RESEARCH
www.commprac.com
ISSN 1462 2815

NAVIGATING THE CLOUD AND FOG: UNVEILING SECURITY CHALLENGES IN AI APPLICATIONS

Jaishree Jain [1], Shashank Sahu [2], Santosh Kumar Upadhyay [3], Yogendra Narayan Prajapati [4], Ashish Dixit [5] and Samender Singh [6]

[1,2,3,4,5,6] Department of Computer Science & Engineering,
Ajay Kumar Garg Engineering College Ghaziabad.
Email: [1]jaishree3112@gmail.com, [2]sahushank@akgec.ac.in, [3]upadhyaysantosh@akgec.ac.in, [4]ynp1581@gmail.com, [5]dixitashish@akgec.ac.in, [6]singhsamendar@akgec.ac.in

DOI: 10.5281/zenodo.11096274

Abstract

There are various security risks with cloud- or fog-based machine learning services. Because machine learning applications rely on these services, it is imperative to secure the underlying cloud or fog services to prevent serious disruptions to the applications. We distinguish based on whether Artificial Intelligence applications are employed in a fog computing network or the cloud because the needs for these applications can also vary. This consequently gives rise to various threats or avenues for attack. Security responsibilities for cloud platforms can be split up amongst several parties. Even though fog computing networks have fewer responsibilities, we still need to safeguard services from physical access to the devices at the network's edge because they have been moved there. Lastly, we go into specific information security requirements for AI-related applications.

Keywords: AI Applications, Cyber Security, Cloud Network, Fog Computing.

I. INTRODUCTION

Artificial Intelligence (AI) has been a popular topic, especially for non-specialist audiences, at least since OpenAI's introduction of ChatGPT in November 2022. It's a little misconception that machine learning (ML) has been used to an increasing number of services in the business, industrial, and private sectors in recent years. Cloud services play a major role in many machine learning applications because they offer a quick, scalable, adaptable, and affordable infrastructure for executing sophisticated machine learning models and algorithms. Businesses may successfully and effectively deploy their ML projects with them. Key advantages of cloud services for machine learning include:

1) Scalability: Cloud services enable scaling up the necessary processing power to satisfy the demands of the specific machine learning application. Not only is the ML application's execution important, but more processing power or memory that can accommodate bigger data sets can help accelerate the models' training process. Scaling down computer resources when they are no longer required is made possible via elasticity.

2) Flexibility: There are many different types of machine learning (ML) services out there, such as simple cloud-based ML development platforms and specialized ML services for speech-to-text, text-to-speech, translation, chats, automated image and video analysis, and a host of other applications. Cloud solutions comprise platforms, software, and infrastructure, all of which can be customized for the various customers and applications. Machine learning applications can be implemented in a variety of ways, such as container orchestration, virtual machines, and serverless computing.

3) Economic efficiency: Organizations utilizing machine learning applications can rent processing capacity or storage together with subscription-based licensing, eliminating the need for them to purchase the necessary gear, perpetual licenses, or pay for its operation. Compared to traditional data centers, Cloud Service Providers (CSP) offer more precise cost models. The global availability of ML services by CSPs enables enterprise ML products to be distributed internationally at a reduced cost.

4) Data management: Large data sets that typically go along with contemporary machine learning applications can be processed and stored via cloud services.

5) Integration: As a now-established technology, cloud services provide a range of other well-established tools and services, such as workflow engines, database connectivity, and visualization tools.

It is not acceptable to generalize, though, and claim that cloud services are the sole way to implement AI and ML applications. Moreover, there are other fields in which machine learning might be applied, such as autonomous driving, where it is either impractical or not always feasible to connect to cloud services. Real-time integration of visual and radar data regarding a vehicle's traffic status is essential for autonomous driving. Especially when someone's life or limb is at danger. Therefore, it is essential that the car has enough memory and processing capability to handle all processing, computation, and decision-making at the moment. As the aforementioned example should demonstrate, standards like low latency or real-time capability are frequently essential for these kinds of uses. Nevertheless, cloud services are a component of the ecosystem for autonomous driving.

An argument in Favor of on-premise hardware and against cloud services typically centres on the need for better control over information security and data protection. However, for some applications, cloud services have been used by sectors like banking and healthcare that have strict security regulations [1]. Building a cloud architecture that is private to an enterprise means keeping it that way in case regulations or trusted hardware requirements call for tight control. The latency and bandwidth need of cloud computing can be decreased by utilizing the fog layer [2].

Machine Learning has been proved to be very important technique to learn the insight in the vast dataset. Reliable and correctly operating cloud services or fog computing networks are a necessary precondition for the proper operation of machine learning systems. On the other hand, it is obvious that ML applications will face severe issues in the event that cloud services or a fog computing network are compromised. As a result, we would want to discuss the security issues that cloud or fog computing-based machine learning systems face in this article, along with best practices and recommendations for mitigating those risks.

There are many applications domain where machine learning is giving excellent results. Few important application domains are Sentiment Analysis [45], Plant Disease detection [46,47,48,49,50], Disease detection [51,52], Cloud Monitoring [53,54,55], Fog security monitoring [32,56]. One instance of a machine learning application implemented in the fog computing environment is the smart grid system's processing of sensor data. An electrical network that monitors and optimizes energy use through the use of sensors and smart devices is called a smart grid. The edge devices in the smart grid can process and analyze the sensor data in a standard fog computing architecture. Machine learning models that have been developed using sensor data are able to estimate energy demand and carry out the necessary

optimizations. Real-time power grid optimization can be accomplished without sending all (potentially privacy-critical) data to a distant cloud by analyzing the data at the edge. This can boost system performance and lower latency. Furthermore, the fog computing network contributes to improved security by doing away with the requirement for data transmission across open networks, of the smart grid. Because it is located on the edge device, sensitive data is more shielded from potential threats.

