May 2022

AUTHENTICATED MACHINE LEARNING IN 5G NETWORK DEPLOYMENTS

Niranjan M M

Nagaraj Kenchaiah

AUTHENTICATED MACHINE LEARNING IN 5G NETWORK DEPLOYMENTS

AUTHORS:
Niranjan M M
Nagaraj Kenchaiah

ABSTRACT

Techniques are presented herein that employ a HyperLedger (i.e., a private blockchain) to allow authenticated machine learning (ML) components to interact with each other and sign messages (having pre-processed data) which are exchanged between the components to provide message authenticity. Along with authenticity, aspects of the presented techniques provide for the traceability and accountability of ML components. Aspects of the presented techniques may be employed across different operators and domains wherein a HyperLedger provides for the authenticity of the ML components. Further aspects of the presented techniques may be employed within an operator or within an instance (such as an Access and Mobility Management Function (AMF), a Session Management Function (SMF), etc.) wherein a centralized ML system or software can act as central repositories for all of the authenticated ML components along with their corresponding public key (which may be used for verification of the signed messages that are exchanged between the ML components). In such a case, the functionality of the HyperLedger may be performed by a centralized ML system or software.

DETAILED DESCRIPTION

As an initial matter, it will be helpful to confirm the meaning of a few terms that appear in the narrative that is presented below. Specifically:

Term	Meaning
AMF	Access and Mobility Management Function
DoD	Denial of Detection
DoS	Denial of Service
HyperLedger	A private, permissive blockchain-based authenticated ledger
IoT	Internet of Things

ML	Machine Learning
NFV	Network Function Virtualization
SDN	Software-Defined Network
SMF	Session Management Function
UPF	User Place Function

ML techniques are attempting to, and are expected to, solve many of the challenges in Third Generation Partnership Project (3GPP) fifth generation (5G) network deployments regarding improving the performance of the network and the services that use the underlying network as an enabler for such services. Additionally, the increasing complexity of communication networks due to a heterogeneity in networking equipment, end-user devices, applications, and services forces the automation of network operations (including, for example, minimizing manual configurations or human involvement; self-control; and adaptation and self-healing with changing user, service, and traffic requirements as well as dynamic network conditions) which, in turn, requires ML.

Within such a context, ML techniques may be used in multiple application areas in a 5G deployment. Those areas include:

- Infrastructure Management: ML may be used for intelligent load balancing, resource allocation (e.g., to manage peak traffic loads), the prioritization of traffic, etc.
- Network Operations: ML may be used to identify the optimal radio frequency (RF) parameters to optimize radio coverage and capacity based on a location, load, etc.
- Service Orchestration: ML may be used to allocate custom resources and functions to application-specific end-to-end slices.
- Assurance: ML may be used to analyze a network to identify and predict faults, and their root causes, as well as to allocate resources to recover from faults and to guarantee agreed upon service levels.
- Security Applications: ML may be used for anomaly- or signature-based intrusion detection as well as for realistic honeypot and vulnerability scanning.

However, ML techniques will also open a network to several serious security vulnerabilities (i.e., challenges) – such as, for example, an unfair use of resources, DoS, DoD, a leakage of private and confidential information, etc. – through the introduction of false data to a system (which is learning or operational), eavesdropping, the interception, or the modification of transmitted data by an adversary using compromised ML component, etc.

To illustrate the types of vulnerabilities that were described above, Figure 1, below, depicts various of the threats that may exist to ML in a 5G deployment.

