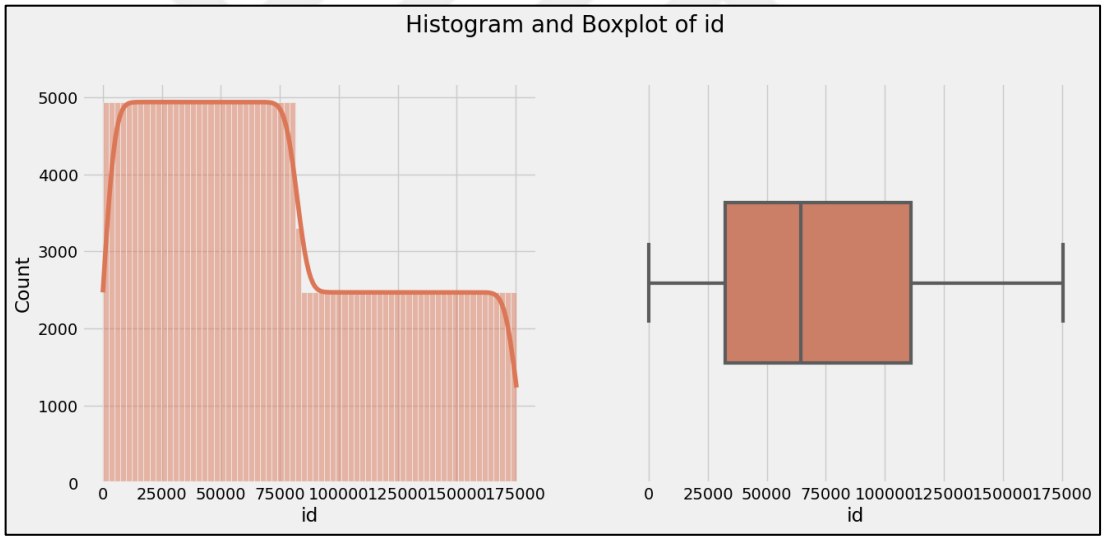

Figure 4.10. Histogram and boxplot of id

Figure 4.9 shows the untamed histogram and box plot, whereas Figure 4.10 shows Id's histogram and box plot.

In Figures 4.8 and 4.9, a graphical representation known as a box plot, commonly referred to as a box-and-whisker plot, is used to provide a comprehensive breakdown of a dataset's distribution. This visualization includes critical statistical variables, namely the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum values. Additionally, the box plot not only presents measures of central tendency but also highlights the interquartile range (IQR), illustrating the dispersion of the middle 50% of

the dataset. Outliers, defined as data points that deviate significantly from the average range, can be easily detected. Box plots offer a comprehensive visual representation of data dispersion and asymmetry, aiding in understanding data distribution and identifying potential outliers.

In contrast, histograms provide a graphical depiction of the frequency distribution of quantitative data. Histograms represent data by dividing it into intervals, often called bins, and visually conveying the frequency or quantity of observations within each bin using vertical bars. This graphical representation facilitates a thorough data distribution analysis, focusing on the frequency of events occurring within distinct intervals. Histograms offer significant advantages in analyzing extensive datasets as they can reveal underlying patterns, detect outliers or anomalies, and uncover potential gaps or clusters within the data. This visualization aids in understanding the distributional characteristics of the data, thereby assisting analysts in making informed judgments and deriving valuable insights.

The model's training versus the proposed model's validation loss is defined in Figure 4.11, where the blue color represents the training loss. In contrast, the orange curve represents the validation loss.

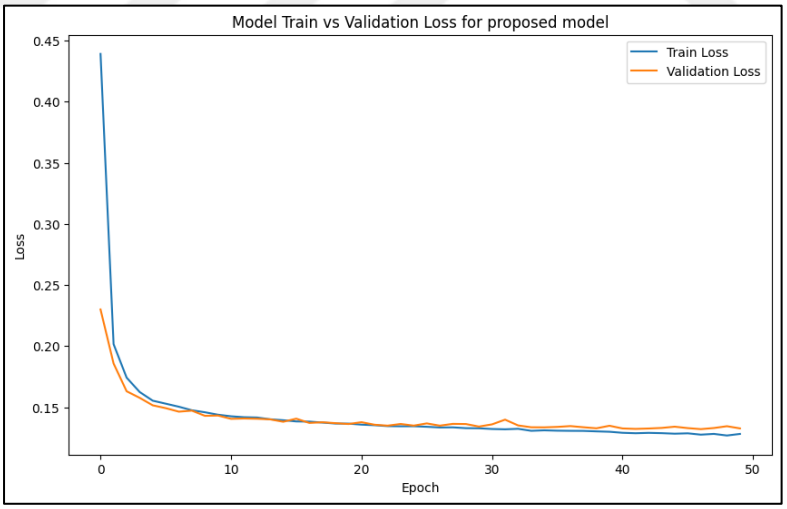

Figure 4.11. Model train vs validation loss for proposed model

The diagram depicted in Figure 4.11 illustrates the training curve demonstrating a steady decrease in validation loss as the number of epochs increases, which is a strong indicator of effective model training and successful convergence. This curve visually represents the progressive improvement in a machine learning model's performance throughout multiple training iterations. Initially, it's common for both the training and

validation losses to have relatively high values, indicating a lack of accurate predictions at the beginning. However, as the model undergoes further epochs, it gradually adjusts its internal parameters to capture the underlying patterns within the dataset more effectively.