The following is how the paper is organized: Security issues are covered in Section II. Security issues with fog computing are discussed in Section III. In conclusion, Section IV discusses unique security issues with machine learning applications in cloud or fog environments. A conclusion and a vision for future work round out the report.

II. DIFFICULTIES IN CLOUD COMPUTING SECURITY

Machine learning services that are offered via the Internet depend on the security of the underlying cloud services. Successful attacks on the cloud services that machine learning systems rely on have the potential to cause significant harm to these systems. Cloud services also need to be carefully protected against attacks for other reasons. The traditional Cloud Security Responsibility Model (CSRM) primarily assigns roles based on the user and cloud provider, with particular emphasis on the Infrastructure as a Service (IaaS), PaaS, and SaaS service categories [3]. No matter how many people are involved in sharing accountability for It is especially crucial to guarantee accountability at the interfaces between various parties when it comes to the cyber security of a cloud service. This is due to the possibility that a security issue affecting one party's domain could jeopardize the domains of other parties.

The system architecture's layer model makes it abundantly evident that security flaws at one level can instantly affect a higher level, like the level that houses user data. Using the Common Vulnerability Scoring System (CVSS), Süß et al. compiled and ranked the information security problems that cloud services were experiencing at the time at the Cloud Computing 2019 conference [4]. Everyone's life has shifted to the cloud, and many businesses and organizations have become more. As a result, scammers are currently trying to take advantage of weaknesses in cloud computing systems. Frequently, these were traditional assaults that may have been against the web services.

A. Breach of Data

We begin by looking at threats that are explicitly aimed at cloud data. This could contain private client data, private company documents, or private health information. A data breach gives unauthorized access to sensitive data, which can clearly have negative impacts on individuals as well as businesses [5]. Data breaches can be caused by targeted assaults or unintentional disclosure of personal information due to configuration mistakes or insufficient security measures [6]. Data breaches usually cause customers and business partners to lose faith in the affected organization, and this is often followed by a substantial loss of public opinion. Businesses that are impacted by data breaches usually fear regulatory fallout because these incidents frequently lead to infractions of laws and rules, such the EU General Data Protection Regulation (GDPR). Additional impairment to company activity is a common consequence of data breaches. For example, it would be just as terrible if an inattentive employee accidentally erased firm data, or if an attacker removed it on

purpose. Many of the cloud services that were accessible during the pandemic make sense to continue using. Although it is evident that implementing stringent access controls and encrypting data stored in the cloud can avoid data breaches, the frequency and severity of data breaches over the previous two years (see, for example, [7] or [8]) are alarming, even though the first year of the Covid-19 2017 has been called the "worst year on record" in terms of data leaks [9].

B. Malware Incidents

In an attack known as a "ransomware attack," the attacker encrypts the victim's data using specialized software and demands payment in exchange for the key's surrender. The WannaCry ransomware assault of 2017 is likely the most well-known example of a ransomware attack. It locked user data on Microsoft Windows machines and demanded Bitcoin ransom payments [10]. Early ransomware versions frequently encrypted only the user's data on the local hard drive, but later versions also started to encrypt data on linked disks and cloud-based storage [11]. Here, those that create ransomware offer criminals their malware as a service. RaaS platforms frequently include a range of choices that enable hackers to design and carry out their own ransomware attacks [12]. These sites frequently let the ransomware be customized, including the ability to choose targets and set the ransom amount. These platforms make it easier for a larger spectrum of people to launch ransomware campaigns by targeting less tech-savvy offenders. This raises the possibility of ransomware attacks on cloud services as well as other kinds of IT systems. As previously said, multiple parties may bear some of the responsibility for cyberattacks targeting cloud services. In this sense, ransomware assaults are not unique. A malicious ransomware infection on a user's computer or mobile device may encrypt user data stored in the cloud. Should there be a cyberattack on the CSP that encrypts the data of multiple (perhaps numerous) customers, the CSP bears accountability.

Updating all software and making frequent backups are the strongest defenses against ransomware attacks. Installing security updates as soon as possible is required. Things become accessible since malicious actors frequently take advantage of flaws in out-of-date software. This also includes updating antivirus software on a regular basis. Backups are essential as, even in the unlikely event that a user finds themselves with no choice but to pay the requested ransom, there is no assurance that their personal data would be restored undamaged. How much do you believe a criminal who has blackmailed you can deliver on his promise?

C. Denial of Service via Distribution

Authorized customers can no longer access the service in question because the bandwidth of the service's Internet connection is insufficient. Thus, DDoS assaults typically target the accessibility of services. There was a noticeable spike in DDoS attacks during the Covid-19 outbreak. This claim is demonstrated in Figure 1 by comparing the data for Germany before or during the pandemic's start (2020) and during its peak (2021).

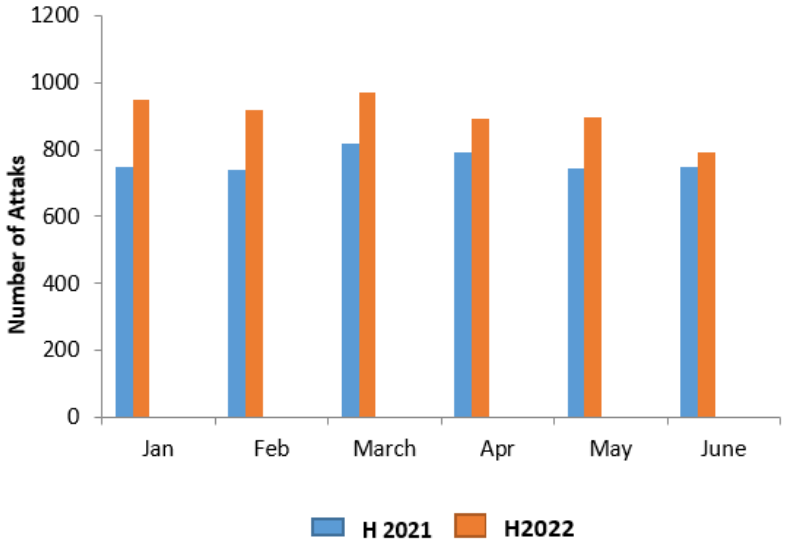

Figure 1: According to [13], the frequency of monthly DDoS attacks during the Covid-19 epidemic in 2021 and 2022 (Germany) was reported