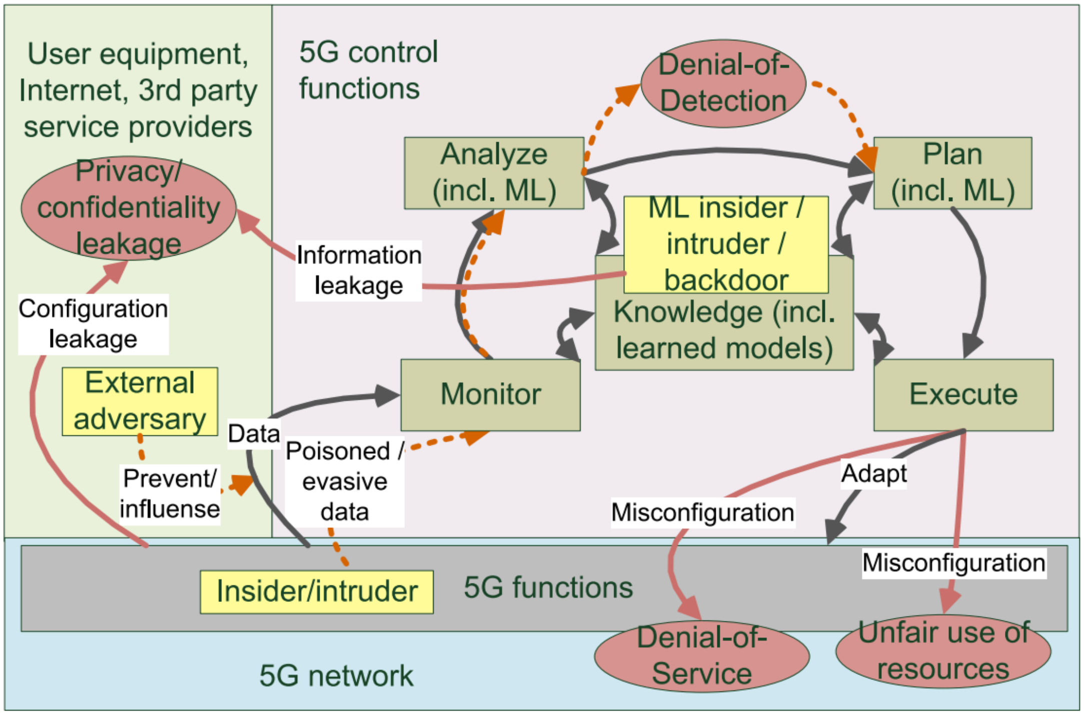


Figure 1: Exemplary Threats to ML in 5G Deployment

There are several reasons for the types of ML-induced security vulnerabilities that were described and illustrated above.

First, most of the learning in a ML context happens through data that is gathered from the environment. Such data may comprise un-scrutinized data which will have serious consequences on machines absorbing the data to produce actionable intelligence for the network. Such data may also comprise scrutinized data which will open privacy issues (challenges).

Second, most of the ML systems for 5G deployments are borrowed from other disciplines that provide excellent results in small, closed environments and deployments.

Third, in other words, the concepts of ML for 5G are being borrowed from existing mature technologies such as machine vision and robotics. Those ML concepts solve a problem that is under consideration, but they can introduce security challenges which include an inefficient use of network resources for gathering and disseminating the data, straining the processing and memory capabilities of different networked nodes, and unintentionally opening the network to the security vulnerabilities.

Fourth, ML algorithms and software may come from different suppliers (which may include open-source components) and an adversary or an attacker may already be aware of such components.

Each of the 5G deployment application areas that were described above face threats if any one of the ML components is compromised or integrated with a malicious third-party (or even open source) component or application (as an attacker is already aware of the open source and/or the third-party algorithms and methods that are used in ML).

While some existing solutions may provide authentication, security (e.g., access controls, a firewall, or physical security), and trust establishment between 5G network entities, there are multiple ML use cases where these are not sufficient to entangle threats that may be induced by compromised or malicious ML component and applications (as will be described below).

Accordingly, a mechanism is needed to authenticate each ML component (and the communication that takes place between the components) while gathering data, processing the data, sharing the data across ML components, and predicting and taking a final decision to solve a specific problem (which could be intelligent load balancing, predicting the faults, prioritizing the traffic, resource allocation, etc.).

Techniques are presented herein, which will be described and illustrated in the below narrative, that address the need that was described above.

As described previously, ML techniques may be used in different 5G network deployments. ML may be used in the access network to increase spectral efficiency or for other intelligent uses of radio resources. ML may be used in the edge (near the access network) to intelligently serve latency-critical services by providing higher resources in the edge and to Internet of things (IoT) devices. ML may be used in the backhaul or transport network for traffic classification or for improving network management with the help of a software-defined network (SDN). ML may also be used for improving the performance of cloud-based services.

