IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 3, Ser. 3 (May. – June. 2024), PP 01-18
www.iosrjournals.org

Scalable Intrusion Detection Systems For Big Data Environments

Vaghani Divyeshkumar[1]

[1]*gannon University, 109 University Square, Erie, Pa 16541, Usa*

Abstract

Background: *This study proposes scalable intrusion detection systems suitable for big data environments. The objectives of the study are; to address the limitations of traditional IDS methods and providing a more robust defense against advanced attacks in the context of large-scale data systems; and to develop an intrusion detection system capable of effectively detecting and mitigating security threats within big data environments.*

Materials and Methods: *The proposed systems include; a CNN-based Intrusion Detection System, a LSTM-based Intrusion Detection System, and a GAN-based Intrusion Detection System, the study utilized the CICIDS2017 dataset to investigate the efficacy of these three prominent deep learning architectures.*

Results: *The results reveals that the GAN model outperformed the LSTM and CNN models across various metrics such as precision, recall, accuracy, and F1-score. The GAN model exhibited high performance in accurately identifying and classifying intrusions within network traffic data, showcasing its ability to capture complex patterns and generate synthetic traffic data resembling real-world scenarios. While slightly trailing the GAN model, the LSTM model also demonstrated robust performance, leveraging its capability to detect long-term dependencies in sequential data. Despite slightly lower performance compared to the GAN and LSTM models, the CNN model exhibited reliable intrusion detection capabilities.*

Conclusion: *This research underscores the importance of leveraging deep learning techniques for IDS development in big data environments.*

Keywords: *attacks, CNN, data systems, defense, deep learning, GAN, LSTM, security threats.*

Date Of Submission: 21-05-2024 Date Of Acceptance: 31-05-2024

I. Introduction

In recent years, the capability to extract, transform, and load massive amounts of data from diverse sources has revolutionized various industries through big data systems. This advancement has transformed decision-making processes, enabling organizations to gather valuable insights and optimize their operations. However, the advent of big data also brings significant challenges, particularly regarding privacy and security. Attackers have developed advanced methods capable of crippling entire big data environments within minutes. Each year sees new records breached by attackers. A recent destructive DDoS attack disrupted more than 70 critical Internet services, including Github, Twitter, Amazon, and Paypal. Attackers have exploited big data environments and Internet of Things technologies to generate enormous volumes of attack traffic, exceeding 665 Gb/s[1,2].

Big data's vast scale, rapid velocity, and heterogeneous nature pose significant challenges for traditional intrusion detection systems (IDS) in identifying and mitigating security risks[3,4]. Intrusion detection systems are vital for protecting computer networks and systems from intrusions, attacks, and anomalies. Traditional IDS solutions typically rely on rule-based or signature-based techniques to identify known attack patterns using predefined rules or patterns[5,6]. However, these conventional methods often struggle to keep pace with the rapidly evolving cyber threat landscape and are ill-suited to the unique characteristics of big data environments. Deep learning, a subset of machine learning, has emerged as a powerful tool for handling complex and extensive data analysis tasks[7]. It has demonstrated exceptional performance in various domains, including speech recognition, natural language processing, and image recognition. Deep learning algorithms like CNNs and RNNs excel at automatically learning intricate patterns and features from large datasets, making them a promising approach for intrusion detection in big data contexts[8].

This research study proposes an intrusion detection system tailored for big data environments to enhance the accuracy and efficiency of detecting intrusions. By leveraging deep learning's capabilities, we aim to address the limitations of traditional IDS techniques and provide more effective protection against sophisticated attacks in large data systems. Our goal is to develop an intrusion detection system that can effectively identify and mitigate security threats in big data environments by addressing these critical factors[9]. The proposed solution has the potential to bolster the overall security posture of organizations by ensuring the integrity, confidentiality, and availability of their valuable data assets. In the following sections of this research study, we will delve deeper into the existing literature on intrusion detection systems, big data environments, and deep learning in the context of security. We will discuss the shortcomings of current approaches and outline our design process.

Objectives of the Research

The primary aim of this study is to suggest scalable intrusion detection systems suitable for big data environments. Additional objectives of this research include:

i. Addressing the limitations of traditional IDS methods and providing a more robust defense against advanced attacks in the context of large-scale data systems.
ii. Developing an intrusion detection system capable of effectively detecting and mitigating security threats within big data environments.

II. Literature Review

Intrusion Detection Systems

Intrusion detection systems (IDSs) are software or hardware solutions that automate the monitoring of events within a computer system or network, analyzing these events for potential security issues. With the rise in both frequency and severity of network attacks in recent years, IDSs have become essential components of most organizations' security infrastructures[10].

Intrusion detection involves observing the activities within a computer system or network and analyzing them for signs of intrusions, which are attempts to compromise the system's confidentiality, integrity, availability, or to bypass its security mechanisms. Intrusions can be perpetrated by external attackers accessing systems via the Internet, authorized users attempting to gain unauthorized privileges, or authorized users misusing their given privileges. IDSs automate this monitoring and analysis process[11].

By employing intrusion detection, organizations can safeguard their systems against the threats posed by increased network connectivity and dependency on information systems. Considering the complexity of modern network security threats, security professionals should focus on determining which intrusion detection features and capabilities to implement rather than questioning the necessity of intrusion detection itself[12].

According to Modi et al.[13], IDSs have become widely accepted as crucial additions to organizational security infrastructures. Despite the proven benefits of intrusion detection technologies in enhancing system security, many organizations still require justification for the acquisition of IDSs. There are several persuasive reasons to adopt and utilize IDSs:

1. To discourage problematic behaviors by increasing the perceived risk of detection and punishment for those attempting attacks or system abuse.

2. To identify attacks and security breaches that other security measures may not prevent.

3. To detect and manage the preliminary stages of attacks, often seen as network probes and other probing activities.

4. To document existing threats to an organization.

5. To serve as a quality control measure for security design and administration, particularly in large and complex environments.

6. To provide valuable information about any intrusions that occur, aiding in improved diagnosis, recovery, and correction of underlying issues.

IDS Analysis

There are two main methods for analyzing events to identify attacks: misuse detection and anomaly detection. Misuse detection, which targets known malicious activities, is the approach most commonly used by commercial systems. Anomaly detection, which seeks out unusual patterns of behavior, remains a significant focus of ongoing research and is utilized to a limited extent by various IDSs. Each method has its own advantages and disadvantages, and the most effective IDSs tend to combine misuse detection with some elements of anomaly detection[14].

Misuse Detection

Misuse detection involves analyzing system activities to find events or sets of events that match predefined patterns indicative of known attacks. These patterns, known as signatures, make misuse detection synonymous with "signature-based detection." The typical commercial misuse detection system uses a unique signature for each known attack pattern. However, more advanced techniques, known as "state-based" analysis, can use a single signature to identify multiple related attacks[15].

Advantages:

- Misuse detectors are highly effective at identifying attacks without producing a large number of false positives.
- They can quickly and accurately identify specific attack tools or techniques, aiding security managers in prioritizing response actions.
- Misuse detectors enable system managers, regardless of their security expertise, to monitor and manage security issues and initiate incident response procedures.

Disadvantages:

- Misuse detectors can only identify attacks they have signatures for, necessitating constant updates with new attack signatures.
- Many misuse detectors use highly specific signatures, limiting their ability to detect variations of common attacks. While state-based misuse detectors can mitigate this issue, they are not widely used in commercial IDSs.

Anomaly Detection

Anomaly detectors identify unusual or abnormal behavior on a host or network. They operate on the premise that attacks differ from "normal" (legitimate) activity and can thus be detected by systems that recognize these discrepancies. Anomaly detectors create profiles representing the typical behavior of users, hosts, or network connections, using historical data gathered during periods of normal operation. They then monitor current events and apply various measures to determine when activities deviate from these established norms[16].