As per [14], adversaries are increasingly utilizing cloud-based Virtual Private Servers (VPS) to create botnets that are employed for DDoS attacks. As the past few years have demonstrated, botnets constructed on unreliable Internet of Things devices are significantly weaker than this setup. Attack durations are decreasing, while ransom DDoS attacks are increasing in frequency. The goal of this is to threaten the victim with extortion for the ransom. DDoS assaults can be prevented in part by using firewalls in conjunction with intrusion detection systems (IDS), traffic filtering, and load balancing.

D. Reliance on Outside Software

Instead, it is common practice to rely on well-established libraries, etc., which are produced and provided by external sources. It is possible to perform a critical study of the integrated program and look for vulnerabilities in the source code of software that is available as open source. Unfortunately, users often think that this third-party software has been thoroughly tested and is safe without doing any additional research. It is impossible to totally rule out vulnerabilities brought on by bad software design or programming mistakes, and these problems are sometimes only discovered after the application has been in use for a long time.

One well-known instance of vulnerability in third-party software is Log4Shell, which received the maximum CVSS severity rating of 10.0 [15]. A Java-based logging tool called Log4j is utilized in both private and open source software, and it has become the industry standard for this use case. Due to the Log4Shell vulnerability, adversaries were able to remotely run any code on the host system, which included mining cryptocurrency. Numerous systems were impacted, including Apple's iCloud [17] and Amazon Web systems (AWS) [16].

Which functionality is realized by third-party software is, in theory, irrelevant. However, in actual use, these are typically security protections with technically particular cloud aspects rather than ones that are really novel. To clarify, the application of security roles, authorization guidelines, key or certificate management, and procedures related to public key Infrastructures (PKI) might come up. Naturally,

the caliber of these libraries affects the security of cloud services that make use of third-party software. For instance, a breach in an integrated authentication service may expose the personal information of CSP clients, in violation of the EU GDPR. In this case, it doesn't initially matter if the information is widely available online to everybody (data breach) or if it is considerably more difficult to obtain; what matters is that the information is taken and made public by an attacker. It is the CSP, not the programmers of the (open source) library, who have responsibility and liability in this instance.

Utilizing third-party software invariably carries some risk. It is therefore advised to check any third party to reduce this third-party software thoroughly for flaws and to thoroughly test it in conjunction with one's own software components.

E. Insecure Application Programming Interfaces

Applications and cloud services communicate with one another through APIs, or application programming interfaces. APIs could constitute a significant security risk if they are not adequately guarded. For instance, a cloud API with an insecure interface may allow anyone to access private information (see data breaches, Subsection II-A).

An excellent summary of API security concerns is given through the API Security Project OWASP [18]. The top 10 security issues they list are as follows:

1) Object Level Authorization is broken
2) Violated Authentication of Users
3) Abnormally High Data Exposure
4) Insufficient Resources & Rate Restriction
5) Split Function Level Permission
6) Large-Scale Task
7) Misconfigured Security
8) Injection
9) Ineffective Asset Administration
10) Inadequate Recording & Observation

Many of these topics have previously been covered in this paper, and for readers who are knowledgeable about information security, all of the items speak for themselves. It should be noted at this point, though, that the quoted paper goes into great length to describe each of the aforementioned security issues and also suggests suitable solutions. This was written as of the OWASP API Security Project is putting the finishing touches on their top 10 list for 2023.

F. Security via Cloud-Native

"Cloud, Clusters, Containers, and Code are the 4C's of cloud Native security," according to the Kubernetes documentation [19]. Docker, the original and most well-known container engine, specifically prioritizes developer experience and usability over security. This results in exploits such as DirtyCOW; nevertheless, even for seasoned kernel developers, it is challenging to comprehend the fundamental vulnerability of the Linux operating system that underlies this [20].

There are currently other container engines available, such as Google gVisor, AWS Firecracker, OpenStack KataContainers, and Pod-man. Many of them concentrate on safety and offer a rootless mode, for example. An overview of applied container security is given by Lize Rice [21]. The Shared Responsibility Model, which was first presented by Amazon [22], stipulates that users are in charge of security "in" the cloud. GitGuardian's 10 Rules for Better Cloud Security [23] offer a starting point for actions that can be implemented in accordance with the Shared Responsibility Model:

1) Recall that developer credentials are important (in both public and private code repositories).
2) Constantly check the default settings.
3) List storage that is open to the public.
4) Conduct frequent access control audits.
5) Make use of network structures.
6) Establish proactive logging and monitoring.
7) Expand the list of assets you have.
8) Avoid having your domain stolen.
9) It is mandatory to have a catastrophe recovery plan.
10) Boundary by hand settings.

The development of Trusted Execution Environments (TEE) for containers is one area of research concerning container security. Intel SGX is one platform on which secure Linux containers can be built, as Arnautov et al. [24] have shown. Lastly, the 4Cs mentioned above form the foundation of Kubernetes security; all major CSPs offer security and hardening guidelines for their Kubernetes deployments [19]. Workload security in Kubernetes is concerned with things like network policies, managing application secrets and encrypting them while they're at rest, and making sure that pods adhere to established pod security standards [25].

