

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/354832826>

Secure Federated Learning for Cybersecurity by Virtue of Network Cooperation

Preprint · September 2021

CITATIONS

0

READS

466

3 authors:



Michael Enright

Quantum Dimension, Inc.

24 PUBLICATIONS 125 CITATIONS

SEE PROFILE



Eman Hammad

East Texas A&M University

68 PUBLICATIONS 1,124 CITATIONS

SEE PROFILE



Ashutosh Dutta

Johns Hopkins University

31 PUBLICATIONS 401 CITATIONS

SEE PROFILE

Secure Federated Learning for Cybersecurity by Virtue of Network Cooperation

Michael A. Enright
Quantum Dimension, Inc.
Huntington Beach, California, USA
menright@qdimension.com

Eman Hammad
Computer Science & Engineering
Texas A&M University - Commerce, RELLIS
College Station, Texas, USA
eman.hammad@tamu.edu

Ashutosh Dutta
Applied Physics Laboratory
Johns Hopkins University
Laurel, Maryland, USA
Ashutosh.Dutta@jhuapl.edu

Abstract—Many of the Artificial Intelligence and Machine Learning (AI/ML) systems of today tend to be operated by single entities such as companies, universities, research institutions and others. Consequently, they tend to have stove-piped design where the ML models, training data, etc. are housed in a single location, or at least, under control of one entity. This non-distributed approach has advantages in some ways, such as security, but disadvantages in others, such as computing power, integrating outside entities and more. An important distributed use case is 5G cellular and, more generally, future cellular networks, often referred to as Beyond 5G (B5G). These networks provide enhanced capabilities for edge computing and Internet of Things (IoT) applications. With many new applications and nodes in the network, security is critical and is in need of a real-time dynamic security architecture that contains the processing capability to handle large amounts of network and other security related data, such as continuous modeling with digital forensics. By sharing data across the network, greatly improved and predictive security is possible. A distributed autonomous network architecture that supports Federated Learning (FL), where nodes share ML duties, is needed. Not only must FL network process the data, but internally, it must attain the highest level of security to ensure that the network is safe. This can be achieved by integrating the tenets of a Zero Trust Architecture (ZTA) into the system architecture. The objective of this work is to lay the foundation for a solution that can meet this need by employing core capabilities of today, i.e. cloud computing, distributed computing, and open systems and software.

- **Impact Statement** A dynamic security architecture will have the ability to handle both current and future threats, whether the application is future networks or a single enterprise. A framework is needed that will enable technologies to be developed to support this need. One example is network monitoring with ML and sharing training datasets between nodes. However, network monitoring is not the only area that needs FL and an adaptive architecture. Real-time digital forensics, e.g. monitoring compute and storage resources, is another area that must be integrated into the dynamic FL architecture. With the proposed architecture, new techniques and ML models can be developed for enhanced real-time security.

Index Terms—machine learning, distributed machine learning, federated learning, security, cybersecurity, trust, privacy, zero trust architecture

I. INTRODUCTION

Network cooperation provides many more capabilities and can address many more problems than non-cooperative networks. While this has been show to be true in both technical and non-technical environments, the most obvious technical

case is the Internet, but other recent examples clearly exist. Mobile ad-hoc networks (MANETs), cooperative navigation networks [1] to provide resilient navigation are just some examples. Even information theory with Shannon water-filling [2] can be seen as an example of the benefits of many channels, or in this case network elements, or a homogeneous single channel. Here, "cooperation" is achieved by optimizing, say bit rate, subject to an energy constraint as in Eq. 1. Here, the Lagrangian, formed by including an objective (cost) function and the energy constraint [3], is

$$J = \sum_{i=1}^M Q \left(\sqrt{\frac{2E_i}{N_i}} \right) W_i + \lambda \sum_{i=i}^M E_i, \quad (1)$$

where E_i is the energy, $Q \left(\sqrt{\frac{2E_i}{N_i}} \right)$ is the symbol error probability for channel i .

Federated Learning is one of the most recent examples where network cooperation provides benefits that are difficult to address for a single entity - powerful distributed computing. In the same manner as is used for multicarrier communication, a new cost function that is based upon compute capacity at a node with the constraint being wait time or some other metric can be created. This is also the case in ML techniques such as in supervised neural networks where a cost function is optimized subject to a set of constraints. With regard to FL, network cooperation opens up new set of possibilities.

A critical challenge in network cooperation, regardless of application, FL, etc., is cybersecurity. Every day, there is news of one attack or another, and as the network aperture expands, so do the threats. This will continue to be the case - as B5G evolve, so too will security threats to these networks. Consequently, it is critical to have a dynamic security component that is a core element of the cooperative network, one that has the ability to evolve over time to counter existing and future threats. Equally as important is the fact that this security architecture must operate autonomously, or at the very least semi-autonomously, in order to mitigate security threats in the most time-efficient manner.

Federated learning is being examined for different applications. It was used in [4] to improve communication efficiency. A good summary of challenges and future directions of FL for intrusion detection systems (IDS) can be found in [5].

