Alka Kaushik et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 4, Issue 3, Sept 2017, pp. 313-319

Security and trust issues in Fog computing used in Medical Domain applications

Alka Kaushik

B. Tech, M. Tech MDU Rohtak
Email: cakapilraj@yahoo.com

Abstract- **Fog computing uses one or more collaborative end users or near-user edge devices to perform storage, communication, control, configuration, measurement and management functions. It can well solve latency and bandwidth limitation problems encountered by using cloud computing. First, this work discusses and analyzes the architectures of Fog computing, and indicates the related potential security and trust issues. Then, how such issues have been tackled in the existing literature is comprehensively reported. Finally, the open challenges, research trends and future topics of security and trust in Fog computing are discussed.**

Keywords— Fog computing, Cloud computing, Trust Security

I. INTRODUCTION

Cloud computing (the Cloud in brief) has drastically changed the landscape of information technology (IT) by providing some major benefits to IT users, including eliminating upfront IT investment, scalability, proportional costs, and so on [1–5]. However, as more and more devices are connected, latency-sensitive applications seriously face the problem of large latency. In addition, Cloud computing is unable to meet the requirements of mobility support and location awareness. To overcome these problems, a new paradigm called Fog computing (the Fog in brief) was proposed in 2012 [6]. According to Bonomi et al. [8], the Fog is a highly virtualized platform that provides storage, computing and networking services between the Cloud data centers and end devices. Both Cloud and Fog provide data, computation, storage and application services to end users [9]. However, the latter is distinguished from the former by its decentralization, processing large amounts ofdata locally, software installation on heterogeneous hardware [10], proximity to end-users, dense geographical distribution, and support for mobility [11]. Here, we show an example of a traffic light system to discuss the relationship between them when dealing with latency. In a traffic light system without the Fog, there may be 3~4 hops from the monitoring probe to the server in the Cloud. Hence, realtime decisions cannot be made immediately and the system faces the challenge of network latency. However, by using the Fog, the monitoring probe acts as a sensor, and the traffic lights act as an actuator. The Fog node can send conventional compressed video that may endure some time latency to the Cloud. When the Fog node detects an ambulance's headlight flashing, it makes an immediate decision to turn on the corresponding traffic lights, so as to let the ambulance go through without any delay. However, the Fog cannot replace the Cloud but supplements it. Many companies and institutes, such as ARM, Cisco, Dell, Intel, Microsoft Corp., Cloudlet, Intelligent Edge by Intel and the Princeton University Edge Laboratory are devoted to research and development of the Fog. OpenFog (Found in 2015) Consortium workgroups are working towards creating an open architecture for the Fog to enable its interoperability and scalability [12]. Network equipment like switches and gateways is provided by Cisco, Huawei, Ericsson, etc. The current research trends reflect the tremendous potential of the Fog.

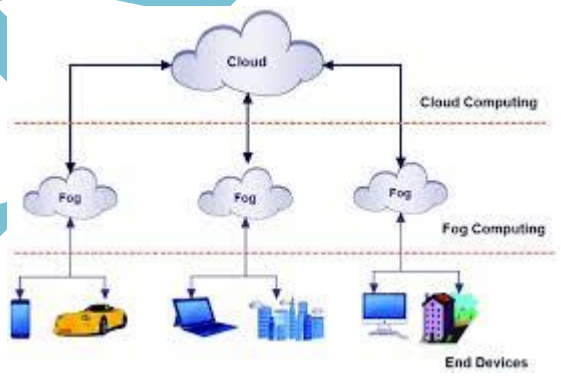

Fig. 1

The Fog features with location awareness, low latency and edge location [13]. It fits to a scenario where a huge number of heterogeneous ubiquitous and decentralized devices communicate, need to cooperate, and perform storage and processing tasks [6]. Users can visit their Fog anytime by using any device that can be connected to the Fog network. The Fog has many applications in such areas as smart city and healthcare. It can also provide better Quality of Service (QoS) in terms of fast response and small energy consumption. The Fog uses network devices (named Fog nodes in this paper) for latency-aware processing of data collected from Internet of Thing (IoT). Fog nodes are denoted as heterogeneous components deployed in an edge network in Fog environments. They include gateways, routers, switchers, access points, base stations, and specific Fog servers. The Fog facilitates uniform and seamless resource management including computation, networking and storage allocation. Fog nodes are often the first set of processors that data encounter in IoT, and have the resources to implement a full hardware root of trust. This root of trust can be extended to all the processes and applications running on them, and then to