III. DIFFICULTIES WITH FOG COMPUTING SECURITY

The phrase "fog computing," which was first used by Cisco [26], refers to a distributed computing architecture that connects cloud computing and Internet of Things devices. In fog computing, calculations are done closer to the data source, either on the IoT devices themselves or on local edge servers, as opposed to sending all data to a distant cloud for processing. This maximizes system efficiency, reduces latency associated with long-distance data transit, and enhances real-time capabilities. Fog computing solves problems like latency, bandwidth limitations, and security difficulties that might arise with cloud computing by moving computation and storage closer to the data sources [27]. Higher mobility technologies such as the Internet of Things (IoT) and Vehicular Ad-hoc Networks (VANETs) benefit greatly from this strategy since it gives users speedier software services and communication. Fog computing provides better quality and reduced latency by shortening the distance between devices and processing resources with relation to conventional cloud computing [2][28].

While fog computing and cloud computing have many similarities, they also differ in a few key areas, including how they handle storage, network management, and the balance between central and local computation. Fog computing offers more effective, real-time control and enhancements for a number of systems, including parking systems, traffic patterns, healthcare, and more, thanks to this balance. Fog computing is not without its difficulties, though. Less resources than cloud computing, higher latency in some situations, energy consumption issues, load balancing, data management, and security risks are some of its drawbacks [29].

Fog computing can be viewed as an addition to regular cloud computing, given the previously described properties. Furthermore, these qualities in the context of this paper enable selecting between using the cloud or fog as the foundation for particular ML research or applications.

Fog computing faces a number of security risks that are either the same or similar to those faced by cloud computing. Fog computing often uses a dispersed network of devices, which raises the possibility of outages and network disturbances. Thus, maintaining high availability is essential to assuring continuous service delivery in addition to data confidentiality, authenticity, and integrity. Attacks have the potential to prevent fog computing systems from operating as intended and can result in data leaks, illegal access, or system failures [30]. Fog computing systems need to have strong security features like access control, intrusion detection and prevention, strong encryption, and constant monitoring in place to lessen these attacks. Furthermore, making certain that security guidelines and best practices are followed can reduce the possibility of security lapses in fog computing settings [31]. In Section II, a few security risks related to fog computing have already been discussed. To interrupt and affect services, DDoS assaults, for instance, target fog computing networks by flooding fog nodes or networks with excessive traffic [32]. Other traditional network attacks exist as well, such as replay and Man in the Middle (MITM). In fog computing, a Man-in-the-Middle (MITM) attack entails listening in on and altering messages between authorized components, jeopardizing the availability, confidentiality, and integrity of the system [33]. Replay attacks are a kind of security hazard in which a malicious party records and replays communications that have already been sent back and forth between parties in a communication session, giving the impression that they are the original sender [34]. To carry out this attack in the context of fog computing, an adversary may pose as end devices or the fog broker. Replay attacks just require the opponent to replay the messages in order to take advantage of the system; they do not require the adversary to comprehend the content of the captured messages or decrypt any encrypted data. This may result in a number of unfavorable outcomes, including data modification, illegal access, or service interruptions.

Here, we concentrate on assaults and vulnerabilities that are unique to fog computing rather than cloud computing.

A. Tactical Assaults

In fog or edge computing, a physical assault [35] entails breaching the system's actual hardware, including servers or other devices. In some systems, this can be especially difficult due to their infrastructure is dispersed throughout different geographic areas. Inadequate physical protection for these devices may make them vulnerable to harm or tampering. Physical attacks have the potential to disrupt services within the narrow geographical area that each device or server serves. For this reason, it's imperative

that fog computing adopt robust physical security measures in addition to cybersecurity safeguards.

B. User Impersonation Attack and Fog

This kind of cyberattack involves a hacker impersonating a different device or user on the network in order to target network hosts, steal information, distribute malware, or get around security measures. This kind of attack is especially sneaky since it can be hard to identify because the attacker is using what are thought to be legitimate credentials inside the framework. In the context of fog computing, impersonation attacks have the potential to impede communication between fog nodes and end devices, resulting in misunderstandings, data theft, or even a service interruption. Tu and colleagues propose a countermeasure that involves integrating physical layer security measures with a reinforcement learning algorithm. This will enhance security against impersonation attacks and optimize the process of determining which entities are legal and which are unauthorized [36].

C. Malevolent Fog Node Infiltrations

A malevolent fog node has the power to undermine network functions via many forms of assaults, hence impacting the dependability of fog-to-fog cooperation. It's also critical to recognize malevolent fog computing devices. Organizations should have a complete security strategy that includes data encryption, safe communication protocols, authentication and authorization in order to prevent malicious fog node concerns methods for detecting intrusions, maintenance of trust, routine observation, segmentation of the network, access control, and an incident response strategy.

These tactics support system resilience, improve overall security, and lower hazards. As a result, it becomes difficult to achieve comprehensive security against attacks because it requires processing data and assigning restricted capabilities. The goal of current study is to identify suitable countermeasures; as examples, we cite Ke Gu et al. [38] and Al-Khafajiy et al. [37].

The latter introduce a fog computing-based VANET that employs a method to identify rogue nodes, or automobiles or other objects that pose a threat. The fog server uses their method to calculate a reputation score for any node that might be dangerous. By analyzing the correlation between the information gathered from the node and the general configuration of the network. The fog server can more precisely identify and flag nodes that might be a danger to the security and functionality of the network by looking at these parameters.

D. Malicious Fog Nodes

When a malicious node poses as a valid fog node and joins the network to carry out attacks like data theft, denial of service, or eavesdropping, it is known as a rogue fog node attack [39]. To avert attacks by rogue fog nodes, the subsequent actions can be implemented:

- Authentication and authorization: Prior to being permitted to connect to the network, fog nodes must undergo authentication and authorization. Mutual authentication and secure boot can be used to accomplish this.
- Encryption: To avoid listening in on private conversations between fog nodes, sensitive information should be encrypted theft of data.

