

三、研究計畫內容（以中文或英文撰寫）：

（一）研究計畫之背景。請詳述本研究計畫所要探討或解決的問題、研究原創性、重要性、預期影響性及國內外有關本計畫之研究情況、重要參考文獻之評述等。如為連續性計畫應說明上年度研究進度。

1. Project Background, Motivation, and Goals

A major challenge in the wide-spread adoption of the Internet-of-Things (IoT) architecture lies in the high probability of system security being *weak*. This security challenge is mainly due to the innate *distributed* architecture of IoT in which weak links can crop up in the form of “insecure things” connected to the IoT system or weak *Fog* computing platforms that use only *default* data encryption mechanisms are still susceptible to different malware/intrusion attacks due to weak APIs, system/application vulnerabilities, and shared technologies. A typical example of an “insecure thing” is a connected *smoke detector* that can be easily hacked since it normally does not have any sort of security design. A typical example of a weak *Fog computing platform* is the presence of insecure APIs. Conventional security issues become even more serious in an IoT system. In our target architecture, the introduction of a fog computing platform between the sensor devices and the cloud servers results in a new layer of vulnerabilities because fog computing platforms have limited resources which were designed to help in data processing or localized computation and thus its usage for security would itself be a deviation from its original purpose. As mentioned in a survey on Fog security [1], there are 12 different kinds of security issues in Fog-based IoT platforms, which will be described later in this proposal; however, in this project we will focus on resolving two issues that are currently of much higher significance due to their prevalence in many applications. **In this project, we propose to resolve the two important security issues of IoT, namely data integrity and anomaly detection, through an integration of Blockchain and deep neural networks, respectively, into a Fog-based IoT system.**

The reasons for selecting these two target security issues are as follows. First of all, since IoT is a distributed architecture, data integrity becomes of paramount importance as large amounts of data are being generated continuously by geographically distributed sensors, being fused together (features selected or engineered), and automatically processed in various ways, thus the integrity of data is of major concern, especially if the data affects the final outcomes of applications. Over time, the data generated and collected in an IoT becomes big data with the 3 Vs, namely volume, variety, and velocity. Thus, any solution for protecting data integrity must also cope with the 3 Vs of big data.

Second, since it is very difficult to completely protect a distributed architecture such as IoT with thousands or even millions of connected things, such as sensors, for cyber-physical systems (landslide detection, smart grids, smart traffic, etc.), it becomes invariably a necessity that the IoT system is “*designed with anomaly detection*” so that anything abnormal is detected before it causes irrecoverable damage to the system.

Further, the above two target issues, data integrity and anomaly detection, are also co-related in the fact that the solutions that resolve the two issues can cooperate with each other so that data integrity protection and anomaly detection methods can go hand-in-hand to enhance the security of IoT systems. By cooperation, we mean that anomaly detections can be made a priori (aka predictions) such that we can try to prevent data integrity attacks or enhance the level of data integrity protection based on the

predicted severity of anomalies that are detected or predicted.

There are two main innovations in this project, including the customization and integration of Blockchain security into IoT and the adaptation of deep neural networks (DNN) in IoT. Currently, both of these techniques are very popular and everyone is talking about them. However, the propaganda for these two techniques are much more wide-spread than there are research work that can actually solve the problems related to the design of these two techniques into an IoT system. As described in [2], 80.5% of research papers were conducted on Bitcoin and only 19.5% focused on other applications. Further, the main problems of Blockchain, namely throughput, latency, and scalability, were not addressed in any of the surveyed 41 papers from 2013 to 2015. Only starting from last year (2016), there are now some research results on how to scale blockchains [3] [4], especially in the network plane and the consensus plane, out of the 5 planes in a blockchain security system design. Further, DNN has been proposed for anomaly detection [5], but it has been restricted to specific applications such as the CAN network in a vehicle. Thus, for both blockchains and DNN in IoT, there are still several open questions which will be answered by this project in the course of 3 years.

There are three main goals of this project including (a) A novel blockchain security platform architecture that is customized for fog-based IoT architecture, (b) A DNN-based design for anomaly detection in fog-based IoT systems, and (c) An integration of the blockchain and DNN-based anomaly detection. The three goals are as depicted in Fig. 1.

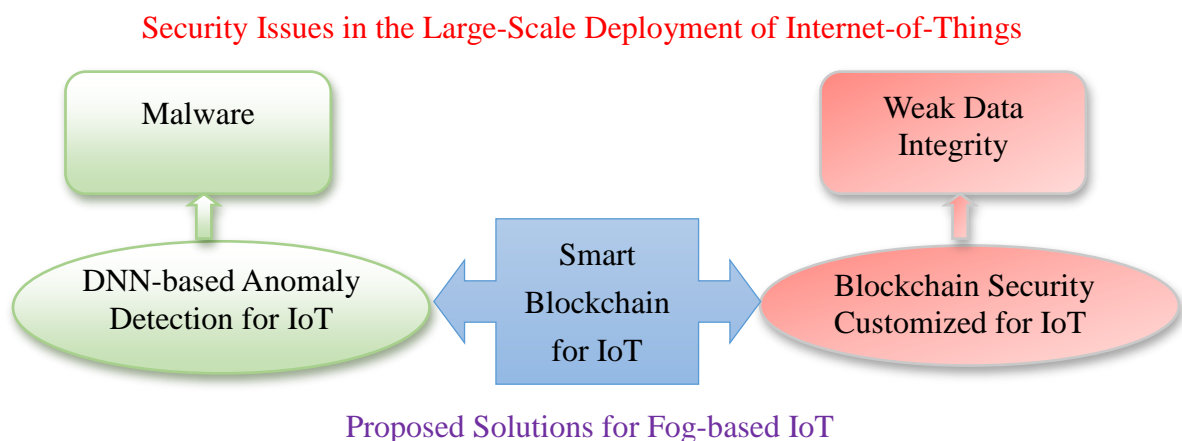


Figure 1. Target IoT Security Issues and Proposed Solutions

Before going into the target research problems, we will first discuss the 12 security issues found in IoT, as detailed in the following [1].

- (1) Advance Persistence Threats (APT) are cyber attacks on company infrastructures to steal data and intellectual property.
- (2) Access Control Issues (ACI) result in unauthorized users being able to obtain confidential data and install malwares.
- (3) Account Hijacking (HI) is the illegal use of a valid account, for example, phishing.
- (4) Denial of Service (DoS) overwhelms a system's finite resources such that it cannot function properly anymore.
- (5) Data Breaches (DB) result in confidential data illegally released or stolen.

- (6) Data Loss (DL) occurs when data is accidentally or maliciously deleted from the system.
- (7) Insecure APIs (IA) are sometimes provided by cloud or fog platforms.
- (8) System and Application Vulnerability (SAV) are exploitable bugs arising from software and configuration errors that an attacker can use to infiltrate and compromise a system.
- (9) Malicious Insider (MI) is a user who has authorized access to the network and system, but has intentionally decided to act maliciously.
- (10) Insufficient Due Diligence (IDD) arises when the adoption, design, and implementation of a system was too rushed and resulted in security flaws.
- (11) Abuse and Nefarious Use (ANU) occurs when free resources are utilized by malicious users for malicious activities.
- (12) Shared Technology Issues (STI) occur due to sharing infrastructures, platforms or applications that do not have enough security supporting features.

The above 12 security issues in IoT occur in different application areas as shown in Table 1 [1].

Table 1 Knowledge gaps for application area based analysing current Fog implementations against the twelve categories of security issues

Application area	APT	ACI	AH	DoS	DB	DL	IA	SAV	MI	IDD	ANU	STI
Virtualised radio access	✓	✓	✓		✓			✓	✓		✓	✓
Web optimization			✓			✓	✓					
5G mobile networks					✓	✓			✓		✓	✓
Smart Meters		✓				✓			✓			
Healthcare systems		✓	✓	✓	✓		✓	✓	✓			
Surveillance Video processing	✓				✓					✓		
Vehicular networks and Road safety				✓	✓							✓
Food traceability			✓			✓		✓				
Speech data						✓			✓			
Augmented Brain Computer	✓	✓			✓					✓		
Managing resources		✓	✓	✓					✓	✓	✓	✓
Energy reduction					✓	✓						
Disaster Response and Hostile environment		✓		✓						✓		

As fog computing security is concerned, there are mainly six categories as follows:

- (1) Visualization issues such as VM-based attacks, side channel attacks, and privilege escalation. Possible solutions include multi-factor authentication, intrusion detection, process isolation, and role-based access control. This category of attacks will have an adverse effect on all Fog services, data and users.
- (2) Web security issues such as SQL injection, cross-site scripting, session/account hijacking, malicious redirections, and drive-by attacks. Possible solutions include secure code, patch vulnerabilities, software updates, periodic auditing, firewall, anti-virus protection and intrusion prevention. This category of vulnerabilities will result in exposure of sensitive data and malicious installation of applications.
- (3) Internal/external communication issues such as man-in-the-middle attack, poor access control,

insecure APIs and services, and single-point of failure. Possible solutions include encryption, multi-factor authentication, isolation, certificate pinning, and transport layer security. This category of attacks result in unauthorized access to Fog resources.