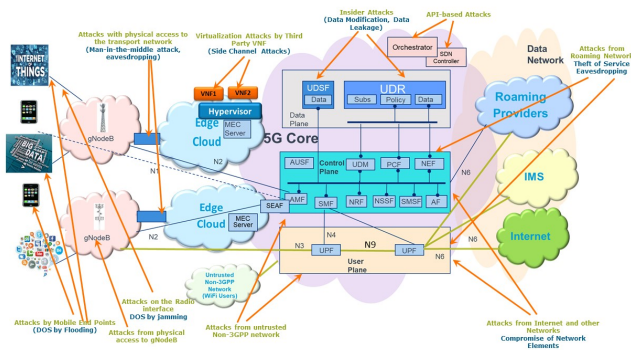


Fig. 1. Attack surface of a 5G network [10].

As demonstrated in this work FL for IDS has been an active research area. Overall, a good summary of AI/ML for different cyber attacks can be found in [6].

Here our intent is not to develop new algorithms, but instead, to develop a framework where the best techniques can be plugged in for dynamic, predictive cybersecurity. This is no easy challenge since network data is very dense and attacks may occur over a large time period. In [7], an attempt was made to estimate the timing of infected hosts. A comprehensive ML approach was developed in [8] by working with data from an enterprise Security Operations Center. A tremendous amount of data had to be processed which could not be performed in real-time for their ML model. This paper illustrates the challenges of developing a real-time security system. However, with computing capability and network cooperation, secure FL can be achieved to protect the network.

Whereas prior work in FL has been more algorithmic, the objective of this research is to demonstrate an architecture that can be used to implement current and future FL capabilities across multiple applications, not just security. However, predictive security and security policy implementation are important needs of 5G/B5G networks, where the 5G security requirements are defined in [9]. These networks are a critical use-case because they include a wide spectrum of components that include small Internet of Things (IoT) devices to edge computing server to cloud computing servers, each of which has its own computing capabilities and security needs. A comprehensive system architecture is required to support such diverse elements.

A. Future Network Security

For the purposes of this work, the use case for network cooperation is 5G/B5G network security, although the architecture can be applied to other areas that can utilize autonomy such as autonomous communication link allocation. The key here is that our focus is on real-time, dynamic environments, which is exactly the nature of dynamic, predictive network security that uses digital security forensics [11] to adapt and train continuously.

The need for autonomy for in 5G and beyond is evident from Fig. 1. In this figure, a 5G network and corresponding

threat vectors are illustrated. From a traditional network security perspective, security functions must be integrated into the network at multiple points - mobile and IoT devices, edge platforms, radio access network (RAN), clouds and the Core Network (CN).

Within these system elements, there is an even greater security vulnerability when one considers that fact virtualization, specifically network function virtualization (NFV) along with disaggregated open systems, are becoming more prevalent. With virtualization, the security risk is greater simply because there is the potential for many more networked devices that must be secured. Open systems, such as Open RAN (O-RAN) and Cloud (C-RAN), have exposed the internal workings of the radio architecture to potential attackers and thereby become a vulnerability. Any device that connects to the network is a potential source of a vulnerability, and with such a wide breadth of potential threat vectors, intelligence and autonomy are crucial. Network intrusion detection has been a topic of extensive research in the security domain. However, this is not the only threat to security and privacy. There are email attacks, computer firmware attacks, ransomware, and others. An architecture is needed that can accommodate a diverse set of autonomous and predictive security algorithms.

AI/ML techniques have rapidly evolved over the past decade due to the continued evolution of powerful, low cost computing resources. While many of the core capabilities of AI/ML have been present for decades, further development of graphics processing units for general-purpose computing, along with corresponding software libraries for AI/ML, such as Google's TensorFlow and others, have greatly increased the adoption of AI/ML based systems. Given the ability of AI/ML systems to learn and adapt in real-time operations, they will be crucial protecting against both known (past and current) and unknown (future) threats. The potential of zero-day attacks, or at least the destruction that they can cause, will be minimized with a dynamic AI/ML-based security architecture.

Autonomy is critical for security operation of advanced networks, such as 5G and Future Networks. Consequently, the focus of this document is the security of 5G/B5G and how AI/ML can be utilized to secure the network from current and future adversarial attacks. AI/ML algorithms alone will not address this problem; the security architecture must have a means to both orchestrate the AI/ML security functions. To do so will require an open system to disseminate, process and update critical information that includes, but is not limited to, current security threat vectors and levels, changes in AI/ML models and associated parameters. The result will be a dynamic system where real-time security situational awareness can be achieved, thereby necessitating real-time management, or orchestration, in order to operate efficiently and effectively against current and future threats. Some of the core elements of such a system are:

- **Device and Edge Platform Security Functions** – Threats that come from the local area that have the potential to become wider network attacks must be mitigated.

- **Network Security Functions** – Threats to the network that come from the Internet will propagate through Future Networks.
- **Supervised and Unsupervised AI/ML Algorithms** – There is no single solution to cybersecurity, so all tools must be utilized, and the ecosystem must support many types of AI/ML models.
- **Open Interfaces** – Interfaces must be specified so that new technologies can be implemented seamlessly and will provide real-time situational awareness.
- **Threat Vector Sharing** – Share threat vector information in real-time with other models.
- **Online Training** – Models must train online and update at appropriate intervals.
- **Live Updates** – Models must be updated in real-time to mitigate security threats and limit the effects.
- **Dynamic Model Generation** – If the current model cannot mitigate the threat, a new model should be developed in-line to mitigate the threat.
- **AI/ML Security Orchestration** – Coordination between all elements of the AI/ML ecosystem must take place.

The next step is to ensure that the system itself is resilient against attacks by using a framework that addresses important security tenets.