However, the use of ML techniques will open a network to several serious security vulnerabilities and challenges (through the introduction of false data to a ML system (which is learning or operational), through eavesdropping, and through the interception and modification of transmitted data (i.e., masquerading) by an adversary using a compromised ML component) such as DoS, DoD, a lack of confidentiality (e.g., a leaking of private and confidential information), a lack of privacy (e.g., a leaking of privacy information), and an unfair use of resources.

As noted previously, while some existing solutions may provide authentication, security (e.g., access controls, a firewall, or physical security), and trust establishment between 5G network entities, there are multiple ML use cases where these are not sufficient to entangle threats that may be induced by compromised or malicious ML component or applications. For example, data sources may be administered in different domains by different entities resulting in their trustworthiness becoming difficult to determine by centralized ML components. Further, misbehaving user equipment (UE) may input malicious data for ML functions which utilize information from the UE ML components. Additionally, open-source components (which are known by adversaries who are practicing attacks) may be integrated with ML algorithms and software. Still further, ML components may be from different suppliers (these are also called white-box attacks).

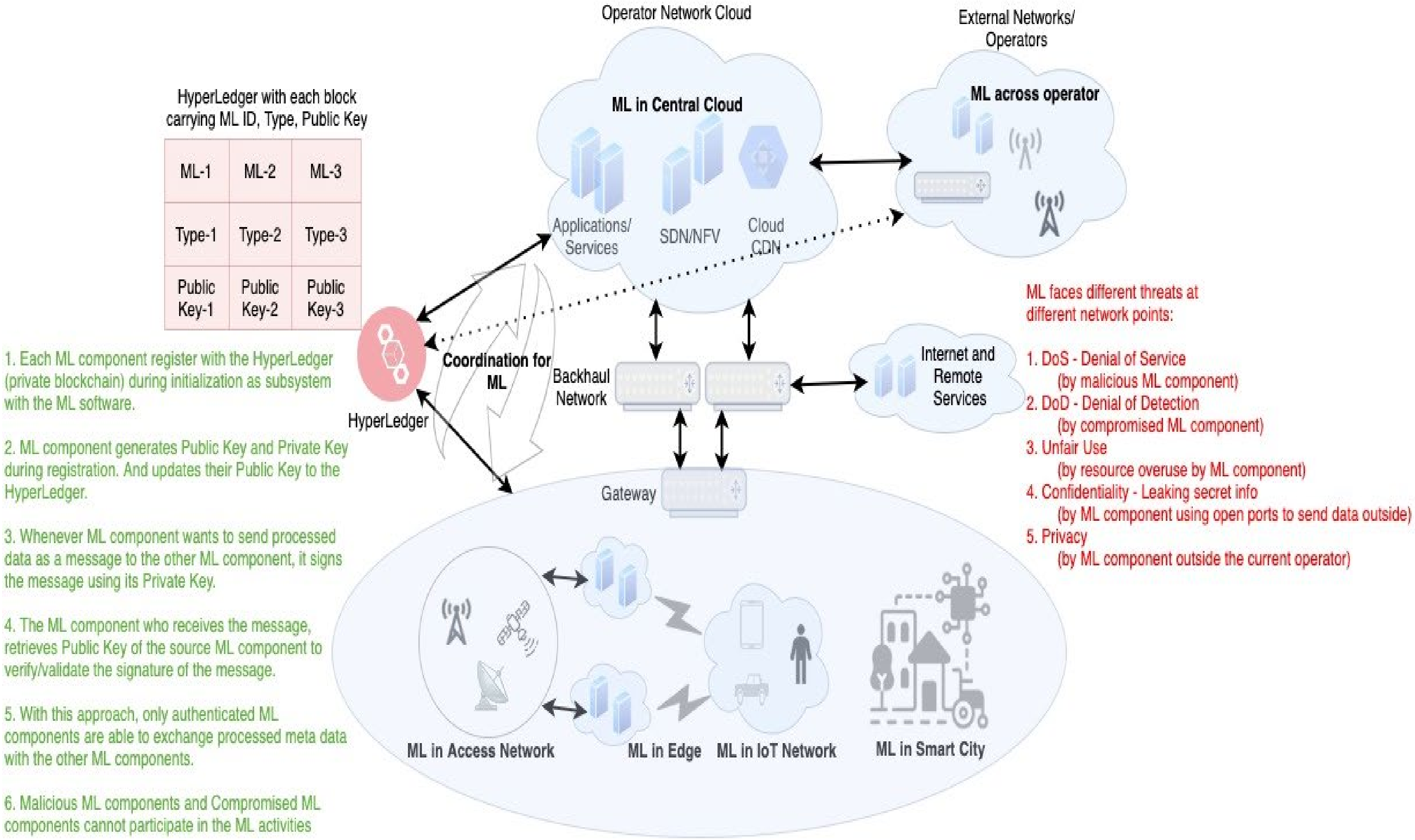
Consequently, as described above, a mechanism is needed to authenticate each ML component (as well as the communication that takes between the components) while gathering data, processing the gathered data, sharing the data across ML components, and predicting and taking a final decision to solve a specific problem (which could be intelligent load balancing, predicting the faults, prioritizing the traffic, resource allocation, etc.).