- (4) Data security related issues such as data replication and sharing, illegal data access, low attack tolerance, malicious insiders, multi-tenancy issues, and DoS attacks. Possible solutions include policy enforcement, security inside design architecture, secure key management, obfuscation, data masking, and network monitoring. This category of attacks result in illegal file and database access, resulting in compromise of both user and Fog system's data.
- (5) Wireless security issues such as active impersonation, message replay attacks, data loss/breach, sniffing, and illegal resource consumption. Possible solutions include authentication, encryption, key management, secure routing, private network, and wireless security protocols.
- (6) Malware such as virus, Trojans, worms, ransomware, spyware, rootkits, and performance reduction. Possible solutions include anti-malware programs, intrusion detection, rigorous data backups, and system restore points. This category of attacks result in reduced performance of Fog platform due to malware infected nodes, allow back-doors to the system and corrupt/damage data.

Traditional solutions to the two selected security issues of data integrity and anomaly detection find it very difficult to cope with the distributed architecture of IoT, thus in this project we propose the use of blockchains and deep neural networks for resolving data integrity issue and anomaly detection, respectively. As described in the following, there are several **open research questions** that we will answer in the course of this 3-year project.

(1) Is blockchain really needed for IoT?

IoT is basically distributed so using a centralized security mechanism restricts the scalability of IoT. If an IoT system is to scale to a very large sized network of things (or millions of sensors such as those connected by *Low-Power Wide-Area Networks*, LPWAN), then a distributed security technology is required. Blockchain is currently a promising distributed security technology. Though IBM Blockchain and Watson IoT Platform seems to present an integrated solution for secure IoT design, there are still several open questions as detailed in the following.

(2) How to make Blockchains Feasible for Fog-based IoT?

On one hand, IoT has limited bandwidth and restricted fog computing resources and storage. On the other hand, blockchain security requires not only large network bandwidth but also powerful computing resources. On the face of it, IoT and blockchains do not seem to be compatible. However, as mentioned above, IoT needs blockchain. In fact, blockchain also needs IoT in the sense that the data sensed by IoT can be directly converted into transactions that can be committed to blockchains, thus making blockchains even more secure, that is, manual intervention and thus possible malicious alterations can be avoided. As described in the following, the open questions on making blockchains feasible for IoT are classified into 4 different planes, that is, layers in a blockchain architecture, including and network, consensus, storage, and view planes:

- Network Plane: Blockchains introduce more traffic into the already limited bandwidth of IoT device to edge and edge to backhaul network. *The open question is how to design and implement*

blockchains such that its use of the limited bandwidth in IoT can be minimized to that required for actual transactions only. A P2P solution is proposed by Khan et al [1], but there is no real design or validation of the P2P concept for IoT. Another solution in Block-based IoT architecture by Dorri et al [4] employ a 3-tier approach (local, shared, and overlay blockchains) and traffic is reduced by unicasting transactions in local and shared blockchains and by arbitrary selection of transactions in the overlay blockchain. Both of these solutions are not optimal solutions to the open question. We will try to resolve this question in this project and details will be provided in the next section.

- Consensus Plane: Bitcoin mining in the form of Proof-of-Work (PoW) is very compute intensive, which is not at all suitable in any network edge devices. *The open question is what kind of consensus algorithm is suitable for IoT, especially those with fog computing edges.* Proof of stake (PoS) is claimed to be thousands of times more cost effective than PoW; however, its convergence (resolution of consensus) is not guaranteed. Another proposal is the Byzantine Fault Tolerant (BFT) replication protocol with a small number of pre-designated trusted entities that greatly enhances scalability (throughput of 4.5K tx/sec with average transaction latency of 1.79 sec for 64 nodes processing batches of 8192 transactions, each of which is 190 bytes long). One more intuitive proposal is sharding protocols, where the task of consensus is split up among concurrently operating sets of nodes such that throughput is increased and per node processing and storage requirements reduced. Sidechains have also been proposed, which are delegations of trust across multiple tiers of consensus instances. In the block-based IoT architecture [4], Beta Reputation System is used for direct and indirect evidence of trust among distributed nodes. All of these are only proposals for reducing the computation and for increasing scalability; however, which one is suitable for IoT is still an open question.
- Storage Plane: The storage for the ledger can be a problem in IoT edge devices due to the limited storage space and bandwidth for communication (downloading ledger, etc.). *The open question is how to store the ledger in an IoT architecture such that its contents can be authenticated during read operations.* Sharding the storage of an *unspent transaction outputs* (UTXO) data structure is a possible solution.
- View Plane: A view is a data structure derived from the full ledger whose state is obtained by applying all transactions. New miners in bitcoin need 4 days to download the full ledger and reconstruct the UTXO set (a view). *The open question is how much of the full ledger need to be stored as a view by the edge devices in a fog-based IoT architecture.* Some proposed solutions include views via replication and outsourcing views via cryptography [3] with view authentication using succinct non-interactive arguments of knowledge (SNARKs). However, which solution is more suitable for IoT or if a new solution is required, this is still an open question.

(3) *How to use DNN for anomaly detection in IoT?*

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [6]. Anomalies in data translate to significant, and often critical, actionable information in a wide variety of application domains. For example, an anomalous traffic pattern in an IoT could mean a hacked “edge or thing” is sending out sensitive data to an unauthorized destination or is under device hijacking. Anomaly detection is important in IoT because IoT is

distributed, computation and energy resources are very limited, data changes continuously, and there are often lots of noise and missing data. *The open question is how anomaly detection should be built-in within a fog-based IoT system design.* Anomaly detection is not a new topic of research and there are numerous work on anomaly detection [6]. However, the application of DNN in IoT for anomaly detection is something new and there are only some work in this area [7]. The topics of research in area include the following.

- How to design a *lightweight* anomaly detection method using DNN in a fog-based IoT system architecture? A possible design solution is to share learning and inference across the sensors, the network edges (gateways), and the cloud. This will be explained in detail in the next section.
- How to make anomaly detection *more accurate* in fog-based IoT? A possible solution lies in the programmability and configurability of the DNN. This will be explained in the next section.
- How to leverage on the spatio-temporal features of IoT data in anomaly detection? A possible solution is to use a specific DNN for detecting collective anomalies. This will also be explained in the next section.

(4) *Why and how to combine DNN with Blockchain in IoT?*

The two technologies adapted in this project, namely DNN and blockchain security, have **not yet been integrated** in any state-of-the-art work currently. *The open question is how to integrate DNN with blockchain in an IoT architecture?* In this project, we will propose a novel *Smart Blockchain Security* (SBS) design for fog-based IoT architecture such that the predictions from DNN can be used to trigger transactions or smart contracts in blockchain. This method will be described in more details in the next section.

2. Related Work

Most related work have already been presented in the previous section, especially those related to blockchain and DNN. In the following, we will present some specific example systems or applications that are very much related to our research proposal.

(1) Blockchain Security for IoT

Several intrinsic features of IoT amplify its security and privacy challenges including: lack of central control, heterogeneity in device resources, multiple attack surfaces, context-aware and situational nature of risks, and scale [4]. Thus, several research initiatives have started on enhancing IoT security and privacy. Distributed capability-based access control method [8] safeguards sensitive information; however, introduces excessive delays and overheads and might comprise user privacy. IP-sec and TLS can be used for IoT authentication and privacy [9], but they are also too computationally intensive for IoT. By measuring the risk of disclosing data to others, a privacy management method was proposed for IoT [10]; however, the benefits of IoT services still outweigh the risks. The above-mentioned methods are not comprehensive enough to address the challenges of IoT security and privacy.

As a promising and comprehensive solution for IoT security, blockchains are being proposed [4]. Blockchains allow us to have a distributed peer-to-peer network where non-trusting members can interact with each other without a trusted intermediary, in a verifiable manner [11]. Due to the salient features of blockchains, namely decentralization, anonymity, and security, they are quite suitable for addressing

the IoT security issue. However, there are also several problems when in using blockchains for IoT including the following:

- Mining is computationally too intensive for resource-restricted IoT devices.
- Mining is too time consuming for IoT with low-latency requirements.
- Blockchains scale poorly with the number of nodes, while IoT networks contain a large number of nodes.
- Blockchain protocols create too much traffic overhead for bandwidth-limited IoT devices.

As shown in Fig. 2, Dorri et al. [4] have proposed a block-based IoT architecture with three tiers, namely local blockchain, shared blockchain, and overlay blockchain. All of the three tiers of blockchains do not compute any Proof-of-Work and thus there is no overhead of any mining (actually, double spending is not an issue in IoT, thus there is no need of mining). Forking is allowed at all three tiers, which is a desired feature for IoT. Local and shared blockchains are private and new blocks are not verified. Overlay blockchain is public and blocks must be verified. Local blockchain do not use any encryption, while shared and overlay blockchains use public/private keys and shared keys.