- Trust Management: The reliability of fog nodes can be assessed using trust management procedures. The conduct, standing, and credentials of the node may be taken into consideration.
- Network Segmentation: By dividing the network into segments, it is possible to separate the attack-prone fog nodes. This may aid in reducing the attack's scope and damage.
- Continuous Monitoring: Any illegal fog node that joins the network can be found by using the network's continuous monitoring feature. System logs, node behavior, and network traffic can all be observed to do this.

E. Attack on Temporary Secret Leakage

The Ephemeral Secret Leak- age Attack [40] poses a significant risk in the field of fog computing because of its dispersed architecture and frequent handling of sensitive data engaged. This attack is predicated on the assumption that an adversary can obtain one of the secret keys—either long-term or short-term—used for secure communication between devices. It is based on the Canetti-Krawczyk adversary model [41]. A potential security breach may occur if an opponent were to disclose a session key, which is a temporary encryption key that allows them to decrypt any data exchanged during that session. Thus, it is essential to use strong cryptographic protocols and efficient key management techniques to keep fog computing systems secure.

IV. UNIQUE SECURITY TROUBLES FACING AI APPLICATIONS

A. Information Showing ML Models

A number of the aforementioned assaults were directed at information kept on cloud or fog computing networks. Apart from the fact that this information may, for instance, be the personal information of clients, which is subsequently handled by there is another significant issue regarding there is another significant issue regarding the ML application.

A sufficiently enough quantity of training data is frequently an essential precondition for successful machine learning initiatives. As good as the training data is, so too are machine learning models. When it comes to new business models or many other areas where machine learning techniques have not yet been implemented, there are sometimes no training data available at launch.

They frequently need to be made laboriously at initially, which includes labeling the training data by hand in addition to potentially requiring a significant number of measurements to produce a large enough sample set. It is crucial to note that ML models pose a concern in light of the previously described attacks and threats give rise to alarm given the attacks and risks that have already been discussed, and their training data are extremely significant resources.

Severe business disruptions may result from models or training data being unavailable as a result of a DDoS attack. However, it would be far worse if training data or models were stolen or made public online and ended up in the hands of a rival business. For a corporation whose business model depends on these kinds of machine learning efforts, this can potentially be the end.

B. Particular Security Concerns with AI

Working with AI algorithms presents a unique set of challenges, including how models are integrated, trained, deployed, and utilized in industrial settings (see Figure 2).

The models are usually trained centrally on dedicated high-performance workstations or servers and transmitted to the application server following instruction and assessment. The data scientist and colleagues generate the data collection by referring to Figure 2, Step A, and extracting information from common sources including image archives, databases, and sensors. The data collection is examined for validity in Step B. Tasks including feature extraction, class assignments, annotation, and the incorporation of further domain-specific knowledge broaden the data set in this phase.

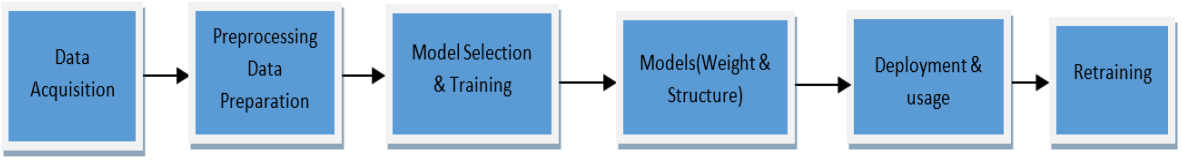

Figure 2: An example of a typical AI workflow in stages (A to F). Note: Usually, separate systems are used for the application and deployment phases

The data set is next subjected to a variety of models and AI architectures, which are subsequently assessed in Step C. The strategy or model with the best accuracy and resilience is typically applied. On high performance computers, training a model can take minutes or many days, depending on the data set. The completely trained model is the subject of Step D. The model's parameters indicate potential class memberships or clusters (the intelligence). The model's implementation and deployment are covered in Step E. This covers feature computation, forward propagation of the feature vector in the AI model, and data preparation procedures in the production environment. Since the program is hosted on an application server, more protection is typically needed. A disengaged Step F involves retraining the AI model with new features that have emerged either by expanding the use case or from more data collected while it was in use. Typically, only a portion of a network's higher layers undergo algorithmic adaptation rather than the entire system being retrained. We now think about the process depicted in Figure 2 against the backdrop of an industrial setting, such as a contemporary production line. Regarding the AI model's security, there are three possible outcomes. The adversary, such as a hostile insider, has introduced damaging information into the data set that does not fit the intended class in the poisoned data set scenario. This negatively affects and disturbs the AI structure following the training process. ML poisoning assaults, as a generic term, refer to data manipulation, such as using a fog environment, for the purpose of (re-)training ML models [42]. Due to a paucity of processing resources, fundamental AI training is carried out at a central processing system, particularly in edge systems. Next, the generalized model is applied to the edge system and modified utilizing smaller local data sets or calibration steps to meet the needs of the application. Due to the local collection of training data without expert supervision, this local adaptation process is vulnerable to attacks.

The following countermeasures can be used to stop ML poisoning attacks:

- Use of Secure Data Sources: Access to data sources must be limited to authorized persons and must be secure. The data must always be able to be examined and verified for accuracy.
- Data Sanitization: Before being utilized to train machine learning models, the data needs to be examined and cleansed. Any information deemed suspicious or out of the ordinary should be deleted. It might not be feasible to automate this, but manual labor is required.
    a. Anomaly detection: Any harmful data in the training data set can be found using anomaly detection techniques. The state of the art in AI anomaly detection can be attributed to auto encoders, generative adversarial networks (GAN), or recurrent neural networks (RNN).
    b. Ensemble Learning: This method involves training several machine learning models on various subsets of the data. As a result, opponents may find it more difficult to alter the data in order to influence the final prediction.
    c. Constant Monitoring: To identify any odd or unexpected results, the behavior of the ML model must be continuously observed. Any irregularities ought to be looked into and dealt with right away.