B. Zero Trust Architecture

Zero Trust (ZT) was recently promoted [12] as a promising approach that in principle presents a shift in complex systems security. ZT acknowledges slowly eroding perimeters and a wider threat landscape comprising external and internal threat actors. ZT frameworks adopt three key principles; to always verify explicitly, grant access to resources on least privilege basis, and to assume breach. By means of those principles, ZT puts very little assumptions (if any) that could often cause gaps in current traditional security approaches. For example, traditional approaches seek in effect to white-list or black-list active entities be it a user, application or flow; however assumptions such as known applications or trusted devices could lead to bypassing security controls resulting in systems' vulnerabilities to several threat vectors [13]. ZT highlights the need to always challenge those assumption as any entity could be exploited at any point in time. Hence, ZT implementations need to be thought through carefully to enable effective remediation and mitigation combining capabilities for authentication, authorization, access control and active monitoring [12], [13].

Security of FL systems presents a unique challenge because of the complexity, sensitivity and particular nature of its sub-components. Future networks are one example of critical systems that will heavily rely on FL to both 1) operate the systems at a high level of efficiency, such as through network autonomy, and to 2) enable its self-adaptive real-time security capabilities limit both the number and magnitude of threats. This motivates investigating how a secure-by-design approach can be developed for FL taking into account ZT principles thereby enabling a unified and structured approach.

Equally important in this endeavor is establishing the required architectural components essential to enable this treatment and inform possible implementations.

The intent of this work is not to define a stringent set of requirements, thereby limiting design choices. Instead we propose an architecture that has the structure to enable advanced FL capabilities across many types of networks that includes 5G/B5G, enterprise and even across the Internet. In essence, this work is intended to support new and advanced, secure AI/ML algorithms and technologies across an array of use cases.

C. Organization of this Work

In Section II, we address threat modeling and analysis for 5G and future networks. Section III describes our concept of our open, cooperative network architecture for FL. Section IV presents a Zero-Trust security architecture implementation to secure the FL network. Finally, future development that includes risks and gaps are presented in Section V.

II. THREAT MODELING & ANALYSIS

In this section, we expand on the cybersecurity challenges of distributed AI/ML. This understanding is essential to enable subsequent discussions in this paper. A distributed AI/ML algorithm can be viewed as a multi-phase system where data is used to train a model that's part of a production architecture [14]. With this perspective, we can proceed to describe threats and relevant security controls as it relates to: 1) data, 2) model, and 3) architecture within the machine learning life-cycle.

A. Threat Taxonomy

Hence, we can summarize the main security challenges in this context as follows:

- **Infrastructure Hardware and Software Security** - Vulnerabilities in hardware, platforms and applications can be exploited by threat actors to gain access and manipulate the data and/or models.
- **Data Integrity** - Adversarial threat actors can inject malicious data in the training stage to affect the inference capability of AI models or add a small perturbation to input samples in the inference stage to change the inference result [14].
- **Data Privacy** - In applications where users provide their data, an adversarial threat actor can repeatedly query a trained model or intercept data exchange between the distributed learning components to infer private information.
- **Model Confidentiality** - A knowledgeable adversary may be able to create a clone model using inferences obtained through a number of queries against the original model.
- **Model Security and Robustness** - In a FL network, model parameters are updated and shared across nodes. Model updates must be done in a secure manner. In addition, the model must be secure against adversarial attacks, such as those current being developed using Generative Adversarial Learning (GAN) [15], [16] concepts.

Given such challenges, a ZT approach for FL requires a clear definition of what comprises an entity and a distinction of applicable threats to each entity type in addition to applicable mitigation approaches that this paper aims to generalize. Based on the standard ML life-cycle we can decompose FL into three entity types 1) data, 2) model and 3) architecture. The combination of those entities will cater to the different classes of applications and their requirements. However, this will remain helpful as traverse the threat vectors against each entity.

Data can be targeted through manipulation. This can be characterized by different attack vectors. In evasion attacks, carefully designed variations of input data are used to drive the model outcomes away from the main objective. For example, to cause a model to overlook anomalies. In poisoning attacks, mixing a small percentage of manipulated data with the rest of the data can be used to significantly impact the model accuracy. Finally, data can be a target by attack vector targeting confidentiality and privacy. Current approaches to mitigate those threats utilize data or algorithms. Data can be augmented with adversarial data samples to train the model on detecting and handling evasion attacks. Similarly, filtering and regression can be utilized to address poisoning attacks, and finally algorithms such as differential privacy can be leveraged to defend against data breach threats.

Models can be targeted in inference attacks to extract the trained model constituting intellectual property theft, and could further be used to generate data that can be leveraged against the original model. Several references aimed to discuss the complexity of potential defenses given the blackbox nature of some of the models. Most recent works have aimed to tackle this through the developments of enhanced model security. Model security can be improved through model 1) detectability, 2) verifiability, and 3) explainability.

Architectures of FL exposes it to a multitude of threat vectors that can be mapped to threat vectors against distributed systems. Threat actors would aim towards not only data and model vulnerabilities but also against the architecture. For example, if the FL model has a centralized global entity, it could be targeted to decimate decision making or shed doubts on the global situational awareness.

B. Example Risk Scenarios

We next discuss a few example risk scenarios based on 5G use cases with FL implementations. This will be considered for two use-cases 1) network security function virtualization for massive Machine Type Communications (mMTC), and 2) operational intelligence to support Ultra Reliable Low Latency Communication (URLLC) applications.