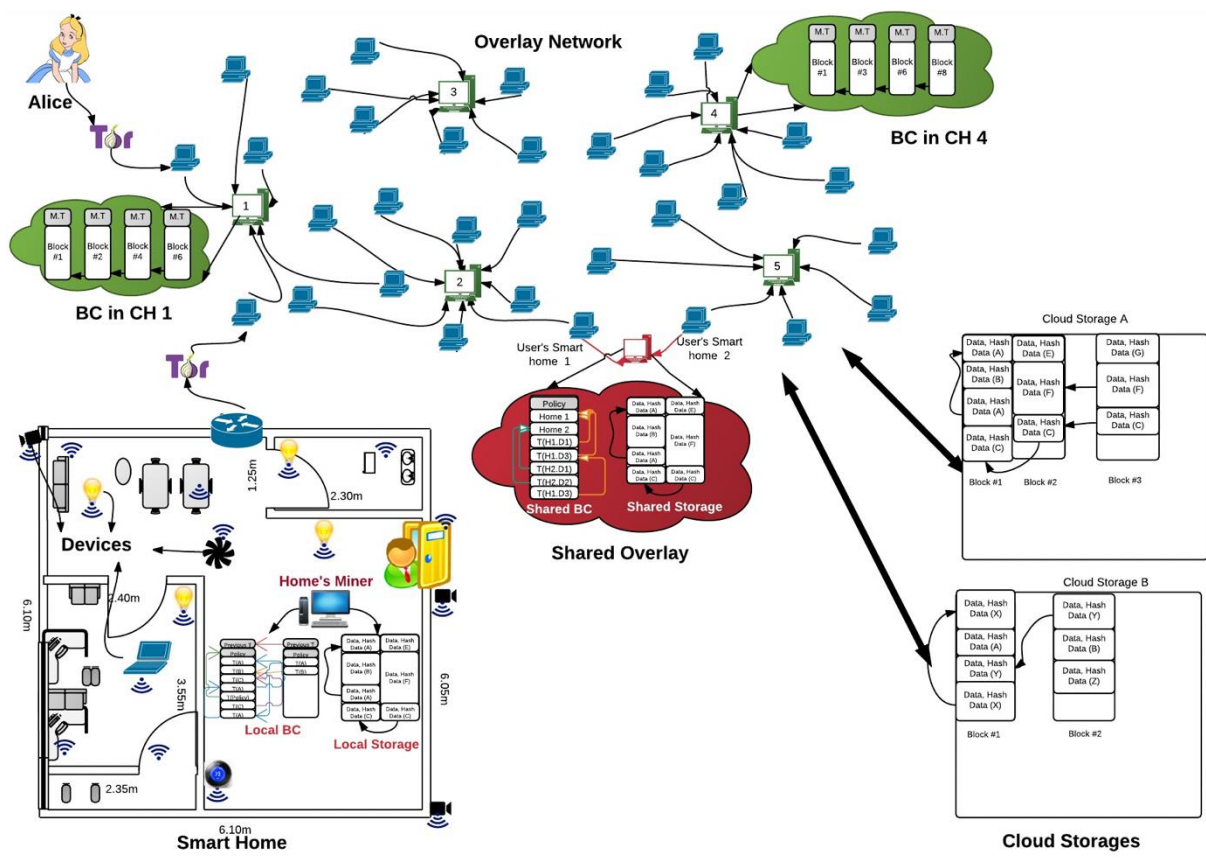


Figure 2. Block-based IoT Architecture [4]

(2) Deep Learning for Anomaly Detection in IoT

Anomaly detection techniques depend on the methods used and the problem characteristics [6]. Methods include machine learning, data mining, statistics, information theory, and spectral theory. Problem characteristics include the nature of data, labels, anomaly type, and the output. Data can be

sequential (e.g., time-series), spatial, and graph. In IoT, we will be targeting at spatio-temporal data, which are time-series, as well as, spatial. Anomalies can be classified into *point* anomalies, *contextual* anomalies, and *collective* anomalies. In IoT, all 3 kinds of anomalies could occur. For example, a malfunctioning sensor could exhibit transient errors in the form of point anomalies such as a sudden change in temperature data from a sensor. Heavy rainfall detected suddenly at a sensor while there was no rainfall in the previous time slot at that sensor and no rainfall was detected by any neighboring sensor. In this spatio-temporal context, the heavy rainfall is a contextual anomaly. An unexpected data pattern could be a collective anomaly in IoT. For example, all sensors in some area on a mountain slide are either not transmitting or transmitting unexpected data. This means there might be some problem in that area, for example, heavy rainfall affecting wireless network communication or local mud-slide that has resulted in all sensors dis-located from their original positions. The output of anomaly detection can be either a score indicating a data's degree of anomaly or a label of normal or anomalous.

Neural networks (NN) have been used for anomaly detection for more than two decades, with corresponding research dating back to 1994 [12]. Several variants of NN have been used including multi layered perceptrons, neural trees, autoassociative networks, adaptive resonance theory based, radial basis function-based, Hopfield networks, Oscillatory networks, and replicator networks. However, for sensor networks NN has not been used much.

Only recently, DNN in the form of *Long Short Term Memory Recurrent Neural Network* (LSTM RNN) has become popular in anomaly detection [13]. The main reason is that LSTM RNN can be used to both remember, as well as, forget past data. Normally, there is an input gate, an output gate, and a forget gate between two memory cells. The forget gate is self-recurrent, thus the name. As shown in Fig. 3, an LSTM block has all the three gates and data could be new inputs or recurrent data. LSTM is a variant of RNN that can effectively solve the problem of gradient vanishing or gradient explosion by introducing a set of memory units. Collective anomaly can also be detected using LSTM RNN [13]. The prediction errors can be assumed to be of Gaussian distribution, thus the Gaussian probability density function for the training set of errors can be used to compute the conditional probability that an attribute occurs in the presence of a certain class. Naïve Bayes principle is then used to detect anomaly. LSTM combined with Gaussian and Naïve Bayes is shown to have a very high accuracy, precision, recall, and F1.

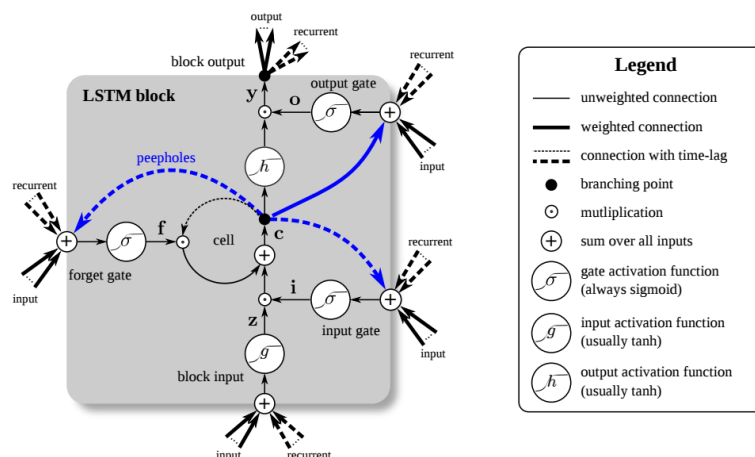


Figure 3. Long Short Term Memory Block in Recurrent Neural Networks

(3) Smart Grid Security

Our target system is Smart Grids. Since smart grids span several levels of system hierarchy including sensors, networks, and cloud, security mechanisms are enforced at each level of the hierarchy. According to NIST's conceptual model, the Smart Grid consists of seven logical domains: Bulk Generation, Transmission, Distribution, Customer, Markets, Service Provider, and Operations. As shown in Fig. 4, information flows across all 7 domains, with 2-way information between the first 4 domains and information collection and power management in the last 3 domains. Nevertheless, anomalies might arise in any of the information flows, even though encryption is throughout the network.

One anomaly that is of utmost importance and must be detected in smart grids is Denial of Service (DoS) attacks, which can be further distinguished into traffic flooding at the application or network level, buffer flooding at the transport layer, ARP spoofing at the MAC layer, and jamming in substations at the physical layer.

To mitigate against DoS attacks, there are normally 4 types of detection methods, including signal-based, packet-based, proactive, and hybrid. In the signal-based detection, at the physical or MAC layer, the received signal strength information (RSSI) is used to detect the presence of an attack, for example, wireless jamming. If the RSSI of many packets is larger than a threshold (means the packets should be correctly received), but the packet decoder outputs errors, an attack is detected. In the packet-based detection, a significant increase in packet transmission failures can indicate an attack. In the proactive detection, probe packets are sent proactively to test or measure the status of potential attackers. In the hybrid detection, different detection techniques mentioned above can be combined into one method for detection DoS attacks.