In the case of the compromised AI model, the an adversary modifies the AI network's training weights, or parameters, producing outputs that are not accurate (see Step D). Therefore, in order to prevent manipulation, security measures that guarantee the integrity of the data must be implemented.

Naturally, these actions must not impede the model's retraining (see Step F). It is possible to disable these safeguards during retraining, but doing so necessitates careful monitoring and security against unauthorized access. A trust management system that assesses the reliability of the data for re-training and identifies altered model parameters should be taken into consideration if this is not feasible or makes no sense, for example, because the re-training is automated.

The third scenario deals with the AI system's deployment, integration, and use within the setting of production (see Step E). The AI network's information inputs and outputs may also be compromised: either the network receives erroneous or noisy information (input from, say, altered sensors) or the outputs are fabricated and subsequently transmitted inaccurately.

This obstructs the subsequent production procedures in both scenarios. The production's statistical analysis can identify these kinds of attacks. The processing architecture might incorporate the aforementioned security measures in different places. For example, in a fog environment, edge systems supervise local information sources; before being integrated into central data sets, status information sent from edge nodes to central units needs to be verified.

Apart from the organizational difficulties associated with AI use Semantic issues also arise with algorithms: Abnormality detection is triggered by rapid changes in the application field caused by changing habits.

Adversaries operating in the network region may implement low-threshold modifications that compromise the anomaly detection procedure. Therefore, it's important to consider how sensitive these methods are.

C. Targeted Assaults on Linguistic Models

We concentrate on unique assaults against language models in this subsection. Let's say an attacker gains access to several ML model snapshots, like those found on predictive keyboards. The change in training data that was utilized to update the model can then be fully disclosed via these snapshots. We refer to this as a model update assault.

Information leakage in real-world applications where language models are often used was examined by Zanella-Béguelin et al. updated, for instance, by matching private data with publicly available, pre-trained language models, adding new data, or removing user data to comply with privacy regulations.

They are now able to carry out this type of leakage study unsupervisedly as they created two new metrics to assess the information leaking [43]. Language models that rely on auto completion are subject to tab assaults, in which the adversary tries to get the model to provide undesirable recommendations or outcomes. Thus, these attacks aim to compromise text recognition systems or test the consistency of language models.

This entails an attempt to knowingly introduce inaccurate or misleading information or produce distortions in the input data in order to trick the language model. Big language models are able to retain uncommon training samples, which, should the model be trained on private user content, presents grave privacy risks. A methodology has been devised by Inan et al. to verify language models for training data leaks.

This makes it possible for the model's developer to ascertain how much training data can be taken out of the model in a realistic attack. Additionally, the model's owner can confirm that the countermeasures that have been implemented function as anticipated, allowing their model to be used in a secure manner [44].

V. CONCLUSION & FUTURE WORK

We have demonstrated how AI applications are dependent on underlying cloud- or fog-based services in this study. Attacks on the networks of fog computing or cloud services that underpin the development of current AI applications will unavoidably occur in problems, security lapses, malfunctions, or issues with the AI applications.

The present hot issue of AI is creating a significant demand for services related to it. This forces individuals to either pay to get their data returned or sell it to the highest bidder, such as competitors. The relationship between AI and information security holds great promise for upcoming studies and applications. For example, the employment of AI techniques to support penetration testing and threat analysis of systems was not even mentioned in this paper.

Language models may already be used to create phishing emails that are tailored to a particular recipient. Owing to the enormous potential for combining AI with information security, we plan to engage in these activities going forward.

References

1) D. K. Sharma *et al.*, “Cloud computing in medicine: Current trends and possibilities,” in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, IEEE, 2021, pp. 1–5.

2) A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, “Fog computing: Principles, architectures, and applications,” in *Internet of Things: Principles and Paradigms*, R. Buyya and A. V. Dastjerdi, Eds., Morgan Kaufmann, 2016, pp. 61–75.

3) National Security Agency, “Cybersecurity information – cloud security basics,” National Security Agency, Aug. 29, 2018. [Online]. Available: https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/ professional - resources / csi - cloud - security - basics. pdf (visited on 06/08/2023).

4) F. Süß, M. Freimuth, A. Aßmuth, G. Weir, and R. Duncan, “Cloud security and security challenges revisited,” in *Proceedings of Cloud Computing 2019*, B. Duncan, Y. W. Lee, M. Westerlund, and A. Aßmuth, Eds., IARIA, May 2019, pp. 61–66.

5) R. Barona and E. A. M. Anita, “A survey on data breach challenges in cloud computing security: Issues and threats,” in *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2017, pp. 1–8. DOI: 10.1109/ICCPCT.2017.8074287.

6) F. Sabahi, “Cloud computing security threats and responses,” in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 245–249. DOI: 10.1109/ICCSN.2011.6014715.

7) M. Henriquez, “The top data breaches of 2021,” Security Magazine, Dec. 9, 2021, [Online]. Available: https://www.securitymagazine.com/

8) J. Fitzgerald, “The 10 biggest data breaches of 2022,” CRN Security News, Dec. 28, 2022, [Online]. Available: https : / / www. crn . com / news / security / the - 10 - biggest - data - breaches - of - 2022 (visited on06/08/2023).

9) Admin, “The 25 biggest data breaches and attacks of 2020,” Stealth- Labs, Dec. 16, 2020, [Online]. Available: https://www.stealthlabs.com/ blog/the-25-biggest-data-breaches-and-attacks-of-2020/ (visited on 06/08/2023).

10) D. Cameron, “Today’s massive ransomware attack was mostly pre- ventable; here’s how to avoid it,” Gizmodo, May 13, 2017, [Online]. Available: https://www.gizmodo.com.au/2017/05/todays-massive- ransomware- attack- was- mostly- preventable- heres- how- to- avoid- it/ (visited on 06/08/2023).