Techniques are presented herein that address the need that was described above using a HyperLedger (i.e., a private blockchain) to enable authenticated ML components to interact with each other and sign the messages (having preprocessed data) that are exchanged between them to provide message authenticity. Along with authenticity, aspects of the presented techniques provide for the traceability and accountability of ML components.

According to aspects of the presented techniques, each ML component may register with a HyperLedger (i.e., a private blockchain) during initialization as a subsystem with a ML system or software. Further, a ML component may generate a public key and a private key pair during registration (i.e., initialization) and then update their public key to the HyperLedger. Additionally, whenever a ML component wishes to send processed data as a message to another ML component it may sign the message using its private key. Still further, an ML component that receives a message may retrieve the public key of the source ML component to verify and validate the signature of the message.

Using the presented techniques, only authenticated ML components can exchange processed data with the other ML components. Malicious ML components and compromised ML components cannot participate in the ML activities.

Figure 2, below, presents elements of an exemplary arrangement according to aspects of the techniques presented herein that is reflective of the above narrative.



HyperLedger with each block carrying ML ID, Type, Public Key

ML-1	ML-2	ML-3
Type-1	Type-2	Type-3
Public Key-1	Public Key-2	Public Key-3

1. Each ML component register with the HyperLedger (private blockchain) during initialization as subsystem with the ML software.

2. ML component generates Public Key and Private Key during registration. And updates their Public Key to the HyperLedger.

3. Whenever ML component wants to send processed data as a message to the other ML component, it signs the message using its Private Key.

4. The ML component who receives the message, retrieves Public Key of the source ML component to verify/validate the signature of the message.

5. With this approach, only authenticated ML components are able to exchange processed meta data with the other ML components.

6. Malicious ML components and Compromised ML components cannot participate in the ML activities

ML faces different threats at different network points:

1. DoS - Denial of Service
   (by malicious ML component)
2. DoD - Denial of Detection
   (by compromised ML component)
3. Unfair Use
   (by resource overuse by ML component)
4. Confidentiality - Leaking secret info
   (by ML component using open ports to send data outside)
5. Privacy
   (by ML component outside the current operator)

Figure 2: Generic 5G Network Architecture Using Authenticated ML

Aspects of the techniques presented herein may be used across different operators or domains wherein a HyperLedger (i.e., a blockchain) provides authenticity for the ML components.

Further aspects of the presented techniques may be used within an operator or within an instance (such as an AMF, a SMF, etc.), wherein a centralized ML system or software can act as central repositories for all the authenticated ML components along with their corresponding public key (which may be used for verification of the signed messages that are exchanged between the ML components). In this case, the functionality of the HyperLedger is performed by a centralized ML system or software.