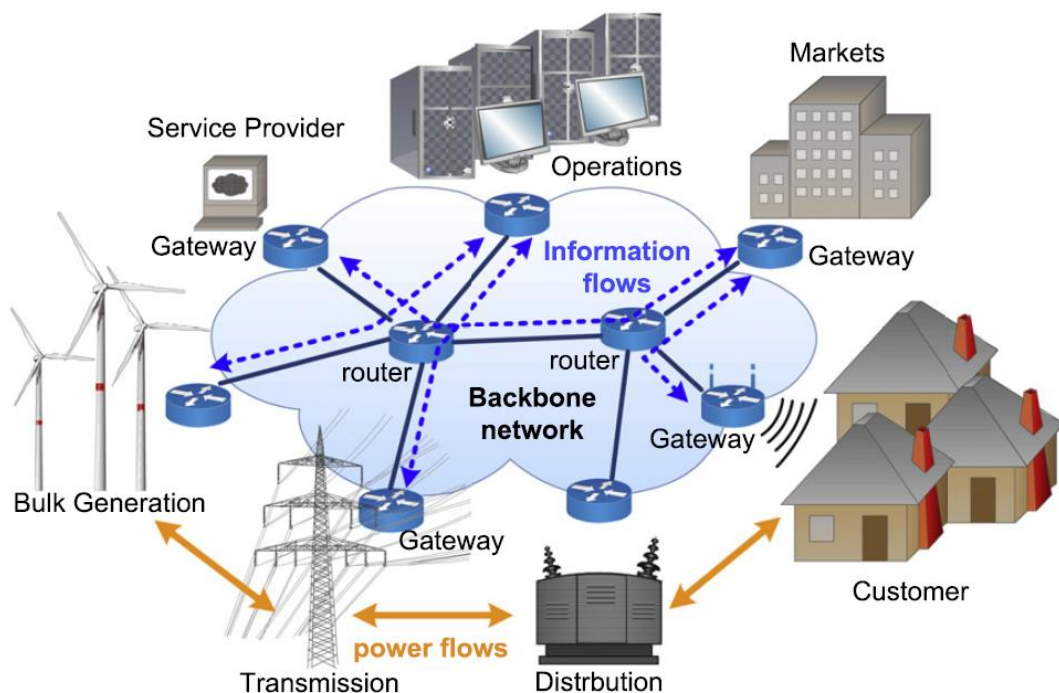


Figure 4. The Network Architecture of the Smart Grid: 7 logical domains

(二) 研究方法、進行步驟及執行進度。請分年列述：1.本計畫採用之研究方法與原因及其創新性。2. 預計可能遭遇之困難及解決途徑。3.重要儀器之配合使用情形。4.如為須赴國外或大陸地區研究，請詳述其必要性以及預期效益等。

1. Research Method

As shown in Fig. 5, the architecture of the smart secure Fog-based IoT proposed in this project is composed of 3 layers, namely *cloud*, *edge*, and *device*. At the cloud layer, the project will accomplish 3 tasks: (a) construct a cloud eco-system for the smart secure Fog-based IoT based on existing public cloud services, (b) train the LSTM RNN model with application IoT data, and (c) construct a blockchain security server platform. At the edge layer, we will accomplish the following 3 tasks: (a) propose a heterogeneous data fusion method, (b) use the trained LSTM RNN model for anomaly detection, and (c) design the blockchain node with transaction initiation. At the device layer, we will accomplish two tasks: (a) create a sensor device simulation too, and (b) inject anomalous data and sensor faults into the simulator to setup an environment for testing whether our security mechanisms work. Further, in this project, we will also create a target application, namely smart grid with smart blockchain (SGSB). In the following, we will describe in details how the project accomplishes the above-mentioned tasks including cloud eco-system development for Fog-based IoT, LSTM RNN model construction, training, and use for anomaly detection in IoT, blockchain security platform design, and SGSB target application design.

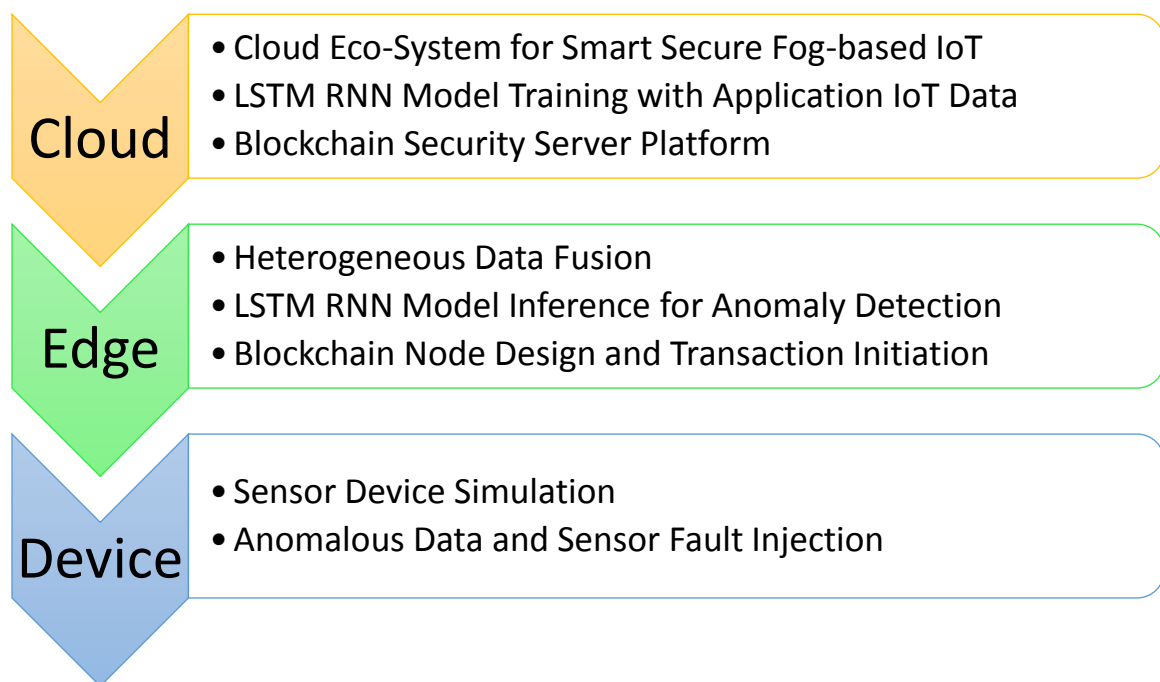


Figure 5. Three Layer Architecture for the Proposed Smart Secure Fog-Based IoT

(1) Cloud Eco-System for Smart Secure Fog-based IoT

Our target system architecture is Fog-based IoT, thus we plan to develop our project cloud eco-system using ARM Mbed platform, which is the most suitable for IoT design. The suitability of this

choice is also evidenced by the decision of Advantech's WISE series of IoT gateways and sensor devices to use ARM Mbed.

ARM Mbed has 3 parts: Cloud, Client, and OS. We will focus on ARM Cloud services, which include Cloud Device Management, Software Update Service, and End-to-end Security Services as described in the following.

- **Cloud Device Management:** This includes device registration, remote monitoring, and data visualization. Our IoT network edges can thus be monitored and data visualized via this device management facility. Data visualization would be especially beneficial when we compare the effects of edge analytics with deep learning.
- **Software Update Service:** Since our target application is Fog-based, especially focusing on smart grid, software updates for our sensors would be very useful, especially for configuring or programming our LSTM RNN models.
- **End-to-end Security Service:** ARM TrustZone separates the secure and non-secure worlds in hardware and keeps the non-secure software blocked from accessing secure resources directly. Switching between the two worlds is controlled by the network edges. In our project, this end-to-end security service will be employed in LSTM RNN-based anomaly detection.

In summary, the project will develop an ARM Mbed based eco-system for our target application, namely smart grid with smart blockchain.

(2) LSTM RNN Model Construction, Training, and Use

We will formulate the overall design of vertically shared deep learning infrastructure as shown in Fig. 6, which consists of three levels, namely cloud, edge, and device.

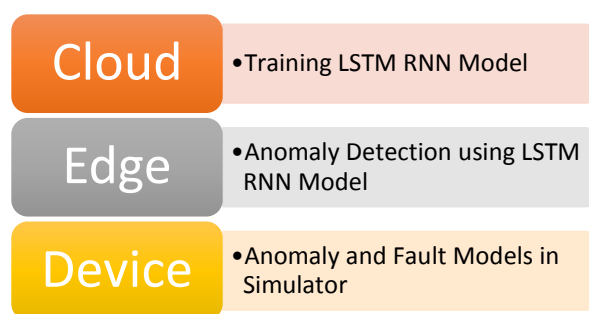


Figure 6. Shared Deep Learning in Fog-based IoT Architecture

Within the three 3 levels of vertically shared deep learning for Fog-based IoT, we will accomplish three tasks, including (a) at the cloud level, construct and train LSTM RNN model with application IoT data, for example, smart grid anomaly and fault data, (b) at the edge level, use the trained LSTM RNN model to test anomaly detection for smart grid applications, and (c) at the device level, create a simulator for anomaly detection and fault model simulation.