The techniques and measures used in anomaly detection include:

- **Threshold detection:** Certain attributes of user and system behavior are quantified, with permissible levels set. Attributes might include the number of files a user accesses in a certain timeframe, failed login attempts, CPU usage by a process, etc. These levels can be static or heuristic, adapting based on observed values over time.
- **Statistical measures:** These can be parametric, assuming the distribution of profiled attributes fits a specific pattern, or non-parametric, where the distribution is learned from historical data.
- **Rule-based measures:** Similar to non-parametric statistical measures, these define acceptable usage patterns as rules instead of numerical quantities.
- **Other measures:** Including neural networks, genetic algorithms, and immune system models.

Currently, commercial IDSs primarily use the first two measures.

Advantages:

Anomaly-based IDSs can detect unusual behavior, identifying symptoms of attacks without prior knowledge of specific details.

Information from anomaly detectors can help define signatures for misuse detectors.

Disadvantages:

Anomaly detectors often generate a high number of false alarms due to the unpredictable nature of user and system behavior.

They require extensive training datasets of system event records to accurately characterize normal behavior patterns.

Despite the frequent false alarms, researchers argue that anomaly-based IDSs can identify new forms of attacks, unlike signature-based IDSs that depend on known attack patterns. Additionally, some forms of anomaly detection can feed data into misuse detectors. For instance, a threshold-based anomaly detector might identify the typical number of files a user accesses, which a misuse detector could then use to trigger an alarm if this number is exceeded by a certain percentage[17].

Although some commercial IDSs incorporate limited anomaly detection, few, if any, rely solely on this method. The anomaly detection present in commercial systems typically focuses on identifying network or port scanning activities. Nevertheless, anomaly detection remains an active area of research and may play a more significant role in future IDSs[18].

Strengths and Limitations of IDSs

While Intrusion Detection Systems (IDSs) are essential components of an organization's security setup, they have specific strengths and limitations. When developing your organization's security strategy, it's crucial to recognize what IDSs can be relied upon to do and which objectives might be better addressed by other security mechanisms[19].

Strengths of Intrusion Detection Systems

According to Varadharajan and Tupakula[19], IDSs excel in the following areas:

- Monitoring and analyzing system events and user behaviors.
- Assessing the security status of system configurations.
- Establishing a baseline security state for a system and tracking any deviations.
- Identifying patterns of system events that match known attack signatures.
- Detecting activity patterns that statistically differ from normal behavior.
- Managing operating system audit and logging mechanisms, along with the data they produce.
- Alerting relevant personnel through appropriate channels when attacks are detected.
- Measuring the enforcement of security policies embedded in the analysis engine.
- Providing default information security policies.
- Enabling non-security experts to perform essential security monitoring tasks.2.3.1 Limitations of Intrusion Detection Systems

Varadharajan and Tupakula[19] also point out the limitations of IDSs, including their inability to:

Compensate for weak or absent security mechanisms within the protection infrastructure, such as firewalls, identification and authentication systems, link encryption, access control mechanisms, and virus detection and removal.

Instantly detect, report, and respond to attacks during periods of heavy network or processing load.

Identify newly disclosed attacks or variations of known attacks.

Effectively counter sophisticated attackers.

Conduct automatic investigations of attacks without human involvement.

Withstand attacks designed to bypass or disable them.

Make up for issues with the accuracy of information sources.

Handle switched networks efficiently.

Types of Computer Attacks Commonly Detected by IDSs

IDSs commonly detect three types of computer attacks: system scanning, denial of service (DOS), and system penetration. These attacks can originate either locally, targeting the attacked machine, or remotely, using a network to access the intended target. Understanding the distinctions among these attack types is crucial for IDS operators, as each necessitates a distinct response[20].

Scanning Attacks

A scanning attack occurs when an attacker probes a network or system by transmitting various types of packets. While the techniques employed may be similar, the underlying motives driving these activities differ significantly.

By analyzing the responses received from the target, the attacker can glean insights into the system's characteristics and vulnerabilities. Consequently, a scanning attack serves as a reconnaissance tool for attackers, aiding in target identification. Scanning attacks do not breach or compromise systems directly. Common names for the tools used in these activities include network mappers, port mappers, network scanners, port scanners, or vulnerability scanners[3]. Scanning attacks can provide information such as:

- The network topology of the target
- The firewall's permissible network traffic
- Active hosts within the network
- The operating systems and server software versions running on those hosts
- Specific vulnerabilities in hosts identified by vulnerability scanners[4].

Armed with this knowledge, an attacker can pinpoint vulnerable systems on the target network and identify specific attack vectors for penetration attempts. Consequently, attackers employ scanning software to surveil a target before launching actual attacks. Unfortunately, while it may be legal for individuals to survey visible security measures in a bank, some legal experts argue that scanning a host or network may also be legal. To a scanner, such activities appear as lawful exploration of publicly available resources[5].

Legitimate reasons for scanning activities exist, such as web search engines indexing new web pages or individuals searching for publicly accessible content like music repositories or multiplayer games. Essentially, the same technology enabling the discovery of public resources also facilitates security analysis of systems, as seen in vulnerability assessment tools. Effective IDS signatures for malicious scanning can discern between legitimate and malicious scanning activities. Scanning is likely the most prevalent attack type, serving as a precursor to serious penetration attempts. If your network is connected to the Internet, it is highly probable that it undergoes scanning, if not daily, at least several times a week[6].

Denial of Service Attacks

Denial of Service (DOS) attacks aim to impede or halt targeted network systems or services. These attacks are prevalent in certain online communities, where they are often utilized during verbal disputes on platforms like Internet Relay Chat. While DOS attacks are sometimes employed for trivial reasons, they can also inflict significant damage on major organizations. Notably, well-publicized incidents have attributed substantial losses to electronic commerce operations, whose customers were unable to access their services for purchases. DOS attacks typically fall into two categories: flaw exploitation and flooding. It is essential for an IDS operator to distinguish between these types[7].

Flaw Exploitation DOS Attacks

Flaw exploitation attacks capitalize on vulnerabilities in the target system's software to induce processing failures or deplete system resources. For instance, the 'ping of death' attack involved sending unusually large ping packets to specific Windows systems, causing them to crash. Resource exhaustion attacks target CPU time, memory, disk space, buffer space, or network bandwidth. Often, patching the software can thwart this type of DOS attack[8].

Flooding DOS Attacks

Flooding attacks inundate a system or its components with an excessive volume of information, overwhelming its processing capabilities. Even if the attacker cannot supply enough information to overload the system, they may monopolize the network connection, denying others access to the resource. Since there is no inherent flaw in the target system, patching is not a viable solution against flooding attacks. Consequently, these attacks pose significant challenges and concerns for organizations. Although there are few universal remedies for preventing flooding attacks, the target can implement various technical adjustments to mitigate their impact[9].

The term "distributed DOS" (DDOS) falls within the realm of DOS attacks. DDOS attacks are essentially flooding DOS attacks orchestrated using multiple computers under the control of a central attacker. By coordinating thousands of compromised hosts, an attacker can effectively overwhelm even the most robust systems, rendering them inoperative[21].

Penetration Attacks

Penetration attacks involve the unauthorized acquisition and/or modification of system privileges, resources, or data. These actions violate integrity and control, in contrast to DOS attacks, which target resource availability, and scanning attacks, which do not involve illegal activities. Various software vulnerabilities can be exploited to gain control of a system[22]. Below are the most common vulnerabilities and their security implications.

While penetration attacks exhibit significant variation in their specifics and impact, the primary types include:

User to Root: A local user on a host gains full control of the target host.