the Cloud. Without a hardware root of trust, various attack scenarios can compromise the software infrastructures of the Fog, allowing hackers to gain a foothold. The requirements of life safetycritical systems mandate the sorts of security capabilities available on the Fog. Hence, new security and trust challenges emerge with the rise of the Fog. The existing methods cannot be directly applied to the Fog because of its mobility, heterogeneity, and large-scale geo-distribution [12].

Fig. 2

This work reviews these concerns in the Fog and the existing solutions. Differing from other survey papers about Fog computing, this paper focuses on its security and trust issues, especially in the region of the Fog. The rest of this paper is organized as follows. Section 2 reveals a Fog architecture as well as related security and trust issues. Section 3 summarizes the related work to cope with security and trust issues. Section 4 presents open research problems. Section 5 discusses the future work. Finally, Section 6 concludes this survey paper.

II. Fog Computing: The State of the Art

To solve the facing challenges of the cloud computing, Cisco proposed the fog computing to expand the centralized cloud computing. Due to the advantages of fog computing, the researchers have done some researches, and the fog computing has become a hot research direction for the radio access network, wireless access network, vehicular network and internet of things. In [7], in order to handle these challenges caused by fog computing, the authors presented the three-layer hierarchical game framework to manage network resources. To solve the security problem of fog computing, the research [8] proposes an architecture framework to guarantees that the user information will not be leaked when the channel is attacked. The research [9] gives a general answer to the ten hot issues of fog computing, such as what is fog computing, what is the relationship between fog computing and cloud computing, what are the scenarios for fog computing, and so on. In order to reduce the network latency, the research [11] uses the mobile edge network to deploy some VNFs of the service function chain. In [12], the authors studied the fusion of NFV, 5G and fog computing, and proposed a MANO-based architecture to achieve a unified management of internet of things. The research [7] discusses the influence of fog computing on 5G radio access network, and proposes a 5G radio access network based on fog computing. In order to improve quality of experience, the research [8] proposes internet access networks architecture based on fog computing to deploy virtual machines into the user's neighborhood. To deal with the challenges of user growth, in [29], the authors have proposed a radio access networks architecture to provide services, which is based on fog computing and SDN. To improve the efficiency of face recognition and reduce network transmission, the authors present a face recognition system based on fog computing in Internet of Things [10]. In [13], the authors studied the utilizing of fog computing and SDN to provide services in vehicular networks, in order to overcome the instability of fog communication, a method is proposed to reduce theoverhead of control information by using network information. In order to accommodate the increase of vehicle traffic and reduce the delay, the research [11] proposed a vehicular network architecture to achieve mobile computing. These researches [9-10] on fog computing do not take into account the VNF deployment or migration scenarios, hence they cannot be applied to the VNF deployment or migration scenarios. Although [11, 12] combine with fog computing and NFV to conduct research, but they did not study the problem of the VNF/SFC migration. .

III. Fog Computing in Healthcare IoT Systems

In this section, articles that use fog computing are presented and discussed in order to demonstrate the importance of employing fog computing in healthcare IoT Systems. Monitoring is considered as one of the important methods in IoT healthcare Systems. A fog-based monitoring system was presented [10], which provides remote monitoring with low cost. Moreover, in this case, the system is comprised of smart gateways and efficient IoT sensors. Furthermore, ECG signals, body temperature, and respiration rate are collected by sensors and sent wirelessly to gateways in order to produce notifications following an automatic analysis. Considering privacy and security as important aspects of healthcare applications, a fog-based healthcare framework was proposed [11], which implemented fog between the cloud and end devices as an intermediate layer. Privacy and security were enhanced at the edge of the network by using a cloud access security broker (CASB).