We plan to use CPU/GPU to perform the model trainings and hyper-parameter tuning at the cloud level. At the edge level, we will use a user-equipment server that often accompanies network gateways and is basically the fog computing edge resource. At the device level, the simulator will try

to simulate a Fog-based IoT with a large number of nodes that can be injected with sensor faults and data anomalies.

For deep learning to have a high accuracy, sources of large datasets are required. Our sources of data for this project will be from the following:

- Open Energy Data: There are several organizations that provide open data on energy resources including:
 - data.gov.fr: 28 datasets on consumption of buildings, photovoltaic production on communal buildings, windpower census per region, biogas plants, household/industrial/company electricity prices, natural gas prices, heavy fuel oil prices, etc.
 - Enedis Open Data: 16 datasets including daily electric report, half-hourly electric report, installed capacity, quality of supply, coefficients of technical loss, etc.
 - RTE: 21 datasets on power consumption, production per industry, power exchanges in Europe, regional power balances, etc.
- Simulated Data Sources: Since we will develop our own Fog-based IoT simulator for fault and anomaly detection, we will use the simulator to generate data to train our LSTM RNN model in the cloud server. Enough data must be collected. Further, we also need to formalize the model so that the inputs and outputs of the model can be determined before the data are collected from the simulator.

(3) Blockchain Security Platform Design

In a Fog-based IoT system, the number of physical sensor devices can be in the order of thousands or millions and they all have very restricted computation and energy capacities, thus directly incorporating physical sensor devices into a blockchain platform is not a feasible architecture. In this project, a novel two-tier architecture for smart blockchain security platform is proposed for Fog-based IoT. The two tiers are at the network edge and at the cloud server levels. Smartness is introduced at both levels. In the following, we will briefly describe the proposed plan.

- Agent Layer: The agent layer consists of a set of virtual agents that take sensor data as input, process data, and generate transactions, and/or trigger smart contracts. A virtual agent represents a group of physical sensor devices with a common task of sensing some kinds of data. For example, in the blockchain security platform design for smart grid IoT, a *load agent* collects and processes data related to the amount of power consumed by a group of power loads such as those in a certain home, company, or factory. When the average load consumption as processed by the load agent is higher than a given threshold, in a particular context (e.g., for a specific period of time, etc.), the agent creates a *contextual anomaly detected transaction*.

As shown in Fig. 7, it is planned that each agent will be implemented as an application deployed to a Docker container and conforming to the industry standard for distributed control systems IEC 61499. Function blocks in an IEC 61499 node will be used to implement the data

processing, along with heterogeneous data fusion, and transaction creation procedures. A special virtual node will also be associated with LSTM RNN model such that the results of DNN model prediction can also be used to trigger the creation of transactions or the initiation of smart contracts. The current plan is to implement the agent layer at the network edge (user-equipment server at the gateway level); however, the actual computation resources will be in the cloud.

Since the IEC 61499 nodes are deployed into Docker containers, we plan to manage the containers using the *Kubernetes orchestration platform*. Each node in Kubernetes can be deployed to a computation resource in the cloud. Each Kubernetes Pod (smallest execution unit) is a pool of containers which can be deployed on one or more nodes.

In our proposed smart blockchain security platform, a Pod will consist of at least two containers, one for the IEC 61499 node that represents software data processing and one IEC 61499 node that encapsulates a LSTM RNN model so that the DNN model can be virtualized and re-used across different Pods (applications). Figure 6 depicts the architecture of the proposed agent or node layer in the blockchain design.

As shown in Fig. 7, we will be using 3 types of sensors and simulating them, including the current clamps (for measuring the amount of current flowing through an electricity cable) which will be used for load consumption recording, the weather sensors for collecting temperature, wind speed, humidity, etc. that are required for estimating the amount of power that can be generated by photovoltaic cells, turbines, etc. The last type of sensors are battery state-of-charge. Using these 3 types of sensors, we can get an accurate model of the inputs for the LSTM RNN model, which can then deduce or predict if there is any anomaly in the IoT system.

In the sigmoid function of the output layer of the LSTM RNN model, we plan to define the output of the model as a value between 0 and 1 such that it represents the probability of anomaly. A threshold will have to be selected for actually making the predicted results into an actionable data. The threshold will depend on the application. For example, for our smart grid, the threshold could also depend on what kinds of anomalies we are targeting at. Smart meter readings might have a larger disruption in the readings; however, with a context (the average temperature of that period, or the season: summer or winter) the meter readings could be more stable. The threshold will thus be chosen based on the application and the target data.

Since the agent layer will be implemented on a fog computing platform, we have to ensure that the computation capability of the selected fog computing platform is sufficient for executing the function blocks in the IEC 61499 nodes. Though a Kubernetes Pod will be deployed on a Kubernetes node, which in turn is actually a cloud server, the function blocks could be executed partially on the fog computing platform and partially in the cloud. This workload distribution or offloading will depend on the application itself that is implemented in the proposed blockchain architecture.

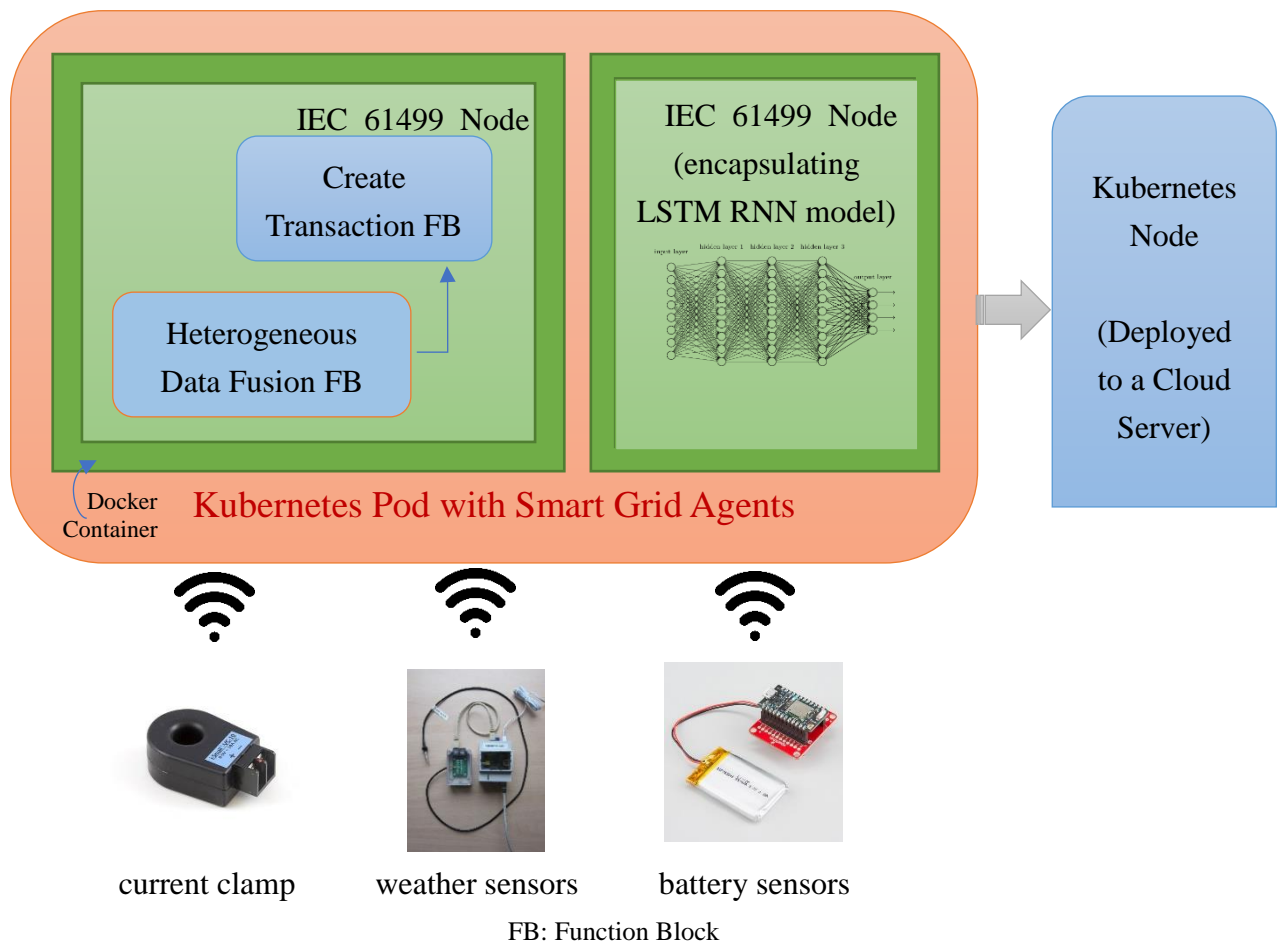


Figure 7. Proposed Agent Layer in the Blockchain Design of Fog-based IoT Architecture

- **Ledger Layer:** The ledger layer consists of the actual blockchain data, that is, all the inter-chained blocks. We plan to use the HyperLedger Fabric to implement the blockchain transactions, blocks, and smart contracts. Similar to Kubernetes, HyperLedger Fabric is also open source, thus we will try to customize and integrate them into a smart blockchain security platform for Fog-based IoT.