Remote to User: An attacker on the network gains access to a user account on the target host.

Remote to Root: An attacker on the network gains complete control of the target host.

Remote Disk Read: An attacker on the network gains unauthorized access to read private data files on the target host.

Remote Disk Write: An attacker on the network gains unauthorized access to write to private data files on the target host.

Remote vs. Local Attacks

DOS and penetration attacks manifest in two forms: local and remote.

Authorized User Attack:

Authorized user attacks originate from a legitimate user account on the target system. These attacks often involve some form of privilege escalation.

Public User Attack:

Public user attacks, conversely, are initiated without any user account or privileged access to the target system. They are executed remotely via a network connection, utilizing only the public access granted by the target[9].

A common attack strategy involves initiating a public user attack to gain initial system access. Subsequently, once inside the system, the attacker employs authorized user attacks to seize complete control of the target[23].

Determining Attacker Location from IDS Output

In notifications of a detected attack, IDSs will often report the location of a attacker. This location is most commonly expressed as an source IP address. The reported address is simply the source address that appears in the attack packets. As attackers routinely change IP addresses in attack packets, this does not necessarily represent the true source address of the attacker[24].

The key to determining the significance of the reported source IP address is to classify the type of attack and then determine whether or not the attacker needs to see the reply packets sent by the victim.

If the attacker launches a one-way attack, like many flooding DOS attacks, where the attacker does not need to see any reply packets, then the attacker can label his packets with random IP addresses. The attacker is doing the real world equivalent of sending a postcard with a fake return address to fill a mailbox so that no other mail can fit into it. In this case, the attacker cannot receive any reply from the victim[25].

However, if the attacker needs to see the victim's replies, which is usually true with penetration attacks, then the attacker usually cannot lie about his source IP address. Using the postcard analogy, the attacker needs to know that his postcards got to the victim and therefore must usually label his postcards with his actual address[26].

In general, attackers must use the correct IP address when launching penetration attacks but not with DOS attacks.

However, there exists one caveat when dealing with expert attackers. An attacker can send attack packets using a fake source IP address, but arrange to wiretap the victims reply to the faked address. The attacker can do this without having access to the computer at the fake address. This manipulation of IP addressing is called "IP Spoofing."

IDSs and Excessive Attack Reporting

Numerous IDS operators face overwhelming volumes of attack reports generated by IDSs. It becomes practically infeasible for an operator to scrutinize the hundreds or even thousands of daily reports from some IDSs. The crux of the issue lies not in the quantity of attacks but rather in how IDSs relay these incidents[27].

Certain IDSs generate a distinct attack report each time an attacker accesses a different host. Consequently, scanning a subnet of a thousand hosts could trigger a thousand individual attack reports. Some vendors propose a remedy for this predicament. Their latest IDSs are now adept at consolidating redundant entries, prioritizing the most critical attacks for operator review[28].

Attack Naming Conventions

Until recently, there lacked a universal naming convention for computer attacks or vulnerabilities. This lack of standardization posed challenges in comparing the efficacy of different IDSs, as each vendor's IDS generated distinct results when analyzing events corresponding to the same set of attacks. It also hindered the coordination of multiple IDS types within a network, given their disparate messages upon detecting identical attacks[29,30].

Thankfully, ongoing efforts within the network security community aim to establish a common nomenclature for computer vulnerabilities and attacks. Foremost among these initiatives is the Common Vulnerabilities and Exposures List (CVE), managed by MITRE with input from global security experts. Many network security product vendors commit to making their products CVE-compatible[31].

Attack Severity Levels

Numerous IDSs assign severity levels to detected attacks to assist operators in accurately gauging the attack's impact, thereby facilitating appropriate responses. However, assessing the impact and severity of an attack is subjective and not necessarily synonymous, contingent upon the target network and organizational environment. For instance, a highly effective Unix attack targeting a large heterogeneous network may have a low impact on a Windows-based network segment but remains severe for the entire network. Hence, the severity levels reported by IDSs offer valuable insights for security managers but necessitate contextual consideration within the specific system environment[32].

Empirical Studies

Deep learning solutions have been extensively explored in various studies to explore alternative approaches to intrusion detection within big data environments. One such approach is the DeepIDS IDS framework, tailored specifically for massive data scenarios, which leverages deep learning techniques[21]. DeepIDS utilizes deep belief networks along with stacked denoising autoencoders to extract intricate features from network traffic data, subsequently feeding these features into a support vector machine classifier for

intrusion detection. Compared to traditional methods, DeepIDS exhibits superior detection accuracy and adeptly handles the intricacies of big data environments.

Another innovative approach introduced in recent research is a hybrid deep learning model combining CNNs and GRUs for intrusion detection in large-scale data systems[9]. This model incorporates GRUs to capture temporal dependencies and CNNs to discern spatial patterns within network traffic data. The hybrid model outperforms conventional machine learning techniques in both accuracy and efficiency, achieving commendable detection rates.

In addressing intrusion detection challenges in big data settings, Manimurugan et al.[23] propose a novel Deep-NN anomaly detection (ADT) technique. This technique involves learning representations of typical network traffic through unsupervised learning employing Restricted Boltzmann Machines (RBMs). By assessing the reconstruction error derived from the RBMs, the method effectively identifies anomalies within big data systems.

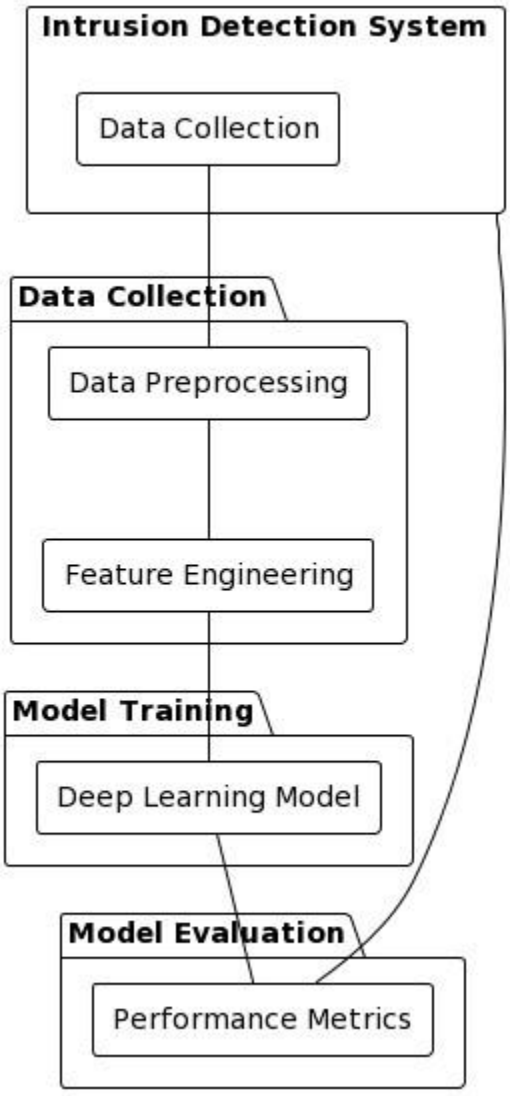

Figure 1: Deep Learning based IDS System

In a recent study by NG and Selvakumar[24], the utilization of Generative Adversarial Networks (GANs) for intrusion detection within big data settings was proposed. The GANIDS framework comprises a generator network trained on the typical distribution of network traffic data, alongside a discriminator network tasked with distinguishing between typical and abnormal traffic. Demonstrating adaptability to evolving attack patterns, the GANIDS technique exhibited promising outcomes in detecting both known and unforeseen threats.