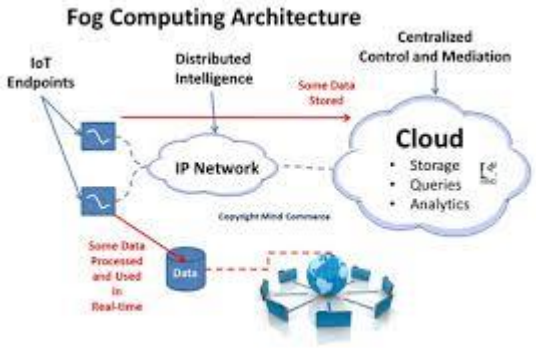

Fig. 3

The framework was applied by applying a modular approach. Data aggregating from multiple sources could be supported by the framework and adequate cryptographic assessment. Latency-sensitive healthcare data could affect the performance of healthcare applications. A fog-based computation platform was discussed in order to deal with latencysensitive healthcare data [12]. The large-scale geographically distributed healthcare application was managed by using a programming model. In this application, data consistency and data accuracy can be retained and service

delivery time can be improved. A fog computing system architecture was proposed in order to validate and evaluate sensed raw health data. In this process, embedded computing instances were constrained by resources [13]. The identification of the important patterns was performed through instances and then these were forwarded to the cloud. The primary objective of this system is to process huge data using reduced power fog resources. A smart e-health gateway was implemented [14] in fog computing as a means to support healthcare services in IoT and to offer data processing, data analysis, and realtime local storage. The smart e-health gateways were distributed and positioned geographically. The responsibility of each gateway is to carry out the task of managing a set of IoT devices that are directly connected to the patient. The system has the ability to monitor patients independently irrespective of their movement. In the fog-based system, energy-, mobility-, and reliability-related issues can be resolved effectively. The diagnosis of the patients who were infected with Chikungunya virus (CHV) was proposed in the fog-based healthcare system [15]. The system constitutes of three main layers: wearable IoT sensors, fog, and cloud layer. The system is used for identifying and controlling CHV virus. The diagnosis of the infected patients was carried out using Fuzzy-C means (FCM) and emerging alerts. Time-sensitive healthcare application data was proposed [16] for brain strokes and heart attacks, wherein fog computing was used to notify the users as early as possible. In these applications, fog computing enhances the execution time, network usage, and energy consumption cost. A fog computing distributed computational approach was proposed [17] for chronic obstructive pulmonary disease (COPD) and people suffering from mild dementia for Romanian healthcare regulations. eWall as a monitoring system is used to meet the requirements of the procedure. Fog computing reduces the communication overload and maintains patient privacy. Fog computing is implemented in the proximity of end-user devices/users as well as for large scale geographically distributed devices, communication in real time, mobility support, interoperability, heterogeneity and preprocessing with respected interplay connection with cloud. Fog computing has the ability to handle a variety of devices and sensors in addition to providing local processing and storage [28]. All the mentioned features of fog computing ensure that fog computing is the most suitable technique for Healthcare IoT systems which require the specified features. Fog computing differs from the traditional solutions to Healthcare IoT systems; fog-assisted system architecture has the ability to withstand the issues in numerous healthcare systems like scalability, energy awareness, mobility, and reliability, as shown in the architecture layer of fog computing in Fig. 2. [10]..

IV. A COMPARISON OF FOG COMPUTING AND RELATED COMPUTING PARADIGMS

This section focuses on the comparison of fog computing and related computing paradigms to demonstrate the value of fog computing in a variety of use cases. Moreover, this section provides a better understanding of how these computing paradigms can benefit the current and future landscape of connected devices. We compare fog computing with cloud computing as well as other related computing paradigms and summarize this section.