For simplicity of exposition, the following description of the techniques presented herein is divided into four parts, each of which will be described in detail below. A first part encompasses the identification and authentication of the ML components. A second part encompasses the authentication of the messages and data that is exchanged between ML components. A third part encompasses the periodic updating of ML status data and statistics to the HyperLedger. A fourth part encompasses traceability and accountability.

The first part, which encompasses the identification and the authentication of the ML components, employs an identifier (ID), a public key (PK), a private key (SK) and a type (TYPE) for each ML component. As part of the initialization (e.g., the registration) of a ML component with a system (e.g., a ML application or software), a PK and SK would be generated by the ML component. The SK is securely stored locally (within the component) and the PK is shared with other ML components by adding it into the distributed HyperLedger (i.e., a private Blockchain). The generated PK and SK may be used later to re-authenticate or re-register with the blockchain.

The second part, which encompasses the authentication of the messages and the data that is exchanged between ML components, employs the SK of a ML component to sign a message (comprising processed data) by the ML component (e.g., a source ML component) before sending the message to the other ML component (e.g., a target ML component). A target ML component that receives the message from a source ML component retrieves the PK of the source ML component from the HyperLedger. That PK may then be used to validate the signature of the message and, in turn, verify the authenticity of message that was sent by the source ML component.

The third part, which encompasses the periodic updating of ML status data and statistics to the HyperLedger, includes a ML component periodically updating the HyperLedger with information about the ML's status and statistics (e.g., the number of errors (both known and unknown), the percentage of correct predictions, the number of messages that were received by and sent to the other ML components, etc.). Such information aids in traceability and accountability.

The fourth part, which encompasses traceability and accountability, provides information that can assist operators with troubleshooting an issue (e.g., when something breaks down or suffers from abnormal behaviors). Aspects of the presented techniques enable the traceability of ML components, along with their status and statistics, which helps in capturing system behavior.

Within a running ML component, if the component suffers abnormal attacks (such as, for example, receiving message from a compromised ML) the attack processing is also logged as transactions. With such logged transactions of attack trajectories, any future attacks that may be launched on the ML component may be identified (using attack pattern recognition).

It is important to note that any number of existing blockchain-related processing mechanisms (including, for example, sharding methods, etc.) may be applied to the techniques presented herein to scale those techniques to extremely high transaction levels (that are analogous to, for example, the transaction levels of large credit card processing facilities).

Use of the techniques presented herein offers several advantages. For example, aspects of the presented techniques enable an authenticated ML component to rightly predict the future and intelligently take appropriate actions. Further, aspects of the presented techniques may be used across different operators and domains wherein a HyperLedger (i.e., a blockchain) provides for the authenticity of ML components. Additionally, aspects of the presented techniques may be used within an operator or within an instance (such as an AMF, a SMF, etc.), wherein a centralized ML system or software can act as a central repository for all of the authenticated ML components along with their corresponding public key (which may be used for verification of the signed messages that are exchanged between the ML components) wherein the functionality of the HyperLedger is performed by a centralized ML system or software.

Aspects of the techniques presented herein may be employed under several different use cases. For example, aspects of the presented techniques are applicable to IoT environments (such as a smart city), an automotive IoT environment (such as a connected car), etc. wherein ML is an important part of the 5G network deployment. Additionally, aspects of the presented techniques are applicable to the use of ML capabilities in remote services and cloud deployments.

In summary, techniques have been presented that employ a HyperLedger (i.e., a private blockchain) to allow authenticated ML components to interact with each other and sign messages (having pre-processed data) which are exchanged between the components to provide message authenticity. Along with authenticity, aspects of the presented techniques provide for the traceability and accountability of ML components. Aspects of the presented techniques may be employed across different operators and domains wherein a HyperLedger provides for the authenticity of the ML components. Further aspects of the presented techniques may be employed within an operator or within an instance (such as an AMF, a SMF, etc.) wherein a centralized ML system or software can act as central repositories for all of the authenticated ML components along with their corresponding public key (which may be used for verification of the signed messages that are exchanged between the ML components). In such a case, the functionality of the HyperLedger may be performed by a centralized ML system or software.