Another noteworthy contribution comes from Sharafaldin et al.[25], who introduced DeepLog, a system log diagnosis tool not explicitly tailored for big data environments. DeepLog employs LSTM networks to identify anomalous patterns and capture sequential dependencies within log data. This method showcased remarkable accuracy in identifying abnormal behavior and offered valuable insights for system diagnosis.

Proposing a hybrid solution for intrusion detection in big data systems, Ferrag et al.[26] introduced the DBSCAN-DNN technique, which combines the DBSCAN algorithm with DNNs. The DBSCAN technique is utilized to discover clusters within network traffic data, which are subsequently inputted into the DNN model for classification. This hybrid approach exhibited commendable detection rates and effectively addressed challenges inherent in large data environments, such as high dimensionality and data variability.

Furthermore, Chawla et al.[27] presented a deep learning architecture for intrusion detection in big data environments, integrating LSTM networks and CNNs. The CNN component extracts geographical patterns, while the LSTM component captures temporal dependencies within network traffic data. This hybrid LSTM-CNN model surpassed conventional machine learning techniques in both effectiveness and scalability, thereby enhancing detection accuracy.

In a recent study by Samek et al.[28], an autoencoder-based approach for anomaly detection in intrusion detection systems for big data systems was introduced. Leveraging unsupervised learning, autoencoders were utilized to reconstruct typical network traffic data. By assessing the reconstruction error between input and reconstructed data, anomalies were detected. This method effectively identified previously unidentified attacks and demonstrated resilience to changes in network traffic data.

Additionally, Liu et al.[29] proposed an attention-based DL-IDS model specifically designed for huge data environments. Employing self-attention techniques, the model determined the importance of various network traffic aspects, enabling it to focus on information critical for intrusion detection. The attention-based deep learning model exhibited improved detection accuracy and robustness in the presence of noisy or redundant features.

Furthermore, Deng et al.[30] presented a Gated Attention Network (GAN) for intrusion detection in massive data systems. The GAN model utilized attention techniques to dynamically assign weights to different characteristics of network traffic data. The gated mechanism allowed the model to selectively attend to relevant input for intrusion detection. In massive data contexts, the GAN technique demonstrated superior detection performance and adaptability to changing attack patterns.

These recent advancements underscore the increasing utilization of deep learning techniques, as highlighted by Albawi et al.[31], including hybrid architectures, attention mechanisms, and unsupervised learning models, for intrusion detection in big data scenarios. They offer insightful solutions to the challenges posed by the size, complexity, and variability of data in such systems. Further research in this field is crucial to enhance the precision, effectiveness, and scalability of intrusion detection systems in large data environments.

In a study by Gu et al.[32], the efficacy of deep reinforcement learning (DRL) for intrusion detection in big data systems was explored. Utilizing a deep Q-network (DQN), the proposed method learned optimal strategies for making judgments about Network traffic-based IDS. The DRL-based technique exhibited promising results in identifying intricate attacks and offered potential for adaptive and dynamic intrusion detection in evolving big data environments.

Furthermore, Yu et al.[33] introduced the use of Graph Neural Networks (GNNs) for intrusion detection in massive data networks. GNNs enabled the modeling of complex interactions and relationships among network components. Leveraging GNNs, the proposed method learned representations that accurately captured attack patterns and preserved the graph structure of network traffic data. The GNN-based approach showcased superior detection accuracy and adeptness in managing vast, dynamic big data networks.

Moreover, Sherstinsky[34] proposed a hybrid deep learning model merging CNNs and LSTM networks with transfer learning for intrusion detection in big data scenarios. Transfer learning facilitated the adaptation of pre-trained models on extensive datasets to the specific intrusion detection task. The hybrid model exhibited enhanced detection capabilities and shorter training times, making it suitable for real-time intrusion detection in big data systems.

Additionally, Tschannen et al.[35] focused on applying multi-objective evolutionary algorithms to optimize intrusion detection in huge data environments. The strategy aimed to concurrently optimize several objectives, including computational efficiency, false positive rate, and detection accuracy. Through a Pareto-based approach, the proposed optimization system effectively balanced various trade-offs in intrusion detection performance and provided decision-makers with a range of optimal solutions.

Furthermore, Meng et al.[36] and Chen et al.[37] investigated the application of federated learning for intrusion detection in big data systems while preserving user privacy. Federated learning enabled joint model training using diverse distributed data sources without compromising privacy. By utilizing federated learning to train intrusion detection models with local data from multiple sources, the proposed strategy ensured data privacy.

These recent advancements highlight the suitability of deep learning methods for detecting intrusions in massive data environments. They also demonstrate the diverse approaches and methodologies employed to enhance intrusion detection in large data environments through deep learning. Researchers continue to explore novel methodologies to address the unique challenges of intrusion detection in the context of big data systems.

Table 1: Analysis of Existing IDS Systems

Approach	Key Features	Advantages	Limitations	Dataset Used
DeepIDS[8]	Stacked autoencoders, SVM classifier	Improved detection accuracy	High computational complexity	NSL-KDD
CNN-GRU[9]	Hybrid CNN and GRU architecture	High detection rates, efficient	High training time	CICIDS2017
DNN-AD[21]	RBMs for unsupervised feature learning	Effective detection of anomalies	Sensitivity to hyperparameters	UNSW-NB15
GANIDS[22]	Generative Adversarial Networks (GANs)	Detects both known and unknown attacks	Difficulty in training GANs	CICIDS2017
DBSCAN-DNN[38]	DBSCAN clustering, DNN classification	High detection rates, handles variability	Difficulty in determining DBSCAN's eps	UNSW-NB15
LSTM-CNN[23]	LSTM and CNN hybrid architecture	Improved accuracy and efficiency	Difficulty in capturing long dependencies	NSL-KDD
Autoencoder-Based[24]	Autoencoder reconstruction for anomaly	Effective detection of unknown attacks	Sensitive to selection of reconstruction error threshold	UNSW-NB15

Table 2: Publically Available Datasets

Dataset	Description	Size	Number of Features	Attack Types	Year
NSL-KDD	Network traffic data	1.8 GB	41	Multiple	2009
CICIDS2017	Network traffic data	256 GB	79	Multiple	2017
UNSW-NB15	Network traffic data	1.9 GB	49	Multiple	2015
KDD Cup 1999	Network traffic data	743 MB	41	Multiple	1999
DARPA1999	Network traffic data	2.8 GB	41	Multiple	1999
ISCXIDS2012	Network traffic data	3.8 GB	79	Multiple	2012
NSL-KDD+	Network traffic data	2.3 GB	41	Multiple	2009
CIDDS-001	Network traffic data	3.7 GB	48	Multiple	2018
ADFA-LD	Windows system logs	155 MB	N/A	Normal and Anomalous	2015
SADL	Sensor anomaly detection logs	N/A	N/A	Anomalous	2014

Roschke et al.[12] proposed a conventional IDS system called VM-Integrated IDS, which is based on Snort, to identify anomalies. The suggested IDS features an extensible architecture comprising multiple sensors and a central management unit. By leveraging system-level virtualization technology and VM monitoring approaches, the IDS can effectively handle VM-based IDSs, thereby facilitating seamless integration into Cloud architectures.

Likewise, Modi et al.[13] utilized Snort and machine learning classifiers to detect anomalies in network traffic among VMs. Their framework integrates a network intrusion detection system (NIDS) into the Cloud infrastructure, employing Snort and a decision tree (DT) classifier. Primarily focusing on monitoring network traffic, the system aims to identify network attacks in the Cloud. Validation of the system was conducted using the NSL-KDD and KDD datasets.