Cloud computing

Cloud computing has been instrumental in expanding the reach and capabilities of computing, storage, and networking infrastructure to the applications. The National Institute of Standards and Technology. NIST) defines cloud computing as a model that promotes ubiquitous, on-demand network access to shared computing resources [16]. Cloud data centers are large pools of highly accessible virtualized resources that can be dynamically reconfigured for a scalable workload; this reconfigurability is beneficial for clouds services that are offered with a pay-as-you-go cost model [17]. The pay-as-you-go cost model allows users to conveniently access remote computing resources and data management services, while only being charged for the amount of resources they use. Cloud providers, such as Google, IBM, Microsoft, and Amazon provide and provision large data centers to host these cloud-based resources.

Cloud services

Cloud offers infrastructure, platform, and software as services (IaaS, PaaS, SaaS). Application developers can use a variety of these services depending on the needs of the applications they develop. Infrastructure as a service (IaaS) allows cloud consumers to directly access IT infrastructures for processing, storage, and networking resources [18]. Suppose Sam wants to set up a high-tech agricultural system that utilizes IoT devices to monitor the condition of crops. Sam contacts a cloud provider and acquires an IaaS for development of his system. Sam now can configure the IaaS (often offered as a standalone VM) in terms of hardware and corresponding software for his need. Control over infrastructure (IaaS) allows Sam to customize hardware configuration, such as the number of CPU cores and RAM capacity, in addition to systems-level software. Sam can obtain an IaaS from Amazon Web Services (AWS), Microsoft Azure, or Google Compute Engine (GCE). On the other hand, platform as a service (PaaS) allows cloud consumers to develop software and fully supports software lifecycle – often with the help of a middleware – for software management and configuration. If Sam does not need to configure the infrastructure of the cloud, managing and configuration of hardware and software may detract from the productivity of Sam's business. Now, Sam could consider using PaaS offered by Apache Stratos, Azure App Services, or Google App Engine for his business. PaaS manages the underlying low-level processes and allows Sam to focus on managing software for his IoT-specific interactions. Moreover, PaaS providers often include tools for convenient management of databases and scaling applications. Now suppose Sam is willing to spend more money and likes to get full software packages, and he does not want to take care of software issues, such as database scalability, socket management, etc. Software as a service (SaaS) provides Sam an environment to centrally host his applications and removes the need for him to install software manually. Sam's client software now can be hosted on Google Apps or as a Web application. As demonstrated by these examples, cloud services can be utilized for distinct use cases for a variety of end users. Fig. 2 illustrates the re Fig. 2. Common cloud service models and their classifications

relative what portion of the application stack is managed by cloud providers. relationship among IaaS, PaaS, and SaaS with the underlying cloud infrastructure, and illustrates what portion of the application stack is managed by cloud providers.

Cloud resource provisioning

Since the demand for cloud resources is not fixed and can change over time, setting a fixed amount of resources results in either overprovisioning or under-provisioning, as depicted in Fig. 3. A foundation of cloud computing is based on provisioning only the required resources for the demand. This includes the use of virtualization for on-demand application deployment, and the use of resource provisioning to manage hardware and software in cloud data centers. Provisioning resources is an important topic in cloud computing that is widely explored. Since it is difficult to estimate service usage from tenants, most cloud providers have a pay-as-you-go payment scheme. As a result, providers can be more flexible on how to provision resources, and clients only pay for the amount of resources they actually use.