The key observation in this context relates to the decreasing validation loss. This signifies that the model is learning from the training data and developing the ability to apply its learned knowledge to new, unfamiliar data. This is a crucial objective in the field of machine learning. The observed decrease in validation loss indicates the model's improved predictive power and underscores its proficiency in recognizing meaningful patterns. The gradual enhancement of the model over several epochs suggests it is acquiring knowledge robustly and reliably.

However, knowing the potential convergence point at which the validation loss may reach a plateau or exhibit variations is important. This observation suggests the model has reached its maximum capacity to extract information from the provided data. Further training could lead to overfitting, where the model becomes overly tailored to the training data and struggles to generalize to new data.

In summary, the presence of a training curve showing a decreasing validation loss over epochs is compelling evidence of the model's ability to acquire knowledge and apply it in a broader context. This highlights the model's proficiency in extracting significant insights from the training data and using them in unfamiliar scenarios. By carefully monitoring the curve's pattern, one can achieve a highly optimized model that strikes a balance between learning and generalization, ultimately enhancing the model's ability to make accurate predictions.

4.4. Performance analysis

Precision, recall, f-score, and Accuracy obtained by the proposed method are tabulated in Table 4.1. The accuracy, f-score, recall, and precision attained by the proposed model is 0.94.14.

Table 4.1. Performance of the proposed model

S.No	Parameter	Proposed
1	Precision	0.9433
2	Recall	0.9439
3	f-score	0.9443
4	Accuracy (%)	94.14%

4.5. Comparative analysis

Comparative analysis deals with comparing various existing models with the proposed model. Table 4.2 shows the comparative analysis of the proposed model with the existing ones. The existing models employed were I-GA, which resulted in 85.99% accuracy, DBN provided an accuracy rate of 82.00%; CDBN used in the existing study delivered an accuracy rate of 82.29%; the existing model implemented provided an accuracy rate of 86.49, and finally, the proposed framework outperformed the existing models by providing accuracy rate of 94.14%. A graphical representation of the existing and proposed models is depicted in Figure 4.12.

Table 4.2. Comparative Analysis of Existing Models (Barrons, March,29,2021)

UNSW-NB15 dataset	
Model	**Accuracy**
I-GA	85.99
DBN	82.00
CDBN	82.29
Existing model	86.49
Proposed model	**94.14**

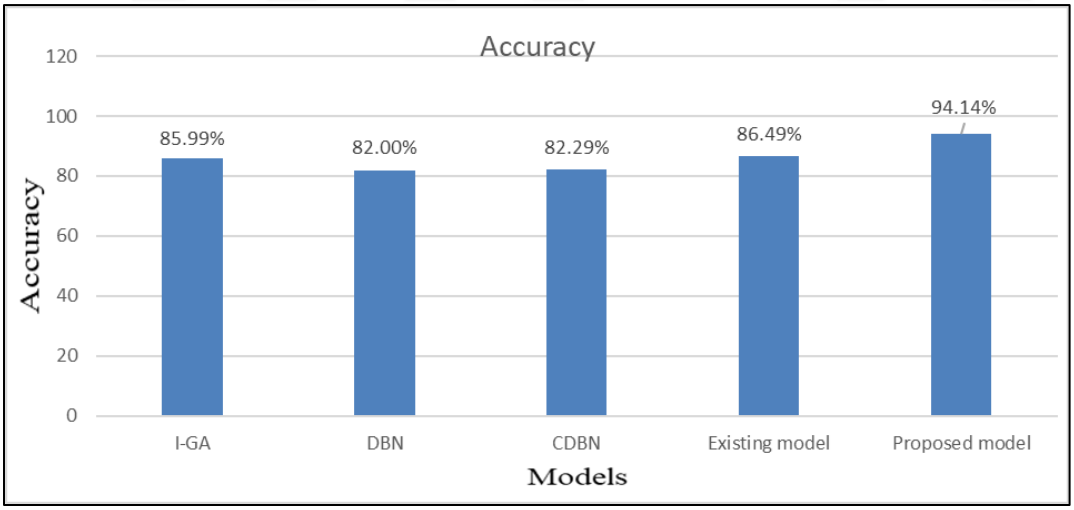

Figure 4.12. Graphical representation of existing models (Barrons, March,29,2021)

Table 4.3 shows the comparative analysis of the existing and proposed models. Various techniques were implemented, in which XGBoost produced an accuracy rate of 77.51%, REPTree produced an accuracy rate of 84.1%, FFDNN produced an accuracy rate of 77.16%, 82.1% of accuracy rate was offered by IGRF-RFE, along with KS delivered an accuracy rate of 81.34%, in addition to it, existing model delivered around

84.24% of accuracy rate, however, proposed methodology outperformed the existing models by providing accuracy rate of 94.14%. A graphical representation of the comparative analysis is mentioned in Figure 4.13.