Gupta et al.[14] introduced a novel IDS for Cloud environments based on an immediate Syscall signature structure to identify malicious program executions. This system exhibits efficiency in terms of complexity and resource consumption. Evaluation of the system was performed in a private Cloud environment based on Open Nebula and Virtual Box, utilizing various datasets from the University of New Mexico (UNM), resulting in high accuracy for detecting several attacks.

In Li et al.'s study[15], Artificial Neural Network (ANN) was employed to detect attacks in the cloud. They proposed a distributed neural network-based IDS with an adaptive architecture to prevent overloading Cloud VMs. Leveraging ANN enables the IDS to identify new attack types with relatively accurate outcomes. Evaluation of the system was conducted using the KDD dataset on a physical cloud testbed, yielding satisfactory results in attacks detection.

Mohamed I. et al.[17] introduced a supervised DoS detection method based on a feed-forward neural network. This method involves three main steps: (1) Collection of incoming network traffic, (2) selection of relevant features for DoS detection using an unsupervised Correlation-based Feature Selection (CFS) method, (3) classification of incoming network traffic into DoS or normal traffic. The approach demonstrates strong performance on the UNSW-NB15 and NSL-KDD datasets.

Mustapha B. et al.[17] introduced a two-phase classifier utilizing the RepTree algorithm and a subset of protocols for network intrusion detection systems. Initially, incoming network traffic is categorized into TCP, UDP, or Other protocols, followed by classification into normal or anomalous traffic. Subsequently, a multi-class algorithm is employed to categorize anomalies detected in the first phase, identifying attack classes for selecting suitable interventions. Experimental evaluations were conducted using the UNSW-NB15 and NSL-KDD datasets.

Gul et. al.[18] proposed a Cloud NIDS designed to detect network intrusions within Cloud environments. This system employs rule-matching techniques to identify network attacks targeting tenant VMs. Additionally, the Cloud IDS is capable of efficiently handling large volumes of data packets and analyzing them effectively.

Varadharajan & Tupakula[19] presented a Hypervisor-based IDS tailored for Cloud environments. This system integrates traditional misuse and anomaly detection techniques with VM introspection to enhance IDS performance in the cloud.

While many IDSs in Cloud environments rely on traditional techniques such as Snort IDS, they often achieve significant performance in standard information system infrastructures or private Clouds. However, they face limitations when confronted with new distributed and sophisticated attacks. Therefore, there is a need for further scalable and distributed techniques to be adapted for Cloud IDSs.

CICIDS2017 Dataset

The CICIDS2017 dataset is a comprehensive collection of network traffic data for intrusion detection research. It includes meticulously curated data that simulate various attack scenarios and user behaviors, providing a realistic representation of actual network traffic. The dataset encompasses a range of attack types such as brute force, port scanning, denial-of-service (DoS), and distributed denial-of-service (DDoS) attacks. It captures network traffic behavior through packet-level properties, flow-level statistics, and time-based features. This dataset is widely used by academics and industry experts to explore network security issues, develop innovative detection methods, and evaluate intrusion detection systems. It offers a realistic environment for testing intrusion detection systems, and researchers can access the dataset and associated tools on the University of New Brunswick's (UNB) website.

Table 3: CICIDS 2017 Intrusion Dataset

Dataset	Description
Name	CICIDS2017
Source	University of New Brunswick (UNB)

Purpose	Intrusion Detection System (IDS) research
Data Size	256 GB
Features	79
Attack Types	Multiple
Year	2017

III. Methodology

Proposed IDS Systems

A) CNN-based Intrusion Detection System (IDS)

The functions and processes executed by the CNN architecture are represented mathematically in a model for a CNN-based Intrusion Detection System (IDS). The mathematical model for a CNN-based IDS includes the following components:

Convolution Operation: In a CNN, the convolution operation involves applying learnable filters to an input tensor. Each filter performs a convolution operation over a local receptive field of the input tensor. The mathematical expression for the convolution operation is represented as follows:

Eq. 5.1 $$O(i,j) = \sum_{m,n} I(i+m, j+n) \cdot F(m,n) + b$$

where O(i, j) represents the output value at position (i, j), I(i + m, j + n) represents the input value at position (i + m, j + n), F(m, n) represents the filter coefficient at position (m, n), and b represents the bias term.

Activation Function: Following the convolution operation, an activation function is applied element-wise to introduce non-linearity into the network. A commonly used activation function in CNNs is the Rectified Linear Unit (ReLU), mathematically expressed as:

$$ReLU(x) = \max(0, x)$$ Eq. 5.2

where x represents the input value.

Pooling Operation: The pooling operation reduces the spatial dimensions of the feature maps while maintaining essential features. Max pooling is a frequently utilized pooling technique in CNNs. It can be mathematically represented as:

$$O(i,j) = \max\{I(m,n) | m,n \in [i, i+k] \times [j, j+k]\}$$ Eq. 5.3

where O(i, j) represents the output value at position (i, j), I(m, n) represents the input value at position (m, n), and k is the size of the pooling window.

Fully Connected Layers: In a CNN, the fully connected layers link every neuron from the previous layer to every neuron in the next layer. The mathematical operations in these layers include matrix multiplications and the application of activation functions.

Consider a fully connected layer with an input vector x, weight matrix W, and bias vector b. The mathematical model for this fully connected layer can be expressed as:

$$y = f(W \cdot x + b)$$ Eq. 5.4

where y represents the output vector and f() is the activation function applied element-wise.

The fundamental operations executed by a CNN-based IDS are encapsulated in these equations. It's essential to note that the mathematical model can vary based on the specific architecture and modifications made to the CNN-based IDS. The model might include additional layers, skip connections, regularization techniques, and other components depending on the requirements and design choices.

Moreover, the mathematical model for a CNN-based IDS encompasses the optimization process during training, alongside the fundamental operations previously discussed. The complete mathematical model for a CNN-based IDS consists of concatenation and composition of these fundamental operations, coupled with suitable activation functions, regularization methods, and optimization algorithms.

The specific equations and mathematical formulations can be further tailored based on the architecture, hyperparameters, and particular IDS objectives. It's important to recognize that while the provided mathematical model offers a general overview of the necessary calculations for a CNN-based IDS, the actual implementation and optimization may require additional factors and methods to improve the IDS's performance and accuracy.

B) LSTM-based Intrusion Detection System (IDS):

An LSTM-based Intrusion Detection System (IDS) leverages a sequence of mathematical processes within the LSTM cell to detect long-term dependencies and manage data storage and retrieval over time. The operations involved are:

The input gate, which regulates the amount of new information integrated into the cell state. The computation involves the following equations using a sigmoid activation function:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{Eq. 5.5}$$

The forget gate controls the amount of data that is discarded from the cell state. Its computation involves the following equations, which utilize a sigmoid activation function:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{Eq. 5.6}$$

The output gate determines how much data from the cell state is passed to the next layer. Its computation involves the following equations, which employ a sigmoid activation function:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{Eq. 5.7}$$

Cell State: The cell state, serving as the LSTM's memory, is updated using the input gate, forget gate, and previous cell state. The following equations are involved:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{Eq. 5.8}$$

Hidden State: The hidden state, representing the output of the LSTM cell, is updated using the output gate and the cell state. The equations involved are:

$$h_t = o_t \cdot \tanh(c_t) \quad \text{Eq. 5.9}$$

LSTM Layer: In an LSTM-based IDS, several LSTM cells are typically stacked to form an LSTM layer. The output from each LSTM cell becomes the input for the next cell in the sequence. The mathematical operations previously described are sequentially applied to each LSTM cell within the layer.

Fully Connected Layers: After the LSTM layer, fully connected layers can be added to further process the LSTM layer's output for classification or detection tasks. The computations in these fully connected layers are similar to those in the CNN-based IDS as discussed earlier.