Types of cloud

There are four types of cloud deployments: private cloud, community cloud, public cloud, and hybrid cloud [16]. Private clouds are designed for use by a singular entity and ensure high privacy and configurability. Private clouds are a good choice for organizations that require an infrastructure for their applications. This type of deployment is similar to traditional company-owned server farms and often do not benefit from a pay-as-you-go cost model. Community clouds are used by a community of users, and the infrastructure is shared between several organizations. A community cloud results in decentralized ownership of the cloud by multiple organizations within the community without relying on a large cloud vendor for the IT infrastructure. Public clouds are the typical model of cloud computing, where the cloud services are offered by cloud service providers, such as Amazon, IBM, Google, Microsoft, etc. Public clouds are generally more popular, easy-to-maintain, and cost-effective compared to private clouds. In contrast to private clouds, public clouds may benefit from the pay-as-you-go pricing model. However, public clouds do not always offer users complete customization of hardware, middleware, network, and security settings. Hybrid clouds are simply a combination of the cloud types mentioned above. Hybrid clouds allow users to have finer control over virtualized infrastructure, and combining the capabilities from different types of cloud deployments is accomplished through standardized or proprietary technology [9]. The cloud computing paradigm was initially established to allow users to access a pool of computing resources for ubiquitous computing. Even though cloud computing has helped bring forth accessible computing, the time required to access cloud-based applications may be too high and may not be practical for some mission-critical applications, or applications with ultra-low latency requirements. Also, the rapid growth in the amount of data generated at the network edge by an increasing number of connected devices requires cloud resources to be closer to where the data is generated. Greater demand for high-bandwidth, geographically-dispersed, low-latency, and privacy-sensitive data processing has emerged – a quintessential need for computing paradigms that take place closer to connected devices and that support low-latency, high-bandwidth, decentralized applications. To address these needs, fog computing has been proposed by both industry and academia [6,7]. In order to provide a detailed comparison among fog computing related paradigms, we introduce various computing paradigms, starting with fog computing.

Fog computing

Fog computing bridges the gap between the cloud and end devices (e.g., IoT nodes) by enabling computing, storage, networking, and data management on network nodes within the close vicinity of IoT devices. Consequentially, computation, storage, networking, decision making, and data management not only occur in the cloud, but also occur along the IoT-to-Cloud path as data traverses to the cloud (preferably close to the IoT devices). For instance, compressing the GPS data can happen at the edge before transmission to the cloud in Intelligent Transportation Systems (ITS) [10]. Fog computing is defined by the OpenFog Consortium [6] as "a horizontal system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum." The "horizontal" platform in fog computing allows computing functions to be distributed between different platforms and industries, whereas a vertical platform promotes siloed applications [11]. A vertical platform may provide strong support for a single type of application (silo), but it does not account for platform-to-platform interaction in other vertically focused platforms. In addition to facilitating a horizontal architecture, fog computing provides a flexible platform to meet the data-driven needs of operators and users. Fog computing is intended to provide strong support for the Internet of Things.

Fog vs. cloud

A common example that is often used to distinguished fog and cloud computing is whether latency-sensitive applications can be supported while maintaining satisfactory quality of service (QoS). Fog nodes can be placed close to IoT source nodes, allowing latency to be noticeably reduced compared to traditional cloud computing. While this example gives an intuitive motivation for fog, latency-sensitive applications are only one of the many applications that warrant the need for fog computing. Nodes in fog computing are generally deployed in less centralized locations compared to centralized cloud data centers. Fog nodes are wide-spread and geographically available in large numbers. In fog computing, security must be provided at the edge or in the dedicated locations of fog nodes, as opposed to the centrally-developed security mechanisms in dedicated buildings for cloud data centers. The decentralized nature of fog computing allows devices to either serve as fog computing nodes themselves (e.g. a car acts as a fog node for onboard sensors) or use fog resources as the clients of the fog. The majority of differences between cloud and fog computing are attributed to the scale of hardware components associated with these computing paradigms. Cloud computing provides high availability of computing resources at relatively high power consumption, whereas fog computing provides moderate availability of computing resources at lower power consumption [13]. Cloud computing typically utilizes large data centers, whereas fog