Table 4.3. Comparative Analysis of Existing Models (Yin et al., 2023).

UNSW-NB15 dataset	
Model	**Accuracy**
XGBoost	77.51
REPTree	84.1
FFDNN	77.16
IGRF-RFE	82.1
KS	81.34
Existing model	84.24
Proposed model	**94.14**

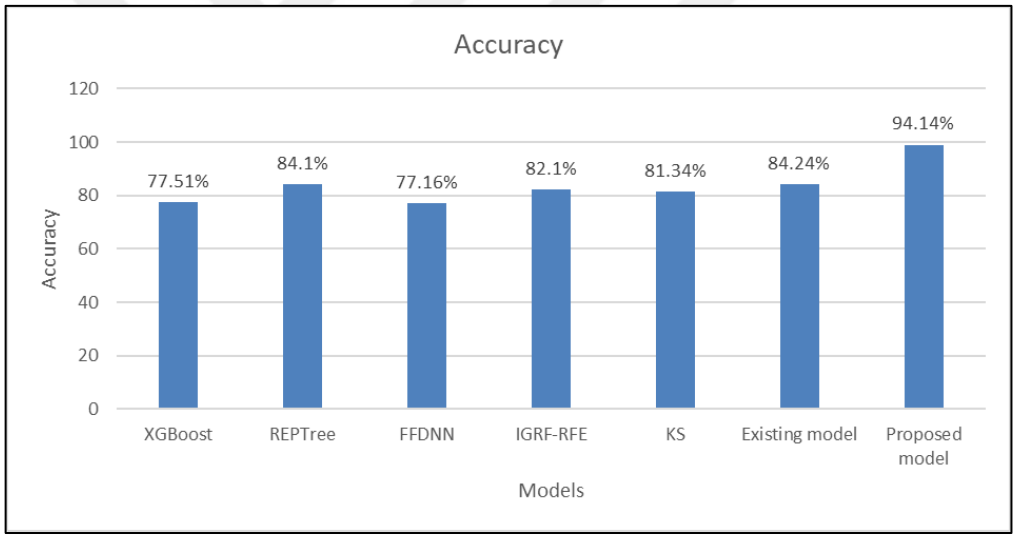

Figure 4.13. Graphical representation of existing models (Yin et al., 2023).

Another comparative analysis of the existing methods with the proposed method is presented in Table 4.4, which compares the accuracy rates of the existing models with the proposed model. The existing model achieved an accuracy rate of Ada-boost model 73.32%, the LSTM model achieved an accuracy rate of 92.43%, the SVM model achieved an accuracy rate of 74.32%, and the existing model had an accuracy rate of 82.3%. In contrast, the proposed model achieved an impressive accuracy rate of 94.14%. The graphical representation of the findings from the existing study is illustrated in Figure 4.14.

Table 4.4. Comparative Analysis of Existing Methods (Jain et al., 2022)

UNSW-NB15 dataset	
Model	**Accuracy**
Ada-boost	73.32
LSTM	92.43
SVM	74.32
Existing model	82.3
Proposed model	**94.14**

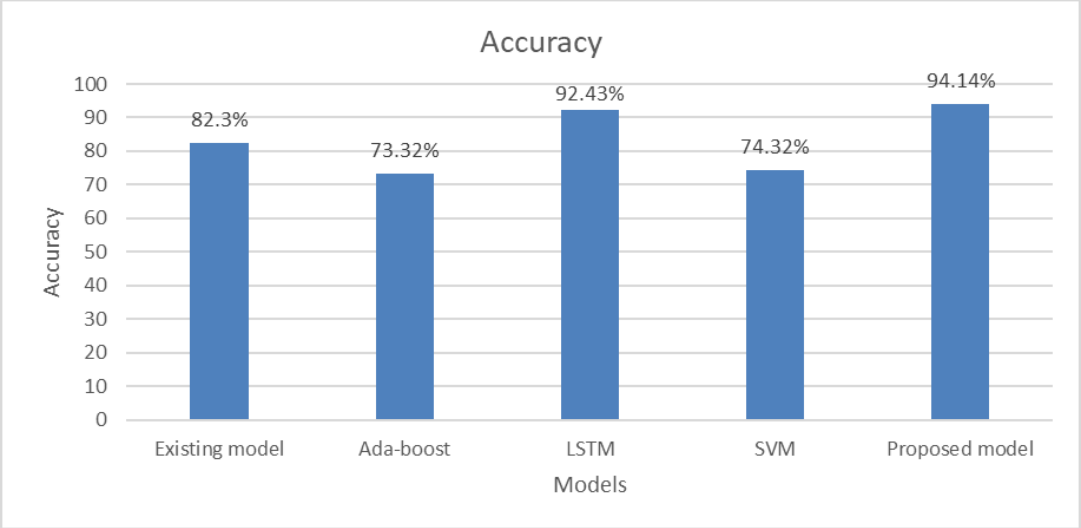

Figure 4.14. Graphical representation of existing models (Jain et al., 2022)