Output Layer: The output layer in an LSTM-based IDS is used for classification or detection. The number of neurons in the output layer depends on the task and the IDS classification requirements. For binary classification, a single sigmoid neuron is used. For multi-class classification, a softmax activation function is employed, with the number of neurons equal to the number of classes.

Loss Function and Optimization: During training, loss functions measure the difference between the predicted output and the ground truth labels. Binary cross-entropy is used for binary classification, while categorical cross-entropy is used for multi-class classification. Optimization of weights and biases is achieved through stochastic gradient descent (SGD) or its variants. Backpropagation through time (BPTT) updates the parameters by computing gradients of the loss function with respect to the parameters, thereby minimizing the loss.

It is important to note that the mathematical model provided gives a general overview of the computations involved in an LSTM-based IDS. The actual implementation might include additional architectural variations, regularization techniques, and hyperparameter tuning to enhance the IDS performance.

The specific equations and mathematical formulations can be tailored based on the IDS requirements, dataset characteristics, and objectives. Experimentation and fine-tuning are often necessary to optimize the model's performance and achieve accurate intrusion detection.

C) GAN-based Intrusion Detection System (IDS):

Generator Network: In a GAN-based IDS, the generator network's purpose is to produce synthetic network traffic data that mimics real traffic. It starts with random noise as input and generates synthetic data samples. The mathematical model for the generator network consists of several fully connected or convolutional layers, each followed by activation functions (e.g., ReLU) and possibly normalization layers (e.g., batch normalization).

For a simple mathematical example of a fully connected generator network, consider an input noise vector z. The generator network can be mathematically expressed as:

$$G(z) = f(W_g \cdot z + b_g) \quad \text{Eq. 5.10}$$

where G(z) represents the generated synthetic sample, f() is the activation function, W_g represents the weight matrix, and b_g represents the bias vector.

Discriminator Network: In a GAN-based IDS, the discriminator network is responsible for distinguishing between real and fake network data samples. It receives input from genuine dataset samples or synthetic samples generated by the generator network. The mathematical model of the discriminator network typically includes activation functions (such as ReLU), followed by normalization layers, and potentially a series of fully connected or convolutional layers.

Consider a basic mathematical formulation for a fully connected discriminator network, akin to the generator network. The discriminator network can be expressed as follows given an input sample x (whether genuine or synthetic):

$$D(x) = f(W_d \cdot x + b_d) \quad \text{Eq. 5.11}$$

where D(x) is the discriminator's output, f() is its activation function, W_d is its weight matrix, and b_d is its bias vector.

IV. Results

Evidently, the GAN model surpasses the LSTM and CNN models in terms of precision, recall, accuracy, and F1-score based on the assessment findings using the CICIDS 2017 dataset. The GAN model obtains a precision of 0.976 and a recall of 0.978, both of which show low false positive and false negative rates, respectively. This suggests that network traffic data invasions are efficiently detected and classified by the GAN model.

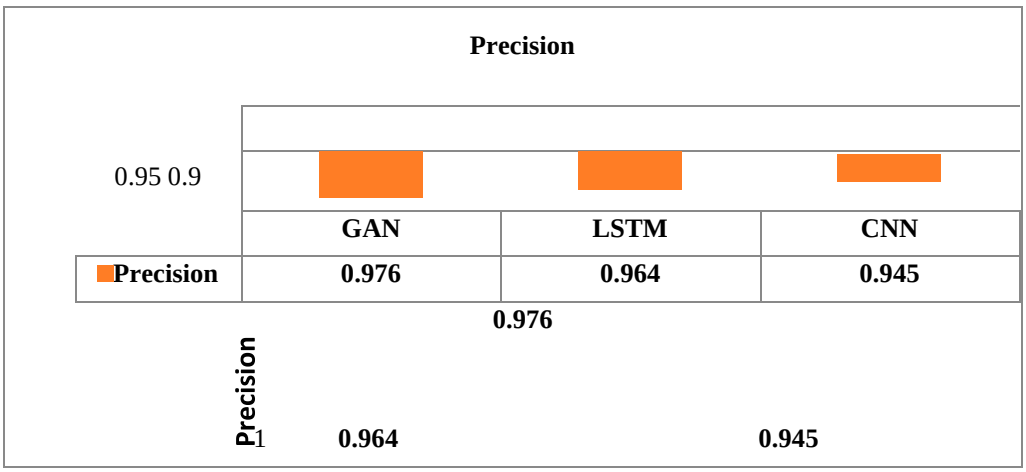

Figure 2: Precision score of DL approaches.

The GAN model also obtains an accuracy of 0.985, demonstrating a high degree of overall accuracy in its predictions. A performance that strikes a balance between recall and precision is indicated by an F1 score of 0.965. These findings show how the GAN model performs well at properly identifying intrusions and reducing misclassifications.

Figure 3: Recall score of DL approaches.

With precision of 0.964, recall of 0.972, accuracy of 0.978, and an F1-score of 0.962, the LSTM model compares favorably. These results show a great performance in intrusion detection, although being marginally lower than the GAN model.

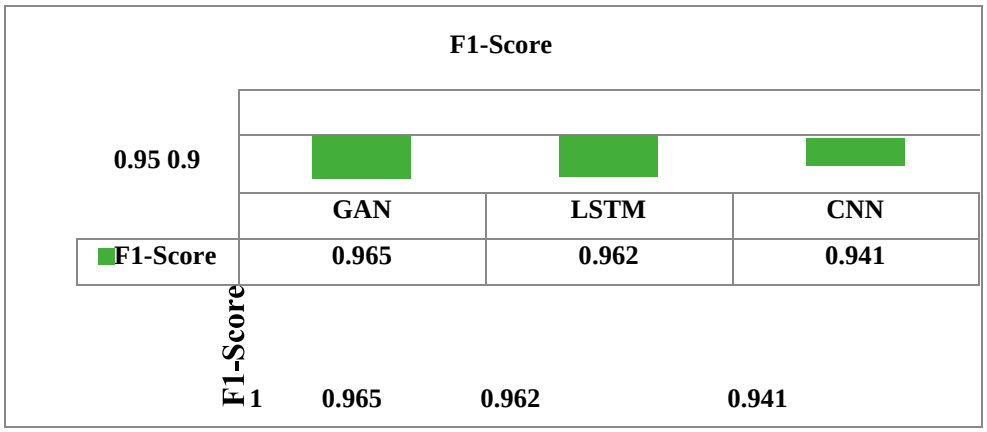

Figure 4: F1-Score score of DL approaches.

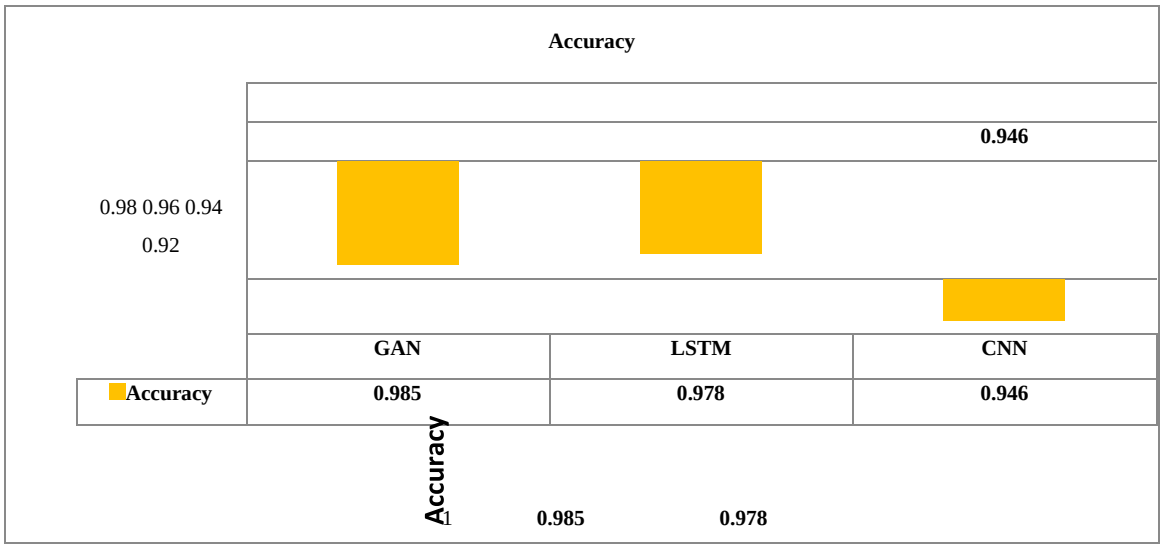

Figure 5: Accuracy score of DL approaches.