- **mMTC Applications** - The 5G mMTC model supports a large number of low data rate sensors that can be on the order of millions. These can be employed in manufacturing and warehousing applications, smart city application such as road sensors, parking meters, etc. Because they are part of the network, each of these has the ability to attack the network. These can be used in distributed denial of service attacks (DDOS) and

others. With these approach described in the following, these attacks can be mitigated by smart algorithms that reside on the edge network that communicates with these sensors or at the Radio Access Network (RAN) if the devices communicate directly two it.

- **URLLC Applications** - This model requires resilient, assured delivery since these applications include medical, public services, etc. Disruption of network security by using jamming, DDOS, man-in-the-middle and other attacks can have grave circumstances for users of these services. Hence, recognition of the network security state is paramount to routing these services around disruption points.

III. TOWARDS AN OPEN SYSTEMS ARCHITECTURE

An open systems architecture (OSA) is a critical for secure FL systems of the future due to their flexibility and open source nature. These can be seen across different applications that include the Future Airborne Capability Environment (FACE), NVIDIA's DriveWorks for autonomous driving and others, which are good examples of architectures that are inherently modular so that they can support distributed applications and computing, while also being modular so that they are extensible for future development.

The FACE standard is published by The Open Group in conjunction with the United States Navy. As its name implies, it was originally intended for use within military aircraft, but its target environments have since been expanded. The need for such a standard arose from the difficulty of adding new sensor capabilities, e.g. GPS receivers, radios, etc., to military aircraft due in part to the proliferation of proprietary vendor solutions that did not communicate with each other. The objective was to develop standard interfaces that allows the devices to communicate and share information. In essence, the sensor hardware could be extracted away, and application software could be developed to collect data from across the network for a particular need. A functional block diagram of the FACE framework is shown in Fig. 1. FACE is comprised of a set of segments that implement different functions of the protocol stack, while the interfaces are flexible to allow for different sensor capabilities.

In the same way the FACE allows sensors to communication in an aircraft, a cooperative security architecture will allow network components to communication information in a number of scenarios, such as device to device, device to a security application or device to central controller. This would not require that each network devices becomes part of large-scale software program, but rather, each device operates in a distributed fashion with open/published interfaces. In this way, network devices can share real-time security information, such as security analytics, AI/ML models and associated parameters, network status updates and more. While the interfaces are open, security functions will be implemented to ensure that information transmitted is both secure and trustworthy. Future research will need to address both of these within the context of an distributed architecture development.

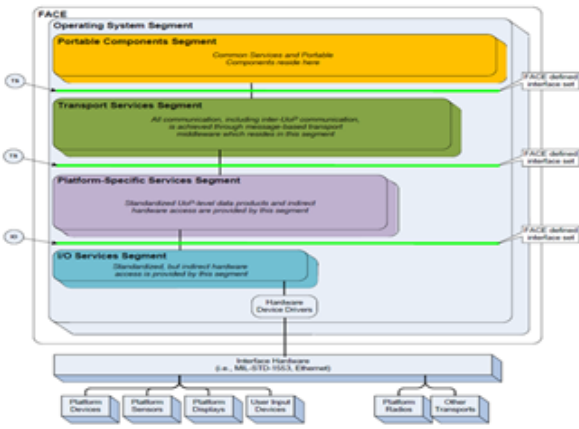


Fig. 2. FACE architectural segments [17]

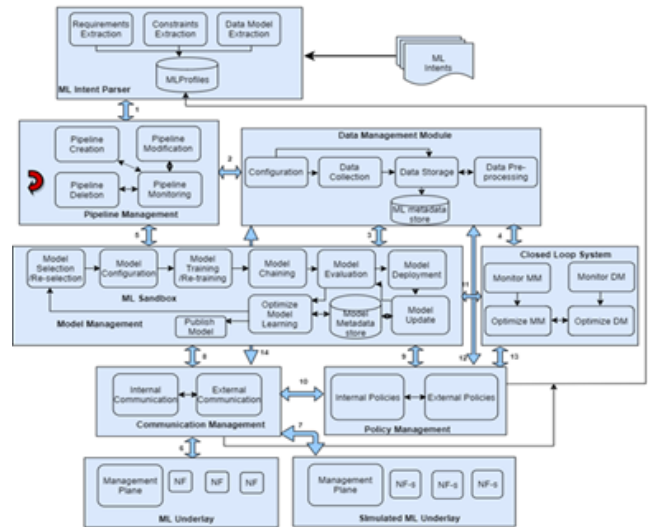


Fig. 4. Architecture of the MLFO [19].

A 5G network is significantly more complex than that found in a sensor network. For future autonomous networks, many specific AI/ML models will need to be employed to mitigate specific threats that include DDOS, jamming and spoofing and others. Future networks security is not simply an AI/ML model, but rather, an AI/ML system that can be managed to mitigate many different types of threats.