HyperLedger Fabric (github.com/hyperledger/fabric) is an implementation of a distributed ledger platform for running smart contracts, leveraging familiar and proven technologies, with a modular architecture allowing pluggable implementation of various functions. It is a permissioned blockchain platform aimed at business use [11]. It supports strong security and identity features based on enrollment certificate authority (CA) and transaction CA. Confidentiality is provided through symmetric key encryption of transactions and states with a blockchain-specific key that is available to all peers with an enrollment certificate for the blockchain. The HyperLedger Fabric supports a modular pluggable implementation of the consensus protocol. It currently has implemented the Practical Byzantine Fault-Tolerant (PBFT) Consensus protocol [14], which is quite prominent in distributed consensus research and has been analyzed in many environments. In this project, we will first evaluate if PBFT is suitable for our Fog-based IoT blockchain consensus. If not, we will customize it for our IoT.

From previous work [11], we can observe that using HyperLedger Fabric v0.6 and PBFT consensus on 4 validating nodes, at least 170 query transactions per second and 255 invoke

transactions per second can be executed for a machine with 2 vCPUs and 7.5 GB memory. This shows that given the small memory footprint and high performance of HyperLedger Fabric, adapting it to our CPS IoT with edge computing should be feasible.

(4) Smart Grid with Smart Blockchain

We will implement a target application called *Smart Grid with Smart Blockchain* (SGSB) as a testbed for IoT security experiments and for assessing the effectiveness of the proposed vertical sharing of deep learning and the blockchain security for Fog-based IoT architecture. As shown in Fig. 7, the SGSB application has 3 levels, namely cloud, edge, and device as described in the following:

- At the device level, sensor devices will be simulated with sensing electricity usage, weather features, and battery states, which will be sent via a Low-Power Wide-Area Network (LPWAN) protocol such as LoRaWAN or NB-IoT or SigFox to the edge level nodes.
- At the edge level, we will implement the agent layer of the proposed blockchain security platform using the Kubernetes orchestration platform and the Docker containers, while conforming to the industry standard for distributed control IEC 61499. Each IEC 61499 node will be implemented as an independent application and deployed in a Docker container. All containers will be organized and orchestrated using the Kubernetes platform for ease of management and control.
- At the cloud level, we will implement the ledger layer of the proposed smart blockchain security platform using the HyperLedger Fabric and PBFT consensus protocol.

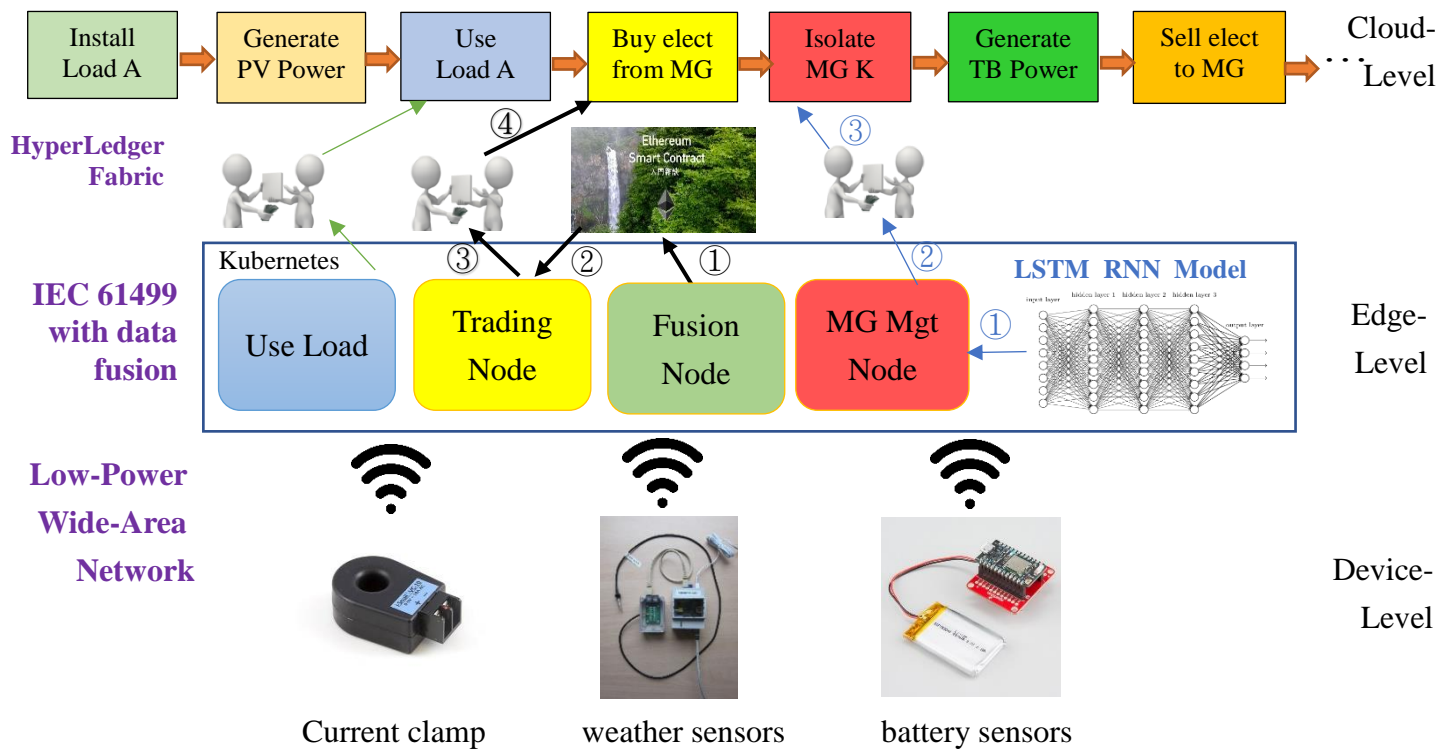


Figure 8. Proposed Smart Grid with Smart Blockchain Application

As depicted in Fig. 8, the blockchain ledger implemented in the cloud level will be the distributed security platform for our target Fog-based IoT system design. We will implement both transactions and smart contracts as shown respectively using blue and black numbers in Fig. 7, which will be

described in the following.

- Transactions:

The LSTM RNN Model will be trained in the cloud level using micro-grid power load consumption, power generation, and battery storage data, that is, the amount used, generated, and stored per hour of each day. The trained LSTM RNN model will then be used at the edge-level for predicting if any micro-grid (MG) is behaving abnormally, for example, in Fig. 7, micro-grid k is using too much power in the wrong context, e.g., time of day, location, etc. (numbered as a blue ①). Then, a transaction is generated by the micro-grid management node (MG Mgt. Node), numbered as a blue ②. The transaction will be committed into the blockchain within the “Isolate MG k” block, numbered as blue ③. Here, for simplicity we only consider one transaction in one block. However, in actual implementation, multiple transactions will be encapsulated (hashed) into one block. This illustrates how DNN can be combined with blockchain technology, through transactions that are initiated by DNN. Thus, blockchains are called “smart blockchains” in this project to emphasize how transactions are initiated a priori to actual events occurring and actions can be taken beforehand. For example, large power consumption by a micro-grid (predicted by DNN) results in a transaction to isolate that micro-grid, even before it has caused any actual problem (voltage instability, blackouts, etc.). This the blockchain proposed in this project derives its innovation on this unique integration of DNN with blockchains.

- Smart Contracts: Similarly, the 4 black numbers indicate how a smart contract is initiated and executed. Smart contracts are pre-determined transactions that are automatically generated and committed based on some triggering conditions. In our smart grid with smart blockchain application, a data fusion node integrates heterogeneous data from multiple types of sensors, the results of which could result in events, such as a deficit or a surplus of electricity. In Fig. 7, we can see the fusion results indicate a deficit of electricity in the future (ref. ①) and if the price of electricity is low enough now, then a smart contract (②) that specifies whenever there is a *deficit* of electricity, we should automatically create a “buy electricity” transaction (③) and execute it ④. Thus, the programming of smart contracts will be designed into our smart blockchain through the use of the constructs available in HyperLedger Fabric.

2. Issues and Possible Solutions

The issues that we might encounter in this project will be mostly related to two areas, namely the integration issues and the technical research issues. In the following we will digress on both of these and also propose candidate solutions.

(1) Integration Issues

There are two main integration issues, including cloud server eco-system development issues and the target application smart grid with smart blockchain integration issues, which we will describe in the following.