While the CNN model achieves precision, recall, accuracy, and an F1-score of 0.945, 0.946, and 0.941, it performs marginally worse than the GAN and LSTM models. It's crucial to remember that these outcomes are still respectable and show how well the CNN model works at spotting intrusions.

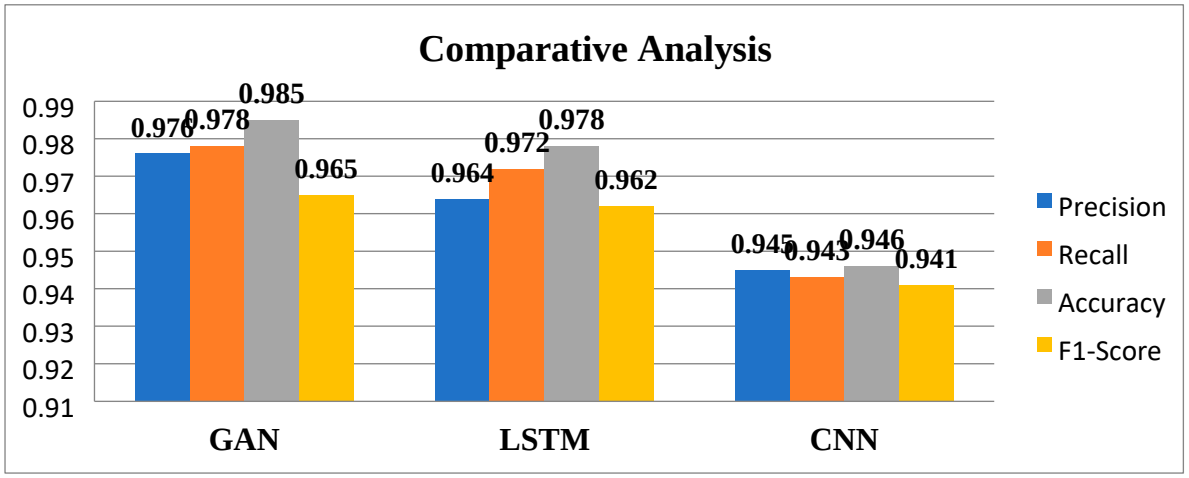

Figure 6: Evaluation score of DL approaches.

The GAN model performs bet overall in terms of many assessment measures, demonstrating its supremacy in precisely detecting intrusions in network traffic data. While the CNN model performs somewhat worse but still has reliable intrusion detection skills, the LSTM model also displays great performance. The GAN model in particular shows promise for obtaining high accuracy and precision in identifying network intrusions, as evidenced by these findings, which highlight the potential of deep learning approaches in the field of intrusion detection systems.

V. Conclusion

This study primarily focused on developing and evaluating deep learning models for intrusion detection systems (IDS) within big data environments. Utilizing the CICIDS2017 dataset, the research investigated the efficacy of three prominent deep learning architectures: CNN, LSTM, and GAN. Evaluation results revealed that the GAN model outperformed the LSTM and CNN models across various metrics such as precision, recall, accuracy, and F1-score. The GAN model exhibited high performance in accurately identifying and classifying intrusions within network traffic data, showcasing its ability to capture complex patterns and generate synthetic traffic data resembling real-world scenarios. While slightly trailing the GAN model, the LSTM model also demonstrated robust performance, leveraging its capability to detect long-term dependencies in sequential data. Despite slightly lower performance compared to the GAN and LSTM models, the CNN model exhibited reliable intrusion detection capabilities. Overall, this research underscores the importance of leveraging deep learning techniques for IDS development in big data environments. The GAN model, in particular, shows promise in generating synthetic traffic data and effectively detecting intrusions. These findings contribute valuable insights into the potential of deep learning models to address the challenges posed by evolving cyber threats in complex environments. Further research aimed at refining and optimizing GAN-based IDS could enhance its performance and robustness. Additionally, exploring hybrid CNN-LSTM architectures and integrating various deep learning methodologies may yield further improvements in intrusion detection outcomes. By harnessing the power of deep learning and leveraging the vast amount of data available in big data environments, IDS can become more precise, efficient, and adaptive in combating network intrusions, thereby enhancing overall organizational and network security.

References

[1] Wikipedia. "2016 Dyn Cyberattack." Accessed November 10, 2017.
[2] The Guardian. "Ddos Attack That Disrupted Internet Was Largest Of Its Kind In History, Experts Say." Accessed April 10, 2017.
[3] Diro, A. A., And N. Chilamkurti. "Distributed Attack Detection Scheme Using Deep Learning Approach For Internet Of Things." Future Generation Computer Systems 82 (2018): 761-768. Doi:10.1016/J.Future.2017.08.043.

[4] Kukkar, A., D. Gupta, S. M. Beram, M. Soni, N. K. Singh, A. Sharma, R. Neware, M. Shabaz, And A. Rizwan. "Optimizing Deep Learning Model Parameters Using Socially Implemented Iomt Systems For Diabetic Retinopathy Classification Problem." Ieee Transactions On Computational Social Systems 10, No. 4 (2022). Doi:10.1109/Tcss.2022.3213369.

[5] Muna, A. H., N. Moustafa, And E. Sitnikova. "Identification Of Malicious Activities In Industrial Internet Of Things Based On Deep Learning Models." Journal Of Information Security And Applications 41 (2018): 1-11. Doi:10.1016/J.Jisa.2018.05.002.

[6] Mehbodniya, A., I. Alam, S. Pande, R. Neware, K. P. Rane, M. Shabaz, And M. V. Madhavan. "Financial Fraud Detection In Healthcare Using Machine Learning And Deep Learning Techniques." Security And Communication Networks (2021): 1-8. Doi:10.1155/2021/9293877.

[7] Vinayakumar, R., M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, And K. Simran. "A Visualized Botnet Detection System Based Deep Learning For The Internet Of Things Networks Of Smart Cities." Ieee Transactions On Industry Applications 56, No. 4 (2020): 4436-4456. Doi:10.1109/Tia.2020.2971952.

[8] Parra, G. D. L. T., P. Rad, K. K. R. Choo, And N. Beebe. "Detecting Internet Of Things Attacks Using Distributed Deep Learning." Journal Of Network And Computer Applications 163 (2020): 102662. Doi:10.1016/J.Jnca.2020.102662.

[9] Ajani, S., And M. Wanjari. "An Efficient Approach For Clustering Uncertain Data Mining Based On Hash Indexing And Voronoi Clustering." In 2013 5th International Conference And Computational Intelligence And Communication Networks, 486-490. Ieee, September 2013. Doi: 10.1109/Cicn.2013.106.

[10] Fernandes, D. A., L. F. Soares, J. V. Gomes, M. M. Freire, And P. R. Inacio. "Security Issues In Cloud Environments: A Survey." International Journal Of Information Security 13, No. 2 (2014): 113-170.