A. Machine Learning Orchestration Framework

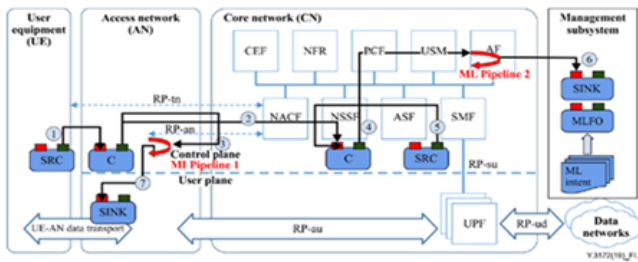


Fig. 3. IMT2020 machine learning architecture [18]

The International Telecommunications Union (ITU), via its Focus Group on Machine Learning for 5G (ML5G), has developed an ML orchestration framework as illustrated in Fig. 3. This high-level architecture describes the relationships between the different ML components with the 5G network. Here, ML Pipeline 1 and ML Pipeline 2 refer to predictions made at the particular network component. For example, ML Pipeline 1 ingests user equipment (UE) data that is used to make predictions at the AN. Likewise, UE data can be combined with CN data that is output via path 4 to ML Pipeline 2 to make CN predictions, which are subsequently sent to the management subsystem.

Within this framework, the ITU has defined the ML Function Orchestrator (MLFO) to be the entity that manages and orchestrates the ML pipelines. A block diagram of the MLFO is shown in Fig. 4. The MLFO is similar in structure to other

orchestration frameworks. For example, there are currently standards in place for network function virtualization (NFV). In addition, the NFV associated requirements have been translated into open source software that includes OpenStack, OpenMano, Tacker and others. To date, there is no open source standard for MLFO, and as a result, this is critical area for future development.

B. Layered View of the Secure FL Architecture

Cloud computing and software-defined networking (SDN) are two areas that are ripe in terms of research and yet mature in terms of implementation. Commercially, both areas have experienced tremendous growth over that past 10-15 years, although the basic tenet of distributed component is not new. What this revolution has brought is software, e.g. OpenStack and others, and a computing architecture that are suitable elements. Starting with the distributed aspect of cloud computing and the management and control aspect of SDN, we propose an architecture, in terms of layers and components, that is suitable for securing distributed machine learning applications of today and in the future.

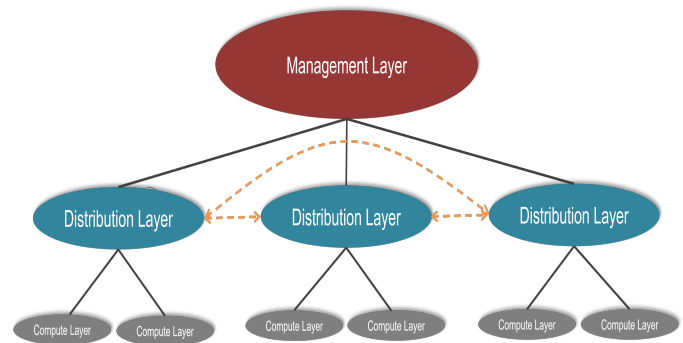


Fig. 5. Layered view of the proposed the secure FL architecture.

Our secure FL architecture is described in terms of layers, as shown in Fig. 5, and components, as shown in Fig. 6. The layers illustrated in the figure demonstrate the system-level functionality, i.e. how the system operates as a whole rather than functions that are assigned to different entities within the network. For example, there is a management component at the control node but the same component does not exist at client nodes. In this way, our layered model is not the same as the TCP/IP or OSI reference models that require the same layers at both a source and destination. The Management Layer is more similar to a control node in SDN.

The **Management Layer** is responsible for network management and security. Some of the network management aspects include:

- Adding and removing nodes from the network
- Contains list and their locations of AI/ML data files, models etc.
- Manages AI/ML model training and updates
- Coordinates traffic among network nodes
- Ensures fault tolerance of network in a CC fashion
- Ensures "authenticity" and "validity" of model updates

The security aspect of the Management Layer has different responsibilities that are split across the control and client nodes which includes:

- Orchestration of security functions throughout the network
- Contains AI/ML models for different security functions, such as IDS, physical layer security, compute layer security, etc.
- Receives status and alerts from nodes in real-time
- Controls and updates security policy based upon network status

In order to minimize traffic and compute requirements at the Management Layer, this layer is primarily responsible for AI/ML and network security orchestration. The Management Layer uses the Distribution Layer to disseminate and receive messages from the nodes. From the component perspective, this layer can be seen as implementing the Management Component and Security Component, although not in its entirety.

As its name implies, the **Distribution Layer** is responsible for the distribution of messages and data between different network nodes. This is especially important for FL, since training data may reside at different nodes in the network and may need to be transferred as such. Hence, the Distribution Layer can be seen as the freeway that connects the different elements and manages traffic. This layer also implements the security policy as defined by the Management Layer. The Distribution Layer implements functionality from the Distribution Component and the Security Component.

From Fig. 5, this layer connects Compute Layer, which does all for the AI/ML processing on its or cluster of nodes. The figure shows only to Compute Layers, but this is not a limit.

Finally, the **Compute Layer** is responsible for the computational processing associated with implementing and training the models. It is envisioned that there are many AI/ML

models running on a particular node or set of nodes. These could be NN models, LSTM, unsupervised models, etc. This layer receives models and data from the Distribution Layer and provides its results to that layer, which are subsequently forwarded to the Management Layer.

With this higher layer architecture, the next step is to describe how to implement such an architecture across a distributed network. Not all functions are needed at both control and client nodes, consequently the next section describes partitioning of the functions across the network entities.