Different Exploratory data analyses have been implemented for the UNSW-NB 15 dataset. Confusion matrix, correlation plot, histograms, boxplot, count plot, and correlation plot are some of the plots that help understand the data effectively.

Comparative analysis helps detect the efficiency and effectiveness of the proposed model compared to the existing models. The analysis detected that the proposed model performed better than the existing models by providing an accuracy rate of 94.14%.

4.6. Summary

The proposed method has incorporated Various performance metrics to evaluate its effectiveness. Additionally, exploratory data analysis (EDA) has been conducted using various plots to gain insights into the data. The results of the proposed method have been compared with existing models through comparative studies, and it has been observed that the proposed method outperforms the existing models, demonstrating superior performance.

5. CONCLUSION AND ADVICES

5.1. Conclusion

Unauthorized access to data poses a significant risk to organizations and individuals, especially in the digital world. To mitigate this risk, intrusion detection systems (IDSs) have been adopted across multiple industries to detect and identify malicious activities and abnormal behavior. An IDS is a network security tool that identifies potential vulnerabilities malicious actors could exploit. It continuously analyzes network traffic and alerts system administrators to any suspicious activity. However, it is important to note that IDSs cannot prevent or remove vulnerabilities that could damage the system.

The main purpose of a network intrusion detection system is to identify potential threats within the network, while firewalls are designed to block and prevent new threats from entering the network. IDSs effectively identify and differentiate potential threats but rely on additional measures to remove them from the network. Organizations can effectively identify anomalies and mitigate unauthorized intrusions into their networks by using IDSs through the deployment of both hardware and software solutions. The primary goal of an IDS is to detect and identify potential security threats, notify system administrators promptly, and take appropriate isolation measures to reduce any potential harm or damage to the system.

In the context of Internet of Things (IoT) systems, IoT devices' storage and computational capacities are often limited, making traditional intrusion detection systems (IDSs) unsuitable or ineffective for IoT environments. Therefore, it is essential to deploy appropriate IDS solutions tailored to IoT environments, safeguarding the confidentiality and integrity of critical data stored in IoT devices. It is worth mentioning that network traffic anomalies can lead to several problems, such as data loss and system privacy breaches. Implementing an IDS is critical to identify and mitigate these potential risks effectively. An IDS can quickly identify various malicious behaviors, including denial-of-service attacks, port scans, and other suspicious network activity. Therefore, an efficient IDS is designed to selectively process known and established attacks and examine the remaining incoming packets for potentially harmful anomalies. The scope of analysis extends beyond incoming packets to include outgoing local traffic, thus enabling comprehensive monitoring. This chapter provides an overview of various types of IDSs, including host-based IDS, network-based IDS, signature-based IDS, and anomaly-based

IDS. IDSs are used to monitor hosts and identify malicious software and unwanted activities that circumvent traditional security measures, such as system calls and files.

Although previous research has used IDSs to identify anomalies and attacks, it is worth noting that these studies have often encountered accuracy issues. The proposed methodology aims to address this particular problem. The primary goal of the proposed method is to identify and classify anomalies efficiently. The proposed approach uses a hybrid architecture consisting of a bidirectional long short-term memory (Bi-LSTM) model combined with convolutional neural networks (CNNs) and rectified linear weights. The proposed methodology seeks to improve the accuracy of anomaly detection and classification in network data by using rectified linear weights instead of conventional weights.

The effectiveness of the proposed model was demonstrated by using various data pre-processing methods on the UNSW-NB15 dataset. The methods used included the examination of missing values, the elimination of outliers, the scaling of features, the encoding of categorical data, and finally, the application of SMOTE sampling. Once the data pre-processing step was completed, the data was split into training and testing datasets. A subset of values from the testing dataset were assigned labels, while the remaining subset remained unassigned. After the training and testing phases were completed, the data was inputted into a model consisting of a Bi-LSTM network with a CNN component.

The Bi-LSTM with CNN model was implemented with several components, including pooling, batch normalization, Bi-LSTM, and dense layers with activation functions such as softmax. In this phase, the rectified linear weight was used to transmit the optimal weights to the Bi-LSTM model, thereby improving the efficiency and effectiveness of anomaly detection. Finally, the proposed model was evaluated using various performance metrics, such as accuracy, precision, and recall. The researchers used the UNSW-NB15 dataset for anomaly detection. To improve the efficiency and effectiveness of the detection process, they used exploratory data analysis (EDA). In addition to conducting exploratory data analysis (EDA), the proposed approaches were subjected to a comparative analysis with established methods, such as the Improved Genetic Algorithm (I-GA), deep belief network, CDBN, XGBoost, REPTree, FFDNN, IGRF-RFE, KS, and other existing methods.