[11] Iqbal, S., M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, And K.-K. R. Choo. "On Cloud Security Attacks: A Taxonomy And Intrusion Detection And Prevention As A Service." Journal Of Network And Computer Applications 74 (2016): 98-120.

[12] Roschke, S., F. Cheng, And C. Meinel. "Intrusion Detection In The Cloud." In Dependable, Autonomic And Secure Computing, 2009. Dasc'09. Eighth Ieee International Conference On, 729-734. Ieee, 2009.

[13] Modi, C., D. Patel, B. Borisanya, A. Patel, And M. Rajarajan. "A Novel Framework For Intrusion Detection In Cloud." In Proceedings Of The Fifth International Conference On Security Of Information And Networks, 67-74. Acm, 2012.

[14] Gupta, S., And P. Kumar. "An Immediate System Call Sequence Based Approach For Detecting Malicious Program Executions In Cloud Environment." Wireless Personal Communications 81, No. 1 (2015): 405-425.

[15] Li, Z., W. Sun, And L. Wang. "A Neural Network Based Distributed Intrusion Detection System On Cloud Platform." In Cloud Computing And Intelligent Systems (Ccis), 2012 Ieee 2nd International Conference On, Vol. 1, 75-79. Ieee, 2012

[16] Idhammad, M., K. Afdel, And M. Belouch. "Dos Detection Method Based On Artificial Neural Networks." International Journal Of Advanced Computer Science And Applications (Ijacsa) 8, No. 4. Doi:Http://Dx.Doi.Org/10.14569/Ijacsa.2017.080461.

[17] Mustapha, B., E. H. Salah, And I. Mohamed. "A Two-Stage Classifier Approach Using Reptree Algorithm For Network Intrusion Detection." International Journal Of Advanced Computer Science And Applications (Ijacsa) 8, No. 6. Doi:Http://Dx.Doi.Org/10.14569/Ijacsa.2017.080651.

[18] Gul, I., And M. Hussain. "Distributed Cloud Intrusion Detection Model." International Journal Of Advanced Science And Technology 34, No. 38 (2011): 135.

[19] Varadharajan, V., And U. Tupakula. "Security As A Service Model For Cloud Environment." Ieee Transactions On Network And Service Management 11, No. 1 (2014): 60-75.

[20] Ring, M., S. Wunderlich, D. Grdl, D. Landes, And A. Hotho. "Flow-Based Benchmark Data Sets For Intrusion Detection." In Proceedings Of The 16th European Conference On Cyber Warfare And Security (Eccws), 361-369. Acpi, 2017.

[21] Haddadpajouh, H., A. Dehghantanha, R. Khayami, And K. K. R. Choo. "A Deep Recurrent Neural Network-Based Approach For Internet Of Things Malware Threat Hunting." Future Generation Computer Systems 85 (2018): 88-96. Doi:10.1016/J.Future.2018.03.007.

[22] Popoola, S. I., B. Adebisi, M. Hammoudeh, G. Gui, And H. Gacanin. "Hybrid Deep Learning For Botnet Attack Detection In The Internet-Of-Things Networks." Ieee Internet Of Things Journal 8, No. 6 (2020): 4944-4956. Doi:10.1109/Jiot.2020.3034156.

[23] Manimurugan, S., S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, And R. Patan. "Effective Attack Detection In Internet Of Medical Things Smart Environment Using A Deep Belief Neural Network." Ieee Access 8 (2020): 77396-77404. Doi:10.1109/Access.2020.2986013.

[24] Ng, B. A., And S. Selvakumar. "Anomaly Detection Framework For Internet Of Things Traffic Using Vector Convolutional Deep Learning Approach In Fog Environment." Future Generation Computer Systems 113 (2020): 255-265. Doi:10.1016/J.Future.2020.07.020.

[25] Sharafaldin, I., A. H. Lashkari, And A. A. Ghorbani. "Toward Generating A New Intrusion Detection Dataset And Intrusion Traffic Characterization." Icissp 1 (2018): 108-116. Doi:10.5220/0006639801080116.

[26] Ferrag, M. A., O. Friha, D. Hamouda, L. Maglaras, And H. Janicke. "Edge-Iiotset: A New Comprehensive Realistic Cyber Security Dataset Of Iot And Iiot Applications For Centralized And Federated Learning." Ieee Access 10 (2022): 40281-40306. Doi:10.1109/Access.2022.3165809.

[27] Chawla, N. V., K. W. Bowyer, L. O. Hall, And W. P. Kegelmeyer. "Smote: Synthetic Minority Oversampling Technique." Journal Of Artificial Intelligence Research 16 (2002): 321-357. Doi:10.1613/Jair.953.

[28] Samek, W., A. Binder, G. Montavon, S. Lapuschkin, And K. R. Müller. "Evaluating The Visualization Of What A Deep Neural Network Has Learned." Ieee Transactions On Neural Networks And Learning Systems 28, No. 11 (2016): 2660-2673. Doi:10.1109/Tnnls.2016.2599820.

[29] Liu, W., Z. Wang, X. Liu, N. Zeng, Y. Liu, And F. E. Alsaadi. "A Survey Of Deep Neural Network Architectures And Their Applications." Neurocomputing 234 (2017): 11-26. Doi:10.1016/J.Neucom.2016.12.038.

[30] Deng, L., G. Hinton, And B. Kingsbury. "New Types Of Deep Neural Network Learning For Speech Recognition And Related Applications: An Overview." In 2013 Ieee International Conference On Acoustics, Speech And Signal Processing, 8599-8603. Ieee, May 2013. Doi:10.1109/Icassp.2013.6639344.

[31] Albawi, S., T. A. Mohammed, And S. Al-Zawi. "Understanding Of A Convolutional Neural Network." In 2017 International Conference On Engineering And Technology (Icet), 1-6. Ieee, August 2017. Doi:10.1109/Icengtechnol.2017.8308186.

[32] Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, And T. Chen. "Recent Advances In Convolutional Neural Networks." Pattern Recognition 77 (2018): 354-377. Doi:10.1007/978-1-4842-2845-6_6.

[33] Yu, Y., X. Si, C. Hu, And J. Zhang. "A Review Of Recurrent Neural Networks: Lstm Cells And Network Architectures." Neural Computation 31, No. 7 (2019): 1235-1270. Doi:10.1162/Neco_A_01199.

[34] Sherstinsky, A. "Fundamentals Of Recurrent Neural Network (Rnn) And Long Short-Term Memory (Lstm) Network." Physica D: Nonlinear Phenomena 404 (2020): 132306. Doi:10.1016/J.Physd.2019.132306.

[35] Tschannen, M., O. Bachem, And M. Lucic. "Recent Advances In Autoencoder-Based Representation Learning." Third Workshop Bayesian Deep Learning. Arxiv Preprint Arxiv:1812.05069, 2018.
[36] Meng, Q., D. Catchpoole, D. Skillicom, And P. J. Kennedy. "Relational Autoencoder For Feature Extraction." In 2017 International Joint Conference On Neural Networks (Ijcnn), 364-371. Ieee, May 2017. Doi:10.1109/Ijcnn.2017.7965877.
[37] Chen, Z., C. K. Yeo, B. S. Lee, And C. T. Lau. "Autoencoder-Based Network Anomaly Detection." In Wireless Telecommunications Symposium, 1-5. Ieee, April 2018. Doi:10.1109/Wts.2018.8363930.
[38] Ajani, S. N., And S. Y. Amdani. "Probabilistic Path Planning Using Current Obstacle Position In Static Environment." In 2nd International Conference On Data, Engineering And Applications (Idea), 1-6. Ieee, February 2020. Doi:10.1109/Idea49133.2020.9170727.