computing utilizes small servers, routers, switches, gateways, set-top boxes, or access points. Since hardware for fog computing occupies much less space than that of cloud computing, hardware can be located closer to users. Fog computing can be accessed through connected devices from the edge of the network to the network core, whereas cloud computing must be accessed through the network core. Moreover, continuous Internet connectivity is not essential for the fog-based services to work. That is, the services can work independently with low or no Internet connectivity and send necessary updates to the cloud whenever the connection is available. Cloud computing, on the other hand, requires devices to be connected when the cloud service is in progress. Fog helps devices measure, monitor, process, analyze, and react, and distributes computation, communication, storage, control, and decision making closer to IoT devices [6] (refer to Fig. 5). Many industries could use fog to their benefit: energy, manufacturing, transportation, healthcare, smart cities, to mention a few.

Fog–cloud federation

There are clear differences and trade-offs between cloud and fog computing, and one might ask which one to choose. However, fog and cloud complement each other; one cannot replace the need of the other. By coupling cloud and fog computing, the services that connected devices use can be optimized even further. Federation between fog and cloud allows enhanced capabilities for data aggregation, processing, and storage. For instance, in a stream processing application, the fog could filter, preprocess, and aggregate traffic streams from source devices, while queries with heavy analytical processing, or archival results could be sent to the cloud. An orchestrator could handle the cooperation between cloud and fog. Specifically, a fog orchestrator could provide an interoperable resource pool, deploy and schedule resources to application workflows, and control QoS [24]. Through the use of SDN, fog service providers will have greater control over how the network is configured with a large number of fog nodes that transfer data between the cloud and IoT devices. 3.2.3. Fog RAN Fog computing can be integrated into mobile technologies in the form of radio access networks (RAN), to form what is referred to as fog RAN (F-RAN). Computing resources on F-RANs may be used for caching at the edge of the network, which enables faster retrieval of content and a lower burden on the front-haul. F-RAN can be implemented through 5G related mobile technologies. On the other hand, cloud RAN (C-RAN) provides centralized control over F-RAN nodes. C-RAN takes advantage of virtualization, and decouples the base stations within a cell of the mobile network from its baseband functions by virtualizing those functions [26]. In C-RAN a large number of low-cost Remote Radio Heads (RRHs) are randomly deployed and connected to the Base Band Unit (BBU) pool through the front-haul links. Both F-RAN and C-RAN are suited for mobile networks with base stations and are candidates for 5G deployments. Also, the use of F-RAN and C-RAN brings a more energy efficient form of network operation. We encourage the motivated reader to refer to reference [27] for more information about F-RAN. Fig. 4 shows a classification of computing paradigms related to fog computing and their overlap in terms of their scope. The figure illustrates our comparison of fog computing and its related computing paradigms. Table 1 lists the acronyms used for this figure and in the paper. We discuss the related computing paradigms in the order of their trend and show how some paradigms resulted in the emergence of others.

V. System Framework and Workflow

For the system, power signals will be collected for prognosis and optimization of machining processes. According to the research of Liu et al. [7] and Sealy et al. [8], power signals from CNC machines can indicate the working and tooling conditions of machines. Power sensors are easy to deploy and the data sampling rate is much lower than that of vibration or acoustic sensors, which has been verified by experiments [5]. Meanwhile, energy consumption for machining processes can be also calculated based on power signals in order to achieve sustainable optimization of the machining system. Therefore, for this research, power signals, which are analyzed in the time domain, are used for both abnormal condition detection and multi-objective (including energyefficient) re-scheduling optimization for detected faults. As aforementioned analysis, the system has been designed based on the fog computing paradigm to provide a smart solution for dynamic prognosis and optimization of machining processes. The system is comprised of three layers to optimize the computation efficiency and latency of data transmission.