- Cloud Server Eco-system Development Issues: Though ARM Mbed provides useful cloud services such as device management, software update, and end-to-end security, we still need to

resolve issues related to the following:

- What is an essential set of web services for a Fog-based IoT design? This will depend on the application domain; however, for our first target application SGSB, we will try to come up with the essential set, which might include node data storage/retrieval, blockchain transaction processing, blockchain smart contract processing, LSTM RNN model parameter passing, and other management services.
- How to integrate the set of web services for IoT with ARM Mbed? Especially, the part related to blockchain might need a big overhaul of the Mbed interface, which needs to be estimated first. Further, RNN configuration and programming with trained parameters should also be integrated via Mbed's software update interface.
- Target SGSB Application Integration Issues: The issues in setting up the target application would include the following.
 - IoT Simulator Design: Anomaly detection and fault simulation are the main purposes for developing a simulator for IoT from sensor to edge processing and connection to the cloud servers. A framework design that meets the security testing requirements while accurately simulating a fog-based IoT is required. So, the question is to what degree of details in data sensing, processing, fusion, and transformation into transactions should the simulator be designed. To keep the abstraction level of the simulator flexible, we will like to use Multi-Agent System (MAS) as the basic architecture for IoT simulation, which is also quite suitable for Smart Grid IoT. We have lots of experience in building this kind of MAS simulators for smart grids [16][17][18].
 - Integrating DNN with blockchain: As shown in Fig. 6 and Fig. 7, DNN can be encapsulated as an IEC 61499 node and used to trigger blockchain transactions. In Fig. 7, we can see that because the LSTM RNN detected an anomaly (high voltage) in a certain micro-grid, thus a transaction to isolate that micro-grid was initiated and finally committed into a block on the transaction. Thus, not only trading actions (buying and selling electricity among micro-grids or between a micro-grid and the main power generation called the utility) can be registered into the blockchain ledger in the cloud. The question was how the predictions or detections by LSTM RNN should be integrated with transactions.

(2) Technical Research Issues

With specific focus on our two main goals, the technical research issues that we might encounter in the project are as follows.

- How to share learning across the different levels of deep learning hierarchy? Though the scheme is to construction and train the LSTM RNN model in the cloud and to program the model for inference use in anomaly detection at the edge level, the research issue is how to program the model? We plan to use deep neural programming (DNP) [15] as a possible way to program the DNN model for inference.

- How to ensure that the smart blockchain security platform is suitable for Fog-based IoT? Suitability of the smart blockchain security platform can be defined in terms of computation and power requirements. An initial estimation with HyperLedger Fabric implementation shows that it can achieve quite a good number of transactions within one second (170 to 255). However, as mentioned in the previous section there are several ways in which we can reduce the computation and power requirements of the smart blockchain security platform. For example, instead of using Proof-of-Work, we can use the BFT consensus protocol. Instead of the full ledger, we could design partial views so that ledger maintenance can be distributed. Since there is no issue of double spending as in cryptocurrencies, we can shorten the latency between transactions so that actions can be committed more efficiently in IoT. Multiple tiers (local, block, overlay) can be used to create side channels or forking so that the blockchain computation can be distributed even further.
- As investigated and discussed earlier in this proposal, there are several open questions in the design of the blockchain for IoT, which have to be resolved in this project. We once again list the open questions in each of the different planes of a blockchain architecture, namely network, consensus, storage and view planes.
 - Network Plane: *The open question is how to design and implement blockchains such that its use of the limited bandwidth in IoT can be minimized to that required for actual transactions only.* A P2P solution is proposed by Khan et al [1], but there is no real design or validation of the P2P concept for IoT. Another solution in Block-based IoT architecture by Dorri et al [4] employ a 3-tier approach (local, shared, and overlay blockchains) and traffic is reduced by unicasting transactions in local and shared blockchains and by arbitrary selection of transactions in the overlay blockchain. Both of these solutions are not optimal solutions to the open question. We will try to resolve this question in this project and details will be provided in the next section.
 - Consensus Plane: Bitcoin mining in the form of Proof-of-Work (PoW) is very compute intensive, which is not at all suitable in any network edge devices. *The open question is what kind of consensus algorithm is suitable for IoT, especially those with fog computing edges.* Proof of stake (PoS) is claimed to be thousands of times more cost effective than PoW; however, its convergence (resolution of consensus) is not guaranteed. Another proposal is the Byzantine Fault Tolerant (BFT) replication protocol with a small number of pre-designated trusted entities that greatly enhances scalability (throughput of 4.5K tx/sec with average transaction latency of 1.79 sec for 64 nodes processing batches of 8192 transactions, each of which is 190 bytes long). One more intuitive proposal is sharding protocols, where the task of consensus is split up among concurrently operating sets of nodes such that throughput is increased and per node processing and storage requirements reduced. Sidechains have also been proposed, which are delegations of trust across multiple tiers of consensus instances. In the block-based IoT architecture [4], Beta Reputation System is used for direct and indirect evidence of trust among distributed nodes. All of these are only proposals for reducing the computation and for increasing scalability; however, which one is suitable for IoT is still an open question.
 - Storage Plane: The storage for the ledger can be a problem in IoT edge devices due to the

limited storage space and bandwidth for communication (downloading ledger, etc.). *The open question is how to store the ledger in an IoT architecture such that its contents can be authenticated during read operations.* Sharding the storage of an *unspent transaction outputs* (UTXO) data structure is a possible solution.

- View Plane: A view is a data structure derived from the full ledger whose state is obtained by applying all transactions. New miners in bitcoin need 4 days to download the full ledger and reconstruct the UTXO set (a view). *The open question is how much of the full ledger need to be stored as a view by the edge devices in a fog-based IoT architecture.* Some proposed solutions include views via replication and outsourcing views via cryptography [3] with view authentication using succinct non-interactive arguments of knowledge (SNARKs). However, which solution is more suitable for IoT or if a new solution is required, this is still an open question.

3. International Collaborations

In the course of this project, we will also work with our international collaborators. Currently, we have partners in India, including the Chitkara University, where our partner teams led by Prof. S.N. Panda and Dr. Rajinder Singh are interested in working on sensor simulation for IoT design. Besides Chitkara University, we will also work in collaboration with another research team led by Director Prof. Sunil Kumar Khatri of the Amity University, Noida, India in terms of the blockchain design for smart grid system design.

(三) 預期完成之工作項目及成果。

1. 預期完成之工作項目

The yearly deliverables for the project are as given in Table 2, which includes cloud eco-system design, deep learning (LSTM RNN model), blockchain platform design, and the Smart Grid Smart Blockchain (SGSB) target application design. The project requires 3 years to complete. The main innovations are as described at the start of this proposal, we answer all the “open research questions” related how blockchain can be customized for IoT and how deep learning can be integrated with blockchain.

Not only are these innovations will be the main contributions of the project, we will also create real working systems that will be made into final marketable products at the end of 3 years. We will strive to produce production-ready implementations of cloud eco-system, deep learning models, blockchain platform, with special focus on our target smart grid application. We have several year experience in smart grid system design, thus we are sure we can be able to design the target application into a real product for the smart grid market.

Figure 9 shows the how the major tasks in the project for the 3 years are inter-correlated. The sensor simulator design will depend on the anomaly model and the sensor fault model. The results of the sensor simulator are input to the Blockchain Agent Layer, which is constructed using Kubernetes and Docker, while conforming to IEC 61499. The LSTM RNN model will support the DNP programming environment so that the trained parameters can be downloaded from the cloud and configured into the LSTM RNN model on the fog computing platform. This LSTM RNN model will interact with the Blockchain Agent

Layer. Further, the Agent Layer will then interact with the Blockchain Cloud Layer which is implemented using HyperLedger Fabric. The LSTM RNN Model Training Environment interacts with the model DNP environment.

Table 2. Yearly Deliverables of the Project

	1 st Year	2 nd Year	3 rd Year
<i>Cloud Eco-System</i>	Implement ARM Mbed cloud services (SDK) for Fog-based IoT design	Customize to SGSB target application	Optimize into business-ready product of IoT cloud eco-system design
<i>Deep Learning</i>	LSTM RNN model construction / training and integration with Mbed	Develop programming environment of LSTM RNN	Optimize LSTM RNN Training and Programming Environments
<i>Blockchain Platform</i>	<ul style="list-style-type: none"> ● HyperLedger Fabric Implementations in Cloud ● Agent layer implementation at edge level using Kubernetes/Docker 	Customize to SGSB target application (refer to Fig. 7) including customized transactions and smart contracts	Optimize platform for production of business-ready IoT blockchain design
<i>SGSB</i>	Fault simulator design for smart grids	Integrate with Blockchain platform, LSTM RNN model, and eco-system.	Production-ready optimization for Smart Grid with Smart Blockchain