C. Component View of the Secure FL Architecture

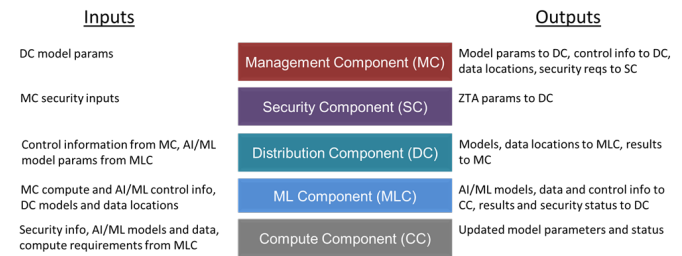


Fig. 6. Component view of the secure FL architecture.

As illustrated in Fig. 6, there are five different components: Management Component (MC), Security Component (SC), Distribution Component (DC), Machine Learning Component (MLC) and Compute Component. In this way, the distributed CC functionality can be partitioned by functionality. For example, the AI/ML algorithm implementation and training can be kept separate from the management and security of the network. Furthermore, these components do not necessarily need to be resident at the control and client nodes, which is illustrated in Fig. 7.

The **Management Component** component is one component of the Management Layer. It is responsible for operating the FL network by orchestrating the AI/ML process by collecting model coefficients and architectures, security parameters, etc. and using its distribution component to distribute these to network nodes. It is also responsible for orchestrating security policy. This layer manages the functions that are implemented by the SC and DC at the control node.

The **Security Component** takes security state information from the MC and creates the security policy and functions. It may use advanced methods such as ML models derived from data received from the network elements. As such, this is the powerful security engine that manages security across the network.

The **Distribution Component** has two incarnations. The first resides on the control node and manages the distribution of data across the network. The second sits on the client node and communicates model and training data with other nodes. While control information is passed from the control node, communication between nodes is direct to minimize network traffic.

The *Machine Learning Component* is responsible for the model implementation and all AI/ML related functions such as continuous monitoring, integration of multiple models, model training, etc. It interfaces with the network through the DC and processing is performed by the CC.

The *Compute Component* performs all computations needed for model processing and training. It runs the models from the MLC, but it may also gather parameters from the node’s operating system regarding potential attacks. Thus, in addition to running models, the CC is responsible for collecting security, or even sensor data, and passing it up to the MLC.

A brief summary of each layer is provided in Table I.

TABLE I
COMPONENT DESCRIPTIONS.

Component	Description
Management	AI/ML Network management and control, similar to ITU MLO
Security	Secures models and contains sandboxing for verification and validation to ensure integrity “accurate, tamper-free” models are integrated into the ML layer, verification of data sources/authentication, trust platform, as in multi-agent platforms
Distribution	Trusted network formation and model parameters, training data distribution
Machine Learning	Hosts AI/ML algorithms of different types (supervised, unsupervised, etc.), integrates these together, maintains specified interfaces and contains model implementations
Compute	Focuses on distributed computing and virtualization, Apache Spark is an option

D. Network Topology, Software Architecture and Use Cases

The aforementioned architecture described thus far is not limited to any particular network topology - centralized, decentralized, hierarchical multi-layer, etc. However, as applied to a 5G/B5G type of network, a centralized topology is in line with those found in practice, or carrier networks. It is also in line with the framework set forth by ETSI for MLFO. Thus, that is the approach that we take in this paper, but obviously, a decentralized approach such as found in blockchain, with cryptocurrency being a prime example, is a very interesting and new area related to FL and it will be an area that we will focus on in the future.

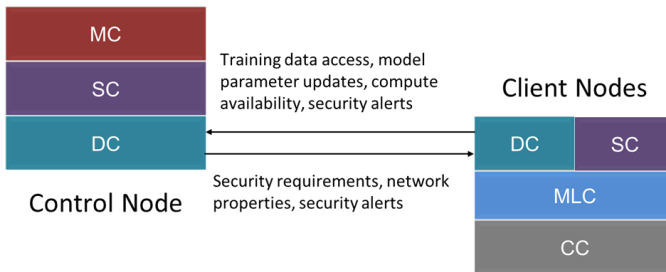


Fig. 7. Control and client nodes.

A centralized model is shown pictorially in Fig. 7. The figure is simple and meant to demonstrate the interaction between

the control node and client nodes and is a mixture of our layer and component perspectives. The duties of the control node are centered on management of network operations. This also includes security policy across the network. For example, the MC maintains the “objects” and databases that are needed for FL, security state, and others. In the case of FL, the MC know the locations of all training data, AI/ML models that are being used, and the security state at each node. Implementation of this policy resides with control node’s SC and distribution takes place via the control node’s DC. Essentially, this is a client-server architecture with the control node acting as a server, but the tasks are not trivial.

All intelligence in terms of network operation, security policy and control, and AI/ML orchestration reside a the client node. Performing these functions for a large network, as is the case in 5G/B5G, is no easy task. From a FL perspective, this is a very interesting problem. Consider the fact that there are many nodes with different sources of data that can be used to secure the network, plus the fact that this data is not sparse. TCP/IP data streams are dense in terms of computing requirements, but these are not the only sources of data. As illustrated in Fig. 8, real-time forensic data, e.g. processing load, file access attempts, and others, paint a picture of the security state of the network.

The client node processing is partitioned into three areas: distribution and security, AI/ML model implementation and training, and compute processing. Distribution and security are linked together due to the fact that secure links are required to share the data. For example, SSH can be used between nodes in the network. Additionally, the SC monitors the security state of the node by receiving security policy information from the control node. A second function of the SC is to collect security metrics, i.e. forensics, and send them to the control node. Essentially, it is monitoring the state of the node to ensure that there has been no intrusion, and hence, any training data or metric collected from the node are valid.