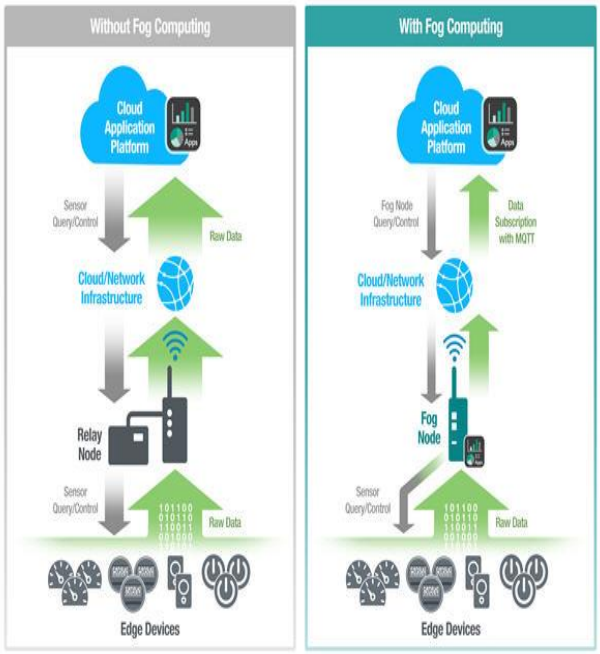

Fig. 4

Data transfer in the system between layers are through the MQTT protocol, taking the advantage of its lightweight communication [9], and information exchange in dual-direction simultaneously [30]. The system structure is illustrated in Fig. 1. The three layers and a coordinator between layers are briefly introduced below. Details will be elaborated in the following Section 4. (1) Terminal layer: This layer is integrated with physical machine equipment via sensor devices, circuits and routers. Machining processes will follow an optimized schedule sent from the cloud layer. During entire machining processes, power signals of machines are

continuously collected via power sensors and transmitted to a fog layer for further processing. When abnormal conditions of machines and tooling (e.g., severe tool wear, tool breakage, spindle failure) are detected on the fog layer, a re-schedule optimization will be triggered on the cloud layer and the generated optimized schedule will be sent to this physical layer for dynamic production adjustment. (2) Fog layer: A trained CNN is deployed on the fog layer for efficient detection of abnormal conditions. To facilitate the efficiency and effectiveness of computing in the CNN, power signals during machining processes are first partitioned into individual stages of the machining process for each component.

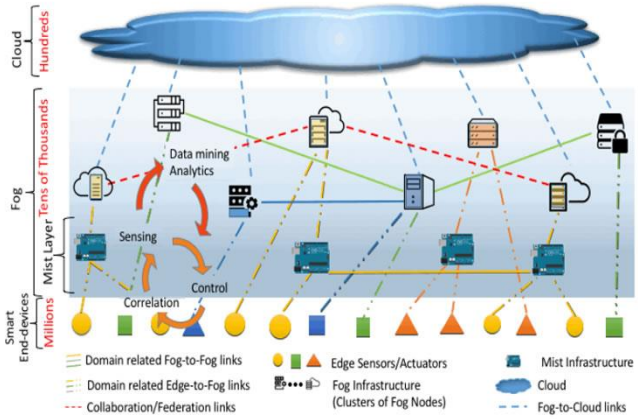

Fig. 5

An algorithm of Gaussian kernel smoothness is embedded for de-noising the signals to facilitate detection. The CNN will be trained on the cloud layer and the training process will be updated for new patterns of signals when new types of components are arranged for machining in production lines. Based on the design, abnormal situations are detected using the trained CNN on fog without relatively long transmission time of the power signals to the cloud layer for judgement and decision making. Thus, signal data pertaining to operations will be kept within companies. Machines can be stopped quickly for maintenance and adjustment when potential abnormal conditions occur. (3) Cloud layer: The cloud layer has databases to store reference signals for each machined components (called reference signals in the paper) and typical abnormal conditions for such components. The reference signals for machined components are built based on historical data. The CNN is re-trained when new components are arranged and reference signals are received. The trained CNN is transmitted back to the fog layer for deployment after the update.