11) M. R. Watson, N.-h. Shirazi, A. K. Marnerides, A. Mauthe, and D. Hutchison, “Malware detection in cloud computing infrastructures,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13,no. 2, pp. 192–205, 2016. DOI: 10.1109/TDSC.2015.2457918.

12) A. K. Kibet, R. A. Esquivel, and J. A. Esquivel, “Ransomware: Ran- somware as a service (raas), methods to detects, prevent, mitigate and future directions,” *Journal of Emerging Technologies and Innovative Research*, vol. 9, no. 11, b264–b278, 2022.

13) Netscout, “Threat intelligence report, issue 7: Findings from 1h 2021,” Netscout, Tech. Rep. p. 7, 2021.

14) O. Yoachimik and J. Pacheco, “Ddos threat report for 2023 q1,” Cloudflare, Apr. 11, 2023, [Online]. Available: https://blog.cloudflare.com/ddos-threat-report-2023-q1/ (visited on 06/08/2023).

15) National Vulnerability Database, “Cve-2021-44228 detail,” National Institute of Standards and Technology, Dec. 10, 2021, [Online]. Avail- able: https://nvd.nist.gov/vuln/detail/CVE- 2021-44228 (visited on06/08/2023).

16) D. Nalley and V. Simonis, “Hotpatch for apache log4j,” AWS Open Source Blog, Dec. 12, 2021, [Online]. Available: https://aws.amazon. com / blogs / opensource / hotpatch - for - apache - log4j/ (visited on 06/08/2023).

17) hoakley, “Last week on my mac: When the internet caught fire,” The Eclectic Light Company, Dec. 12, 2021, [Online]. Available: https:/ / eclecticlight . co / 2021 / 12 / 12 / last - week - on - my - mac - when - the - internet-caught-fire/ (visited on 06/08/2023).

18) E. Yalon, I. Shkedy, and P. Silva, “Owasp api security top 2019,” Open Worldwide Application Security Project, 2019, [Online]. Available: https://owasp.org/www-project-api-security/ (visited on 06/08/2023).

19) Kubernetes, “Overview of Cloud Native Security,” Sep. 2022, [Online]. Available: https : / / kubernetes . io / docs / concepts / security / overview/ (visited on 06/08/2023).

20) Y. Wen and J. Wang, “Analysis and remodeling of the DirtyCOW vulnerability by debugging and abstraction,” in *Structured Object- Oriented Formal Language and Method: 9th International Workshop (SOFL+ MSVL) 2019, Shenzhen, China*, Springer, 2020, pp. 3–12.

21) L. Rice, *Container security: Fundamental technology concepts that protect containerized applications*. O’Reilly, 2020.

22) Amazon, “Shared Responsibility Model,” 2015, [Online]. Available: https://aws.amazon.com/en/compliance/shared-responsibility-model/ (visited on 06/08/2023).

23) T. Segura, “10 Rules for Better Cloud Security,” Dec. 2021, [Online]. Available: https://blog.gitguardian.com/10- rules- for- better- cloud- security/ (visited on 06/08/2023).

24) S. Arnautov *et al.*, “SCONE: Secure linux containers with intel SGX,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16)*, vol. 16, 2016, pp. 689–703.

25) GitGuardian, “Kubernetes hardening tutorial part 1: Pods,” Dec. 2021, [Online]. Available: https://blog.gitguardian.com/kubernetes-tutorial- part-1-pods/ (visited on 06/08/2023).

26) Cisco Systems, Inc., “Fog computing and the internet of things: Extend the cloud to where the things are,” Cisco Systems, Inc., Tech. Rep. C11-734435-00, 2015.

27) F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC ’12, Helsinki, Finland: Association for Computing Machinery, 2012, pp. 13–16, ISBN: 9781450315197. DOI: 10.1145/2342509.2342513. [Online]. Available:

https://doi.org/10.1145/2342509.2342513.

28) R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing: A taxonomy, survey and future directions,” in *Internet of Things*, Springer Singapore, Oct. 2017, pp. 103–130. DOI: 10 . 1007 / 978 - 981 - 10 - 5861 - 5 _ 5. [Online]. Available: https://doi.org/10.1007/978-981-10-5861-5_5.

29) Cisco Systems, Inc., “Cisco fog computing solutions: Unleash the power of the internet of things,” Cisco Systems, Inc., Tech. Rep. C11- 734589-00, 2015.

30) S. Khan, S. Parkinson, and Y. Qin, “Fog computing security: A review of current applications and security solutions,” *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–22, Aug. 2017. DOI: 10.1186/s13677- 017-0090-3. [Online]. Available: https://doi.org/10.1186/s13677-017- 0090-3.

31) A. Aljumah and T. A. Ahanger, “Fog computing and security issues: A review,” in *2018 7th International Conference on Computers Com- munications and Control (ICCCC)*, 2018, pp. 237–239. DOI: 10.1109/ ICCCC.2018.8390464.

32) M. Mukherjee *et al.*, “Security and privacy in fog computing: Challenges,” *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017. DOI: 10.1109/ ACCESS.2017.2749422.

33) F. Aliyu, T. Sheltami, and E. M. Shakshuki, “A detection and pre- vention technique for man in the middle attack in fog computing,” *Procedia Computer Science*, vol. 141, pp. 24–31, 2018, The 9th International Conference on Emerging Ubiquitous Systems and Per- vasive Networks (EUSPN-2018) / The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2018) / Affiliated Workshops, ISSN: 1877-0509. DOI: https : / / doi . org / 10 . 1016 / j . procs . 2018 . 10 . 125. [Online].

34) M. Hosseinzadeh, B. Sinopoli, and E. Garone, “Feasibility and de- tection of replay attack in networked constrained cyber-physical systems,” in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 712–717. DOI: 10.1109/ ALLERTON.2019.8919762.