The focus of this paper has been on using AI/ML for network security. However, this is not a limiting case. Data collection and training of AI/ML models for computer vision, natural language processing (NLP) and future application areas still apply. Consequently, the MLC component may have multiple models to run, and send to the CC, but in a network environment, cybersecurity is still tantamount, even if it is simply executing models and generating forensic-based metrics. For the secure FL architecture that is proposed here, it must coexist with other AI/ML models.

The CC is an importance resource in the network architecture. Not only is in critical to host environment; it can also be used to train models as part of the FL architecture. For example, consider the case of edge computing in a corporate or 5G/B5G environment. Depending the computing resources available, training can be allocated to compute elements in much the same was as water-filling to maximize channel throughput subject to an energy constraint as described above.

E. Open Software Implementation

To implement the secure FL architecture, there are many tools that can be used, both proprietary and open source, though we propose an open architecture with open tools. The building blocks are shown in Fig. 8. At the top, there is OpenStack, which is a critical component in managing this cloud network and can be used to implement the ideas put forth in MLFO, and can be found in data centers and public/private clouds where there will be challenges to implement such an approach in a non-homogenous network that has many disparate components. Spring Security is an open source security platform that can be used to implement the essential security functions, such as SSH. To implement communication between nodes requires software that can implement distributed computing, such as Apache Spark. Communication between nodes can be done using a RESTful API which is in common use today.

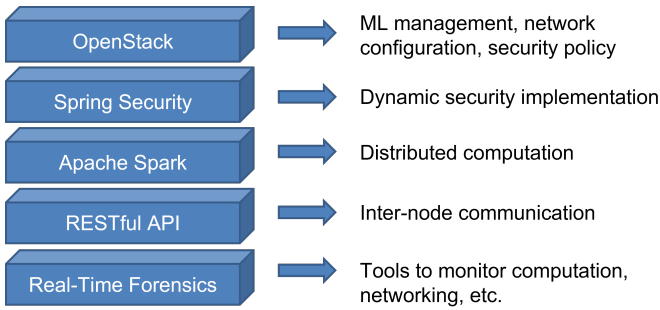


Fig. 8. Open source software implementation.

The aforementioned blocks are essentially the internal components that enable the secure FL architecture. Whether the network purpose is for security or non-security applications, a secure architecture must be integrated. To implement real-time proactive security requires the security state to be sent to the control node. Algorithms and metrics are needed that can accurately represent the security state of the network. We envision these capabilities as residing in the real-time forensics block.

IV. ZERO TRUST ARCHITECTURE VIA NETWORK COOPERATION

Our architectural approach can be applied to different network topologies from enterprise to the Internet, the key point being that cooperation can lead to a ZTA by implementing cooperative algorithms that can address all levels of the security stack. Because of the nature of cyber attacks, a distributed, and federated, system has the ability to detect cyber attacks more effectively. For example, if a node has been compromised, self-detection may not occur since a root-level attack may turn off all defenses. However, should the attacker attempt to communicate with other nodes, clearly, its presence will be detected.

In Table I, we describe how this architecture can be used to implement ZT using cooperation. The key takeaway from

this table is that for a secure network, the following items, at a minimum, are required:

- 1) *Network Registry* - For a secure FL system, a registry of all nodes and resources in the network should reside at the MC. The registry database contains not only the current state of the network, but also the historical transactions, such as file access attempts, etc. This information can be used to set the network security policy via AI/ML methods.
- 2) *Trusted Implementation* - While SSH and encrypted tunnels can secure communication, they do not guarantee trust. It must be assumed that insider attacks will occur, so methods are needed to detect this by monitoring security analytics, file operations, network communication, etc.
- 3) *Real-Time Security Analytics* - Knowledge of the current security state of network nodes is essential to securing the network. Computing metrics, file access attempts, state of the key management system are all important elements in being able to determine whether a node has been compromised.

V. FUTURE DEVELOPMENT; RISKS AND GAPS

The fundamental view of a secure FL architecture is that computing capability should be a core component of the architecture. Rather than simply saying that each node can compute its own model coefficients, for example, we see the need to distribute data across that network such that other nodes can contribute to the compute processing. In doing so, this will create the possibility of a richer set of algorithms, not just from a run-time perspective but also from a training perspective. One such example is with IoT devices that may collect data but not have sufficient compute capability to update model coefficients.

With the aforementioned architecture and future technology developments, AI/ML-based security will be a key component of future networks. By developing the aforementioned technical approaches and frameworks into a suite of 5G and Future Network products, the great potential of AI/ML for security can be realized.

Ultimately, the goal is to develop the dynamic, predictive AI/ML system such that it can support advanced capabilities to secure Future Networks beyond what is capable today. Some of the areas with security AI/ML will be beneficial includes:

- **Enhanced Threat Detection for Network Intrusion Detection and Prevention** – Unlike today's intrusion detection and prevention systems, future AI/ML-based systems will be able to learn and adapt in real-time. New models will be developed that can learn from larger sources of data. For example, higher order parameter vectors have the potential to more accurately detect and remedy threats compared today.
- **Threat Model Online Learning** – Applications such as self-driving cars, relied on large datasets with associated labels to train their supervised Deep Neural Networks

TABLE II
ZERO-TRUST ARCHITECTURE USING LEARNING ALGORITHMS.