VI. Conclusion

On the cloud layer, there is a multi-objective optimization algorithm to be triggered for re-scheduling when necessary. (4) System coordinator: It is arranged between the fog and cloud layers to coordinate tasks as follows: (i) updating the knowledge database on the cloud layer for reference signals of newly machined components; (ii) re-training of the CNN using the new references; (iii) updating the trained CNN on the fog layer; (iv) triggering the scheduling optimization for the situation of abnormal situations during machining processes, and sending the optimized schedule to the terminal layer for production adjustment

REFERENCES

[1]. Gang Sun, Dan Liao, Dongcheng Zhao, Zhili Sun, Victor Chang, Towards provisioning hybrid virtual networks in federated cloud data centers, Future Gener. Comput. Syst. 87 (2008) 457–469.

[2]. Mosharaf Chowdhury, Muntasir Raihan Rahman, Raouf Boutaba, ViNEYard:Virtual network embedding algorithms with coordinated node and link mapping, IEEE/ACM Trans. Netw. 1 (20) (2012) 206–219.

[3]. Gang Sun, Victor Chang, Guanghua Yang, Dan Liao, The cost-efficient deployment of replica servers in virtual content distribution networks for data fusion, Inform. Sci. 432 (2010) 495–515.

[4]. Bing Leng, Liusheng Huang, Chunming Qiao, Hongli Xu, A light-weight approach to obtaining NF state information in SDN+NFV networks, in: IEEE INFOCOM, 2011, pp. 1–9.

[5]. Marouen Mechtri, Chaima Ghribi, Oussama Soualah, Djamal Zeghlache, NFV orchestration framework addressing SFC challenges, IEEE Commun. Mag. 55 (6) (2017) 16–23.

[6]. Gang Sun, Yayu Li, Hongfang Yu, Athanasios V. Vasilakos, Xiaojiang Du, Mohsen Guizani, Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks, Future Gener. Comput. Syst. 91 (2015) 347–360.

[7]. Gang Sun, Yayu Li, Dan Liao, Victor Chang, Service function chain orchestration across multiple domains: A full mesh aggregation approach, IEEE Trans. Netw. Serv. Manag. 15 (3) (2014) 1175–1191.

[8]. Linqi Guo, John Pang, Anwar Walid, Joint placement and routing of network function chains in data centers, in: IEEE INFOCOM, 2018, pp. 1–9.

[9]. Wenrui Ma, Oscar Sandoval, Jonathan Beltran, Deng Pan, Niki Pissinou, Traffic Aware Placement of Interdependent NFV Middleboxes, in: IEEE INFOCOM, 2017, pp. 1–9.

[10]. Satyam Agarwal, Francesco Malandrino, Carla-Fabiana Chiasserini, Swades De, Joint placement and routing of network function chains in data centers, in: IEEE INFOCOM, 2015, pp. 1–9.

[11]. Gang Sun, Yayu Li, Yao Li, Dan Liao, Victor Chang, Low-latency orchestration for workflow-oriented service function chain in edge computing, Future Gener. Comput. Syst. 85 (2016) 116–128.

[12]. Tung-Wei Kuo, Bang-Heng Liou, Kate Ching-Ju Lin, Ming-Jer Tsai, Deploying chains of virtual network functions: On the relation between link and server usage, in: IEEE INFOCOM, 2016, pp. 1–9.

[13]. Zilong Ye, Xiaojun Cao, Jianping Wang, Hongfang Yu, Chunming Qiao, Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization, IEEE Netw. 30 (3) (2016) 81–87.

[14]. Salvatore D'Oro, Laura Galluccio, Sergio Palazzo, Giovanni Schembra, Exploiting congestion games to achieve distributed service chaining in NFV networks, IEEE J. Sel. Areas Commun. 35 (2) (2017) 407–420.

[15]. Keke Gai, Meikang Qiu, Hui Zhao, Lixin Tao, Ziliang Zong, Dynamic energyaware cloudlet-based mobile cloud computing model for green computing, J. Netw. Comput. Appl. 59 (2016) 46–54.