The study demonstrates that the approach presented in this research achieved a high level of accuracy, specifically 94.14%. This finding shows that the proposed model

outperformed the existing models, demonstrating its enhanced effectiveness and proficiency in anomaly detection.

5.2. Advices

Even though IDS has the potential to detect anomalies effectively, it is not very feasible to detect the false alarm rate in large volumes of data. Therefore, in the future, IoT IDS must include features like self-protection, self-configuration, and self-optimization.

Additionally, various methods increase the complexity of network-based attacks, including different network parameters used for staging attacks. Hence, different organizations will implement various solutions to protect their systems and data against vulnerabilities, safeguarding them from unknown and unidentified hackers.

In addition to the existing models, future models will focus more on developing IoT NIDS, which can detect known and unknown attacks without relying on any specific protocol. Therefore, the integration of edge computing methods and fog computing methods will be further explored in future IoT NIDS architectures.

In general, attacks can occur in various streams and different networks. Similarly, technologies like blockchain, which are popular today, are also susceptible to various attacks. Therefore, future work will concentrate on building a secure and reliable framework through blockchain for anomaly-based detection. This framework will help defend and protect IDS nodes against advanced insider attacks.

Sometimes, detecting simple and less dangerous anomalies is easier, while identifying more complex attacks can be challenging. Therefore, future work will involve building a Simple Network Management Protocol (SNMP) to incorporate information from the dataset into Prelude, enabling the detection of larger and more complex attacks. Additionally, mounting more specific modules like Suricata can help identify complex attacks in future work.

Although various attacks are causing unwanted network problems, there is a possibility of different attacks emerging in the future that will test the dependability and reliability of IDS. Hence, in addition to the proposed model, there is a need to develop effective IDS with remarkable accuracy.

6. REFERENCES

Abd Elkhalik W. & Elhenawy, I. (2023). Semi-supervised transformer network for anomaly detection in cellular ınternet of things, *International Journal of Wireless and Ad Hoc Communication,* (4), 56-68.

Abdel-Basset N., Hawash Chakrabortty R, & Ryan J. (2021). Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks, *IEEE Internet of Things Journal,* (8),12251-12265.

Abhale, A. B. (2023). Deep learning perspectives to detecting ıntrusions in wireless sensor networks. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(2s), 18-26.

Agrawal, S., Sarkar, S., Aouedi, O., Yenduri, G., Piamrat, K., Alazab, M.,& Gadekallu, T. R. (2022). Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, 347-359.

Ahanger, T. A., Aljumah, A., & Atiquzzaman, M. (2022). State-of-the-art survey of artificial intelligent techniques for IoT security. *Computer Networks*, *206*, 108771.

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., & Khan, M. S. (2021). Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *EURASIP Journal on Wireless Communications and Networking*, (1), 1-23.

Ahmad, S., Mehfuz, S., & Beg, J. (2021). Enhancing security of cloud platform with cloud access security broker. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT*, 325-335.

Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M. R., & Rodrigues, J. J. (2021). Anomaly detection using deep neural network for IoT architecture. *Applied Sciences*, *11*(15), 7050.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.

Albasheer, H., Md Siraj, M., Mubarakali, A., Elsier Tayfour, O., Salih, S., Hamdan, M., & Kamarudeen, S. (2022). Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. *Sensors*, *22*(4), 1494.

Albulayhi, K., Abu Al-Haija, Q., Alsuhibany, S. A., Jillepalli, A. A., Ashrafuzzaman, M., & Sheldon, F. T. (2022). IoT intrusion detection using machine learning with a novel high performing feature selection method. *Applied Sciences*, *12*(10), 5015.

Alkahtani, H. & Aldhyani H. (2021). Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms, *Complexity*, vol., 1-18.

Aloqaily, M., Otoum, S., Al Ridhawi, I., & Jararweh, Y. (2019). An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, *90*, 101842.

Alsoufi, A., Razak, S., Siraj, M., Nafea, I., & Ghaleb, A., Saeed F. (2021). Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review, *Applied sciences*, (11), 8383.

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions, *Electronics*, (9),1177.

Ashraf, J., Keshk M., Moustafa, N., Abdel-Basset, M., & Khurshid, H. (2021). IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities, *Sustainable Cities and Society*, (72), 1-12.

Awajan, A. (2023). A novel deep learning-based intrusion detection system for IOT networks. *Computers*, *12*(2), 34.

Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A machine learning security framework for iot systems. *IEEE Access*, *8*, 114066-114077.

Bagui, S., Wang, X., & Bagui, S. (2021). Machine learning based intrusion detection for IoT botnet. *International Journal of Machine Learning and Computing*, *11*(6), 399-406.