Tenet	Architecture Support	Notes on Implementation
1. All data sources and computing services are considered resources.	The Management Layer monitors all resources in the network. Access to the network is controlled by the MC.	<ul style="list-style-type: none"> Registry of nodes and elements of the network at MC Knowledge of network node state is required Need for AI/ML algorithms for dynamic security policy
2. All communication is secured regardless of network location.	The SCs within the Control and Client nodes use encrypted channels such as SSH, tunneling protocols, etc.	<ul style="list-style-type: none"> Autonomous management of encryption keys and secure tunnels is required Knowledge of node security state is needed
3. Access to individual enterprise resources is granted on a per-session basis.	The MC knows all resources in the network and distributes access rules to the client nodes via the Distribution Layer.	<ul style="list-style-type: none"> Smart key management routines
4. Access to resources is determined by a dynamic security policy and may include other behavioral and environmental attributes.	The CC monitors computing performance, such as memory usage and system performance, and relays this to the Security Component at the Control Node for continuous monitoring. Similarly, the SC monitors resource access requests and reports these to the Control Node.	<ul style="list-style-type: none"> Policy is set forth at MC Security analytics are needed to derive policy AI/ML algorithms are needed to derive security state from analytics
5. The enterprise monitors and measures integrity and security posture of owned and associated assets.	This is similar to Tenet #4 with real-time integrity monitoring.	<ul style="list-style-type: none"> Security state and real-time analytics are inputs to this function
6. All resource authentication and authorization are dynamic and strictly enforced before access is allowed.	The Management Layer controls access to resources on a per-session basis.	<ul style="list-style-type: none"> Must detect insider attack where key is valid but an attack is underway. Trust through intelligent key management and state monitoring are required. Security analytics and network communication monitoring may be able to detect insider attack.
7. The enterprise collects information about the current state of assets, network infrastructure and communications.	The same approach with real-time monitoring and centralized control as in Tenets #3 and #5 is employed here.	<ul style="list-style-type: none"> This is the security analytics and monitoring that was described earlier.

(DNN). Due to rapidly changing cyber security threat profiles, techniques that train in real-time are needed. AI/ML techniques such as GAN and Reinforcement Learning (RL), among other techniques will play an important role in the AI/ML Security Ecosystem.

- **Smart Network Controllers** – Today, network interface controllers process Ethernet packets and pass the data to the network processor. Newer embedded technologies will allow cybersecurity functions to operate at the device level to mitigate threats before they enter the network. For example, SmartNIC devices with AI/ML algorithms can operate on embedded processors and mitigate attacks before they enter the network. New algorithms can be loaded in real-time as the threat profile changes.
- **DDOS, Jamming and Spoofing Mitigation** – While jamming and spoofing may be considered to be an RF phenomenon only, AI/ML models can be used to detect these threats as they continue to evolve. In doing so, it is possible to develop better situational awareness by recognizing where in the environment that the attacks are taking place.

REFERENCES

- [1] M. A. Enright and C. N. Kurby, "A signals of opportunity based cooperative navigation network," in *Proc. IEEE National Aerospace Electronics Conference (NAECON)*. Dayton, OH, July 2009, pp. 213–218.
- [2] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1949.
- [3] T. Cover and J. A. Thomas, *Elements of Information Theory*, 3rd ed. New York: Wiley, 1991.
- [4] J. K. et. al., "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2017.
- [5] S. A. et. al., "Federated learning for intrusion detection system: Concepts, challenges and future directions," *arXiv preprint arXiv:2106.09527*, 2021.
- [6] M. X. et. al., "Machine learning security: Threats, countermeasures, and evaluations," *IEEE Access*, vol. 8, pp. 74 720–74 742, 2020.
- [7] X. H. et. al., "BAYWATCH: Robust beaconing detection to identify infected hosts in large-scale enterprise networks," in *Proc. of the 46th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks*. Toulouse, France, July 2016, pp. 479–490.
- [8] A. O. et. al., "MADE: Security analytics for enterprise threat detection," in *Proc. of the 34th Annual Computer Security Applications Conference*, Dec 2018, pp. 124–136.
- [9] 3GPP TS 33.501, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security architecture and procedures for 5G system (Release 17)," vol. 17.2.1, Jun 2021.
- [10] A. Dutta and E. Hammad, "International Network Generations Roadmap (INGR), Virtual Workshop, Security Working Group," June 2020.
- [11] S. V. N. Parasram, *Digital Forensics with Kali Linux*, 2nd ed. Birmingham-Mumbai: Packt, 2020.

- [12] S. W. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," 2020.
- [13] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3gpp 5g networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 170–195, 2019.
- [14] Huawei, "AI Security White Paper," <https://www-file.huawei.com/-/media/corporate/pdf/cyber-security/ai-security-white-paper-en.pdf>, accessed: August 24, 2021.
- [15] J. S. I. Goodfellow and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [16] A. C. et. al., "Generative adversarial networks: An overview," *arXiv preprint arXiv:1710.07035*, 2014.
- [17] The Open Group, "FACE™ Technical Standard, Edition 3.1," July 2020.
- [18] International Telecommunication Union, ITU-T Y.3172, "Architectural framework for machine learning in future networks including IMT-2020," Jun 2019.
- [19] International Telecommunication Union, FG ML5G Technical Specification, "FG ML5G Technical Specification Requirements, architecture, and design for machine learning function orchestrator," July 2020.