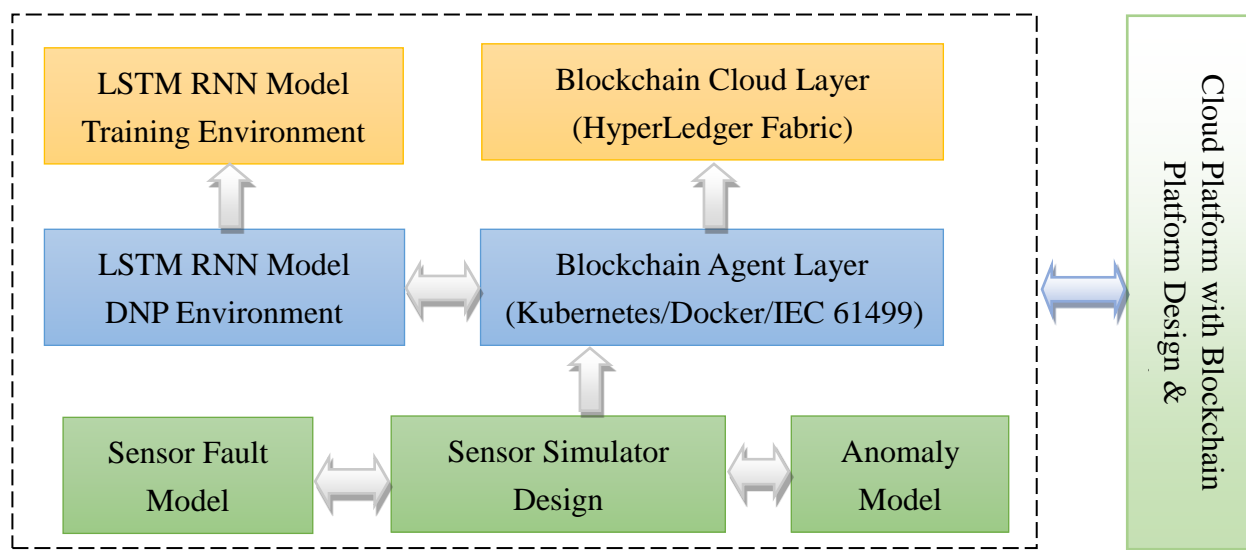


Figure 9. Inter-Relationship among the different tasks in the 3-years of the Project

Within this project, we also plan to collaborate with our industry partners, especially with Metalligence Technology (智上科技公司) and with INER (經濟部核能研究所綠能組). Metalligence Tech is a company devoted to Building Energy Management and the systems that they deploy also need







security measures which include anomaly detection, thus they are very much interested in our project and we plan to work with them on the final results of this project. We also aim to have some technology transfer at the end of the project.

The main deliverables of this project for each of the 3 years of the project execution are as follows.

(1) First Year Tasks and Deliverables (2018/8~2019/7)

- Implement ARM Mbed cloud services (SDK) for Fog-based IoT design.
- Construct and train LSTM RNN model
- Integrate the LSTM RNN model with ARM Mbed
- Implement Smart Blockchain using Hyper Ledger Fabric
- Use Kubernetes and Docker containers to implement the agent layer in blockchain
- Design and implement a fault simulator for smart grids






Table 3. First Year Schedule (2018/8 ~ 2019/7)

Work	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	
1. Implement ARM Mbed cloud services													
2. Construct/train LSTM RNN model													
3. Integrate RNN with ARM Mbed													
4. Design Blockchain Platform using HyperLedger Fabric													
5. Implement Agent Layer using Kubernetes/Docker													
6. Design Fault Simulator for SG													

(2) Second Year Tasks and Deliverables (2019/5~2020/4)

- Customize ARM SDK to Smart Grid application
- Develop DNP programming environment of LSTM RNN
- Customize Blockchain to Smart Grid application, including transactions and smart contracts
- Integrate with blockchain, LSTM RNN, and eco-system for Smart Grid application
- Use fault simulator to test the RNN model for anomaly detection





Table 4. Second Year Schedule (2019/8 ~ 2020/7)

Work	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
1. Customize ARM Mbed to smart grid												
2. DNP Environment for LSTM RNN												
3. Customize BC to Smart Grid												
4. Integrate BC, RNN, and eco-system												
5. Use simulator to test RNN model												

(3) Third Year Tasks and Deliverables (2020/8~2021/7)

- Optimize IoT cloud eco-system for product design
- Optimize LSTM RNN model
- Optimize platform for blockchain product design
- Test all optimizations

Table 5. Third Year Schedule (2020/8 ~ 2021/7)

Work	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
1. Optimize IoT cloud eco-system for product												
2. Optimize LSTM RNN model												
3. Optimize platform for blockchain product												
4. Test all optimizations												

2. 對於參與之工作，預期可獲之訓練

The students and engineers participating in this project will be able to learn not only the technologies related to blockchain design, but also how deep learning can be hierarchically shared across different levels of the design, and how the design blockchain security can be integrated with deep learning. The convergence of blockchain, deep learning, and IoT will be the main focus in this project and thus the innovations created as a result of the project will lay the foundations of future technologies.

3. 預期完成之研究成果（如實務應用績效、期刊論文、研討會論文、專書、技術報告、專利或技術移轉等質與量之預期成果）

The results of this project will mainly consists of three products, namely Smart Grid with Smart Blockchain application, IoT Blockchain Platform, and LSTM RNN Model for IoT. The corresponding technologies in these products will be hierarchical deep learning technology and smart blockchain security technology. Besides top journal and conference publications, we will focus on the actual practical contributions the project has over the current state-of-the-art technology in IoT design, especially for Cyber-Physical Systems such as smart grid system, landslide detection system, and smart traffic system. Technology transfers on both the blockchain design and DNN model design will be made at the end of the 3rd year.

4. 學術研究、國家發展及其他應用方面預期之貢獻

Since our target application is smart grid, we believe that at the end of this project, we will be able to provide smart grid companies with a new technology that can help make smart grids more secure. This will have a very huge benefit to the economic value of smart grid inside and outside Taiwan. In the future, our target market will also include other South Asian countries such as India and Vietnam, where smart grids are becoming more and more prevalent.

References

- [1] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *Journal of Cloud Computing: Advances, Systems and Applications*, Vol. 6, No. 1, August 2017.
- [2] J. Yii-Huumo, D. Ko, S. Choi, S. Park, and K. Simolander, "Where is current research on blockchain technology? – A systematic review," *PLoS ONE*, Vol. 11, No. 10, , October 2016.
- [3] K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gun Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Procs. of the International Conference on Financial Cryptography and Data Security*, February 2016.
- [4] A. Dorri, S.S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and Solutions," *CoRR*, arXiv:1608.05187v1, <http://arxiv.org/abs/1608.05187>, August 2016.
- [5] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep learning neural network for in-vehicle network security," *PLoS ONE*, Vol. 11, No. 6, June 2016.
- [6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, Vol. 41, No. 3, Article No. 15, July 2009.
- [7] X. Xie, D. Wu, S. Liu, and R. Li, "IoT data analytics using deep learning," *CoRR*, arXiv:1708:03854v1, <http://arxiv.org/abs/1708.03854>, August 2017.
- [8] A. F. Skarmeta, L. H.-R. Jose, and M. Moreno, "A decentralized approach for security and privacy challenges in the internet of things," in *World Forum on IoT*, 2014.
- [9] H. Gross, M. Holbl, D. Slamanig, R. Spreitzer, "Privacy-Aware authentication in the internet of things," *Cryptology and Network Security*, Springer International Publishing, pp. 32-39, 2015.
- [10] A. Ukil, S. Bandyopadhyay, A. Pal, "IoT-Privacy: To be private or not to be private," in *IEEE Conference on Computer Communications Workshops*, 2014.

- [11] A. Stanciu, “Blockchain based distributed control system for edge computing,” in 21st International Conference on Control Systems and Computer Science, IEEE, 2017.
- [12] C. Bishop, “Novelty detection and neural network validation,” in Proceedings of the IEEE Conference on Vision, Image and Signal Processing, Vol. 141, pp. 217-222, 1994.
- [13] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, “Collective anomaly detection based on Long Short Term Memory Recurrent Neural Network,” CoRR, arViv:1703.09752, <http://arxiv.org/abs/1703.09752>, March 2017.
- [14] M. Castro and B. Liskov, “Practical Byzantine fault tolerance and proactive recovery,” ACM Transactions on Computer Systems, 20(4):398-461, Nov. 2002.
- [15] K. Selyunin, D. Ratasich, E. Bartocci, and R. Grosu, “Deep neural programs for adaptive control in cyber-physical systems,” CoRR, arXiv:1502.04013v1, <http://arxiv.org/abs/1612.07762>, February 2015.
- [16] H.-L. Chao, C.-C. Tsai, P.-A. Hsiung, and I.-H. Chou, “Smart Grid as a Service: A Discussion on Design Issues,” Scientific World Journal, Vol. 2014, Article ID 53508, pp. 1-11, August 2014.
- [17] H.-L. Chao and P.-A. Hsiung, “A Fair Energy Resource Allocation Strategy for Micro Grid,” Microprocessors and Microsystems, Vol. 42, pp. 235–244, May 2016.
- [18] H.-L. Chao, P.-C. Hsieh, T.-C. Yang, and P.-A. Hsiung, “Model Predictive Optimization for Distribution Management in Smart Grids,” Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, October 2016.