Bala, R., & Nagpal, R. STATE-OF-ART USING INTRUSION DETECTION SYSTEM.

Barrons_-_March_29_2021, https://www.scribd.com/document/517304803/Barrons-March-29-2021 Erişim Tarihi: 10.03.2023.

Bhati, S., & Khari, M. (2021) A survey on hybrid intrusion detection techniques, in *Research in Intelligent and Computing in Engineering: Select Proceedings of RICE*, 815-825.

Chiba, Z., Abghour, N., Moussaid, K., Lifandali, O., & Kinta, R. (2022). Review of recent ıntrusion detection systems and ıntrusion prevention systems in ıot networks. In *2022 International Conference on Software, Telecommunications and Computer Networks SoftCOM*, 1-6.

Da Costa, A., Papa P., Lisboa, O., Munoz, R., & De Albuquerque, C. (2019). Internet of things: A survey on machine learning-based intrusion detection approaches, *Computer Networks,* (151), 147-157.

Dataset: https://research.unsw.edu.au/projects/unsw-nb15-dataset , Erişim Tarihi: 10.08.2023.

Demirkan, S., Demirkan, I., & McKee, A. (2020). Blockchain technology in the future of business cyber security and accounting. *Journal of Management Analytics*, *7*(2), 189-208.

Deshmukh-Bhosale, S., & Sonavane, S. S. (2020). Design of intrusion detection system for wormhole attack detection in internet of things. In *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 1*, 513-523.

Diwan, T. D., Choubey, S., Hota, H. S., Goyal, S. B., Jamal, S. S., Shukla, P. K., & Tiwari, B. (2021). Feature entropy estimation (FEE) for malicious IoT traffic and detection using machine learning. *Mobile Information Systems*, *2021*, 1-13.

Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet of Things Journal*, *7*(8), 6882-6897.

Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., & Mankodiya, K. (2018). Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare, *Future Generation Computer Systems,* (78), 659-676.

Faysal, J. A., Mostafa, S. T., Tamanna, J. S., Mumenin, K. M., Arifin, M. M., Awal, M. A., & Mostafa, S. S. (2022). XGB-RF: A hybrid machine learning approach for IoT intrusion detection. In *Telecom*, Vol. 3, No. 1, 52-69.

Fenanir, S., Semchedine, F., & Baadache, A. (2019). A Machine learning-based lightweight ıntrusion detection system for the ınternet of things. *Revue d'Intelligence Artificielle*, *33*(3), 203-211.

Ferrag, M. A., Shu, L., Djallel, H., & Choo, K. K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics*, *10*(11), 1257.

Fu, R., Ren, X., Li, Y., Wu, Y., Sun, H., & Al-Absi, M. A. (2023). Machine learning-based UAV assisted agricultural ınformation security architecture and ıntrusion detection. *IEEE Internet of Things Journal*, 1-10.

Gad, A. R., Nashat, A. A., & Barkat, T. M. (2021). Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access*, *9*, 142206-142217.

Ge, M., Syed, N. F., Fu, X., Baig, Z., & Robles-Kelly, A. (2021). Towards a deep learning-driven intrusion detection approach for Internet of Things. *Computer Networks*, *186*, 107784.

Gonçalves, F., Ribeiro, B., Gama, O., Santos, A., Costa, A., Dias, B., & Nicolau, M. J. (2019). A systematic review on intelligent intrusion detection systems for VANETs. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 1-10.

Guezzaz, A., Azrour, M., Benkirane, S., Mohy-Eddine, M., Attou, H., & Douiba, M. (2022). A lightweight hybrid intrusion detection framework using machine learning for edge-based IIoT security. *Int Arab J Inf Technol*, *19*(5), 822-830.

Gupta, B. B., & Quamara, M. (2020). An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience*, *32*(21), e4946.

Gupta, S. K., Tripathi, M., & Grover, J. (2022). Hybrid optimization and deep learning based intrusion detection system. *Computers and Electrical Engineering*, *100*, 107876.

Gyamfi, E., & Jurcut, A. (2022). Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors*, *22*(10), 3744.

Hajiheidari, S., Wakil, K., Badri, M., & Navimipour, N. J. (2019). Intrusion detection systems in the Internet of things: A comprehensive investigation. *Computer Networks*, *160*, 165-191.

Hajj, S., El Sibai, R., Bou Abdo, J., Demerjian, J., Makhoul, A., & Guyeux, C. (2021). Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets, *Transactions on Emerging Telecommunications Technologies, (*32), e4240.

Hamolia, V., Melnyk, V., Zhezhnych, P., & Shilinh, A. (2020). Intrusion detection in computer networks using latent space representation and machine learning. *International Journal of Computing*, *19*(3), 442-448.

Heidari, A., & Jabraeil Jamali, M. A. (2022). Internet of Things intrusion detection systems: A comprehensive review and future directions. *Cluster Computing*, 1-28.

Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., & Bellekens, X. (2020). Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset). In *International networking conference*,73-84.

Ibitoye, O., Shafiq, O., & Matrawy, A. (2019). Analyzing adversarial attacks against deep learning for ıntrusion detection in IoT networks, 1-6.

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, *185*, 115524.

Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., & Cho, G. H. (2021). Towards machine learning based ıntrusion detection in IoT networks. *Computers, Materials & Continua*, *69*(2), 1801-1821. https://doi.org/10.32604/cmc.2021.018466

Jabraeil Jamali, M. A., Bahrami, B., Heidari, A., Allahverdizadeh, P., Norouzi, F., Jabraeil Jamali, M. A., & Norouzi, F. (2020). IoT architecture. *Towards the internet of things: Architectures, security, and applications*, 9-31.

Jain, S., Pawar, P. M., & Muthalagu, R. (2022). Hybrid intelligent intrusion detection system for internet of things. *Telematics and Informatics Reports*, *8*, 100030.

Jyothsna, V. & Prasad, M. (2019). Anomaly-based intrusion detection system,*Computer and Network Security, (*35).

Khattak, H. A., Shah, M. A., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems*, *100*, 144-164.

Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, *4*, 1-27.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics*, *8*(11), 1210.

Kim, S., Hwang, C., & Lee, T. (2020). Anomaly based unknown intrusion detection in endpoint environments, *Electronics*, (9), 1022.

Li, W., Meng, W., & Au, M. (2020). Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments, *Journal of Network and Computer Applications*, vol. 161, 102631

Liu, J. W., Zuo, F. L., Guo, Y. X., Li, T. Y., & Chen, J. M. (2021). Research on improved wavelet convolutional wavelet neural networks. *Applied Intelligence*, *51*, 4106-4126.

Liu, P., Wang, L., Ranjan, R., He, G., & Zhao, L. (2022). A Survey on active deep learning: From model driven to data driven, *ACM Computing Surveys (CSUR)*, (54), 1-34.

Manhas, J., & Kotwal, S. (2021). Implementation of intrusion detection system for internet of things using machine learning techniques. *Multimedia Security: Algorithm Development, Analysis and Applications*, 217-237.

Mishra, S., & Tyagi, A. K. (2019). Intrusion detection in internet of things (IoTs) based applications using blockchain technolgy. In *2019 third international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, 123-128.

Mohan Krishna, A., & Amit Kumar, T. (2020). Intrusion detection in intelligent transportation system and its applications using blockchain technology, *international conference on emerging trends in information technology and engineering (IC-ETITE)*, 1-8.

Mokbal, F. M. M., Wang, D., Osman, M., Yang, P., & Alsamhi, S. H. (2022). An efficient intrusion detection framework based on embedding feature selection and ensemble learning technique. *Int. Arab J. Inf. Technol.*, *19*(2), 237-248.

Noor, U., Anwar, Z., Amjad, T., & Choo, K. K. R. (2019). A machine learning-based fintech cyber threat attribution framework using high-level indicators of compromise. *Future Generation Computer Systems*, *96*, 227-242.

Om Kumar, C. U., Marappan, S., Murugeshan, B., & Beaulah, P. M. R. (2023). Intrusion detection model for IOT using recurrent kernel convolutional neural network. *Wireless Personal Communications*, *129*(2), 783-812.

Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: A deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3), e3803.

Pratum. (2023). Why intrusion detection and prevention systems are still important.

Raghuvanshi, A., Singh, U. K., Sajja, G. S., Pallathadka, H., Asenso, E., Kamal, M., & Phasinam, K. (2022). Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. *Journal of Food Quality*, *2022*, 1-8.

Rashid, M. M., Sabrina, F., Ray, B., Morshed, A., Gordon, S., & Wibowo, S. (2022). Anomaly detection in IoT applications using deep learning with class balancing. In *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering CSDE*, 1-6.

Reddy, D. K. K., Nayak, J., & Behera, H. S. (2022). A Hybrid semi-supervised learning with nature-ınspired optimization for ıntrusion detection system in IoT environment. In *International Conference on Computational Intelligence in Pattern Recognition*, 580-591.

Roy, S., Li, J., Choi, B. J., & Bai, Y. (2022). A lightweight supervised intrusion detection mechanism for IoT networks. *Future Generation Computer Systems*, *127*, 276-285.

Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, *99*, 107810.

Said, A. M., Yahyaoui, A., & Abdellatif, T. (2021). Efficient anomaly detection for smart hospital IoT systems. *Sensors*, *21*(4), 1026.

Said, A. M., Yahyaoui, A., Yaakoubi, F., & Abdellatif, T. (2020). Machine learning based rank attack detection for smart hospital infrastructure. In *The Impact of Digital Technologies on Public Health in Developed and Developing Countries: 18th International Conference, ICOST 2020, Hammamet, Tunisia, June 24–26, 2020, Proceedings 18*, 28-40.

Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., & Portmann, M. (2022). Feature extraction for machine learning-based intrusion detection in IoT networks. *Digital Communications and Networks*, 1-15.

Shareena, J., Ramdas, A., & AP, H. (2021). Intrusion detection system for iot botnet attacks using deep learning. *SN Computer Science*, *2*(3), 1-8.

Sharma, R., & Athavale, A. (2019). Survey of intrusion detection techniques and architectures in wireless sensor networks, *International Journal of Advanced Networking and Applications*, (10), 3925-3937.

Smys, S., Basar, A., & Wang, H. (2020). Hybrid intrusion detection system for internet of things (IoT). *Journal of ISMAC*, *2*(04), 190-199.

Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., & Sakurai, K. (2020). Machine learning-based IoT-botnet attack detection with sequential architecture. *Sensors*, *20*(16), 4372.

Spadaccino, P., & Cuomo, F. (2020). Intrusion detection systems for IoT: Opportunities and challenges offered by Edge Computing and Machine Learning. *arXiv preprint arXiv:2012.01174*.

Sudarshan, K. G. (2019). Smart agriculture monitoring and protection system using IOT, *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES* vol. 2, no. 12, 308-310.

Sugi and Ratna., S., R. (2020). Investigation of machine learning techniques in intrusion detection system for IoT network, 1164-1167.

Susilo, B., & Sari, R. F. (2020). Intrusion detection in IoT networks using deep learning algorithm. *Information*, *11*(5), 279.

Thamilarasu, G., & Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, *19*(9), 1977.

Thamilarasu, G., Odesile, A., & Hoang, A. (2020). An intrusion detection system for internet of medical things. *IEEE Access*, *8*, 181560-181576.

Tharewal, S., Ashfaque, M. W., Banu, S. S., Uma, P., Hassen, S. M., & Shabaz, M. (2022). Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. *Wireless Communications and Mobile Computing*, 1-8.

Tsimenidis, S., Lagkas, T., & Rantos, K. (2022). Deep learning in IoT intrusion detection. *Journal of network and systems management*, *30*, 1-40.

Tyagi, H., & Kumar, R. (2021). Attack and anomaly detection in IoT networks using supervised machine learning approaches. *Revue d'Intelligence Artificielle*, *35*(1).

Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access*, *9*, 103906-103926.

Vaigandla, K., Azmi, N., & Karne, R. (2022). Investigation on ıntrusion detection systems (IDSs) in IoT, *International Journal of Emerging Trends in Engineering Research, (*10).

Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, *111*, 2287-2310.

Verma, P., Dumka, A., Singh, R., Ashok, A., Gehlot, A., Malik, P. K., & Hedabou, M. (2021). A novel intrusion detection approach using machine learning ensemble for IoT environments. *Applied Sciences*, *11*(21), 10268.

Vitorino, J., Andrade, R., Praça, I., Sousa, O., & Maia, E. (2021). A comparative analysis of machine learning techniques for iot intrusion detection. In *International Symposium on Foundations and Practice of Security*, 191-207.

Wu, W., Li, R., Xie, G., An, J., Bai, Y., Zhou, J., & Li, K. (2019). A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 919-933.

Yaacoub J, P., Noura H., Salman, O. & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations, *Internet of Things*, (11), 100218.

Yasmeen, S. A., Bader, A. & Amr, M. (2022). Network intrusion detection using machine learning techniques, *Advances in Science and Technology Research Journal* vol. 16, no. 3, 193-206.

Yeboah-Ofori, A., & Islam, S. (2019). Cyber security threat modeling for supply chain organizational environments. *Future Internet*. 11(3):63. https://doi.org/10.3390/fi11030063

Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *Journal of Big Data*, *10*(1), 1-26.

Zainel, H., & Koçak, C. (2022). LAN intrusion detection using convolutional neural networks. *Applied sciences*, *12*(13), 6645.

Zhong, M., Zhou, Y., & Chen, G. (2021). Sequential model based intrusion detection system for IoT servers using deep learning methods. *Sensors*, *21*(4), 1113.

CURRICULUM VITAE

STUDENT INFORMATION	
Name Surname:	Sindibad Ali Fayyadh Fayyadh
Nationality:	Iraqi
Orcid No:	0000-0001-9729-5965

SCHOOL INFORMATION	
Undergraduate Study	
University:	Al-Rafidain University College
Faculty:	
Department:	Computer Science
Graduation Year:	2001
Graduate Study	
University:	Kırşehir Ahi Evran University
Institute:	Institute of Natural and Applied Sciences
Department:	Advanced Technologies
Graduation Year:	2023

Articles and Papers Produced from the Thesis
Fayyadh, S. A. F., & Kayabaş, A. (2023). Semi-Supervised learning for ıntrusion detection in IOT networks with UNSW_NB15 dataset. *International Journal of Advances in Engineering and Emerging Technology*, 14(1), 229-243.