35) A. M. Alwakeel, “An overview of fog computing and edge computing security and privacy issues,” *Sensors*, vol. 21, no. 24, p. 8226, Dec. 2021. DOI: 10.3390/s21248226. [Online]. Available: https://doi.org/10. 3390/s21248226.

36) S. Tu *et al.*, “Security in fog computing: A novel technique to tackle animpersonation attack,” *IEEE Access*, vol. 6, pp. 74 993–75 001, 2018. DOI: 10.1109/ACCESS.2018.2884672.

37) M. Al-khafajiy *et al.*, “COMITMENT: A fog computing trust man- agement approach,” *Journal of Parallel and Distributed Computing*, vol. 137, pp. 1–16, Mar. 2020. DOI: 10.1016/j.jpdc.2019.10.006. [Online]. Available: https://doi.org/10.1016/j.jpdc.2019.10.006.

38) K. Gu, X. Dong, and W. Jia, “Malicious node detection scheme based on correlation of data and network topology in fog computing- based vanets,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1215–1232, 2022. DOI: 10.1109/TCC.2020.2985050.

39) S. Yi, Z. Qin, and Q. Li, “Security and privacy issues of fog computing: A survey,” in *Wireless Algorithms, Systems, and Applications*, K. Xu and H. Zhu, Eds., Cham: Springer International Publishing, 2015, pp. 685–695, ISBN: 978-3-319-21837-3.

40) F. Dewanta, “Secure microservices deployment for fog computing services in a remote office,” in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2020, pp. 425–430. DOI: 10.1109/ICOIACT50329.2020.9332025.

41) R. Canetti and H. Krawczyk, “Analysis of key-exchange protocols and their use for building secure channels,” in *Advances in Cryptology — EUROCRYPT 2001*, B. Pfitzmann, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 453–474, ISBN: 978-3-540-44987-4.

42) Y. Qi, M. S. Hossain, J. Nie, and X. Li, “Privacy-preserving blockchain-based federated learning for traffic flow prediction,” *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021, ISSN: 0167-739X. DOI: https: / / Doi . org / 10. 1016 / j. future. 2020. 12. 003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0167739X2033065X.

43) S. Zanella-Béguelin *et al.*, “Analyzing information leakage of updates to natural language models,” in *ACM Conference on Computer and Communication Security (CCS)*, ACM, ACM, 2020. [Online]. Avail- able: https://www.microsoft.com/en-us/research/publication/analyzing-information-leakage-of-updates-to-natural-language-models/.

44) H. A. Inan *et al. (2021)*, *Training data leakage analysis in language models*, 2021. DOI: 10.48550/ARXIV.2101.05405. [Online]. Available: https://arxiv.org/abs/2101.05405.

45) J. Jain, S. K. Upadhyay, S. K. Nayak (2024), “Analyzing the Effectiveness of Machine Learning Algorithms in detecting Fake News”, in *International Conference on cutting edge technology in computing communication and Intelligence (ICCTCCI-2024), CRC Press Taylor and Francis.*

46) Upadhyay, S.K., Kumar, A. (2021). Early-Stage Brown Spot Disease Recognition in Paddy Using Image Processing and Deep Learning Techniques. *TS.38(6), pp. 1755-1766 .*

47) Upadhyay, S.K., Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. *Int. j. inf. tecnol. 14, 185–199.*

48) Rukhsar, Upadhyay, S. K. (2022). Rice Leaves Disease Detection and Classification Using Transfer Learning Technique*, ICACITE. Greater Noida, India, pp. 2151-2156*

49) Upadhyay, S. K., Kumar, A. (2022). An Accurate and Automated plant disease detection system using transfer learning-based Inception V3Model*. ICACITE. India. pp. 1144-1151.*

50) Rukhsar, Upadhyay, S. K. (2022). Deep Transfer Learning-Based Rice Leaves Disease Diagnosis and Classification model using InceptionV3*. CISES. Gr. Noida, India. pp. 493-499*

51) Jain J., Sahu, S.,Dixit, A.(2023),” Brain Tumor Detection model based on CNN and Threshold Segmentation” in *Int. J. of Exp. Res. & Rev.,,32 pp. 358-364 DOI: https://doi.org/10.52756/ijerr.2023.v32.031.*

52) Biswas, A., Agarwal, C., Gupta, S., Jain, J. (2023), "Early Phase Prediction of Chronic Kidney Disease Employing Machine Learning" *in 14th Int. Conference on Computing Communication and Networking Technologies, ICCCNT DOI: https://doi.org/10.1109/CEC.2018.8477876.*

53) Jain J., (2019),"Modern and Advanced Direction on Green Cloud" in *'IJEAT' Blue Eyes Intelligence Engineering & Sciences Publication (BEIESP), ISSN: 2277-3878, Impact Factor: 5.97, 9(2), pp. 3090-3095, https://doi.org/10.35940/IJEAT.F9184.129219*.

54) Jain, J., Singh, A., (2019) Structure of Cloud Sim Toolkit with Cloud" in '*IJEAT', Blue Eyes Intelligence Engineering & Sciences Publication (BEIESP), ISSN: 2249 – 8958, DOI: https://doi.org/10.35940/ijeat.F8918.088619, Impact Factor: 5.97, 8(6), pp. 4644 - 4649*

55) Jain, J., Singh, A., (2019), "Survey on Fog Computing and Cloud Computing" in *International Journal of Computer Sciences and Engineering (IJCSE) E-ISSN: 2347-2693, DOI: https://doi.org/10.26438/ijcse/v7i5.752756, Impact Factor: 3.022, 7(5), pp. 796 – 800.*

56) Jain, J., Singh, A., (2017) "A Survey on Security Challenges of Healthcare Analysis Over Cloud" in *"International Journal of Engineering Research & Technology" (IJERT) ISSN: 2778-0181, 6(4), pp-905-912. DOI: http://dx.doi.org/10.17577/IJERTV6IS040719..*