Providing Secure and Reliable Communication for Next Generation Networks in Smart Cities

Ismaeel Al Ridhawi[1], Safa Otoum[2], Moayad Aloqaily[3], Yaser Jararweh[4], and Thar Baker[5]
[1]Kuwait College of Science and Technology, Kuwait City, Kuwait, i.alridhawi@kcst.edu.kw
[2]University of Ottawa, Ottawa, ON, Canada, K1N6N5, Safa.Otoum@uottawa.ca
[3]Gnowit Inc., 7 Bayview Road, Ottawa, ON, Canada, K1Y2C5, Moayad@gnowit.com
[4]Duquesne University, Pittsburgh, PA, USA, Jararwehy@duq.edu
[5]Liverpool John Moores University, Liverpool, UK, T.Baker@ljmu.ac.uk.

***Abstract*—Finding a framework that provides continuous, reliable, secure and sustainable diversified smart city services proves to be challenging in today's traditional cloud centralized solutions. This article envisions a Mobile Edge Computing (MEC) solution that enables node collaboration among IoT devices to provide reliable and secure communication between devices and the fog layer on one hand, and the fog layer and the cloud layer on the other hand. The solution assumes that collaboration is determined based on nodes' resource capabilities and cooperation willingness. Resource capabilities are defined using ontologies, while willingness to cooperate is described using a three-factor node criteria, namely: nature, attitude and awareness. A learning method is adopted to identify candidates for the service composition and delivery process. We show that the system does not require extensive training for services to be delivered correct and accurate. The proposed solution reduces the amount of unnecessary traffic flow to and from the edge, by relying on node-to-node communication protocols. Communication to the fog and cloud layers is used for more data and computing-extensive applications, hence, ensuring secure communication protocols to the cloud. Preliminary simulations are conducted to showcase the effectiveness of adapting the proposed framework to achieve smart city sustainability through service reliability and security. Results show that the proposed solution outperforms other semi-cooperative and non-cooperative service composition techniques in terms of efficient service delivery and composition delay, service hit ratio, and suspicious node identification.**

Keywords — Next Generation Networks, Cloud, Fog, Sustainable Smart City.

I. INTRODUCTION

As part of ongoing development and advancement to IoT technology for a more sustainable and improved smart city infrastructure, Fog- and Mobile Edge- Computing (MEC) were introduced as two unique state-of-the-art solutions that provide enhanced and faster service delivery for cloud subscribers [1]. Fog Computing and MEC were first adopted to offload cloud computational and storage facilities closer to mobile users for data pre-processing and then moved towards data and service provisioning lately. Fog nodes act as both clients and servers, such that it can acquire services from the cloud and/or provide services to IoT devices. Traditionally, IoT devices sent sensed data to and/or acquired data and services directly from the cloud, which was time consuming. Certain smart city applications, especially in the health-care or emergency sectors could not tolerate such delays. With the introduction of fog and MEC, data analysis and service acquisition are processed at a faster pace.

Smart cities are growing on a daily basis, and are providing faster, sustainable and more diverse services for societies in a variety of areas such as traffic, health-care, education and much more. Service enhancement and faster service delivery are two ingredients towards a successful sustainable smart city [2]. With the abundance of more powerful and smarter IoT devices such as smart phones and sensors, traditional fog and cloud solutions are no longer optimal solutions. For instance, having many devices connect to the same access point, located at the fog or cloud layers, would cause services to be delivered with increased latencies. Although fog and MEC solutions have supported delay intolerant applications, having cloud data and services replicated to the fog layer introduces extra overhead that many researchers may neglect. Moreover, maintaining such replica sites is another complex issue that must be considered. The concept of using node-to-node communication [3] or distributed solutions [4] to compose and deliver services has been a favoured trend lately [5]. Certainly, if not most, smart city services such as multimedia, social, and vehicular applications, would benefit from next generation node-to-node networks for service delivery, composition, enhancement, analysis and much more [6].

This article introduces a state-of-the-art technique for node cooperation and collaboration. Nodes are examined according to certain cooperation willingness criteria to determine their ability and capability of joining other nodes in delivering enhanced, faster, sustainable and more productive smart city services to users. Those criteria, namely, nature, attitude and awareness are all affected by each other. A learning technique that determines how well a node is suitable for cooperation under certain network conditions is developed. Such technique is proved to be less reliant on offline training, thus ensuring faster compositions and service delivery. The service composition process may occur on three different layers: compositions using mobile nodes only, compositions using a mix of fog/edge devices and mobile nodes, and compositions using fog/edge devices only.

By reducing data traffic to the fog and cloud layer, datacenters and storage sites are not only capable of processing service and data requests at reduced latencies, but also data traffic to and from the fog and cloud can be analyzed for suspicious activities. In this article, in addition to the

cooperative technique introduced, communication between mobile nodes and fog/edge devices is examined for intrusion detection through a data traffic analysis, reduction and classification technique. Using a learning technique, traffic is determined as either trusted or a threat. The same data traffic analysis mechanism is used for communication between fog/edge devices and the cloud. Simulations are conducted on three different service composition and delivery techniques, namely, the proposed mobile node and fog cooperative solution, a cooperative fog solution, and a traditional fog to cloud solution. Results showed the effectiveness of the cooperation and intrusion detection strategies using the proposed technique in terms of efficient service delivery and composition delay, service hit ratio, and suspicious node identification.

The contributions and findings of the proposed work are summarized as follows:

- Introduce a novel node characteristics identification technique that determines a node's willingness and capability to cooperate with other nodes to compose complex cloud services. Node capabilities are determined through its ontological property descriptions, while node willingness uses a fuzzification function to determine its level of intelligent awareness, attitude and nature towards task distribution and collaboration.
- Since the process of determining node willingness and capability may be time-consuming, we develop a reinforcement learning technique to select service composition patterns that have been proven to provide adequate QoS results to speed up the overall composition process.
- Introduce an ensemble-based intrusion detection system (E-IDS) with three base classifiers, namely, Support Vector Machine (SVM), Deep Belief Network (DBN), and eXtreme Gradient Boost (XGBoost). E-IDS is adapted into the fog and cloud, such that intrusion detection and classification are conducted separately at the fog and cloud, respectively.
- Conducted thorough simulations to test the proposed cooperative service composition solution against other cooperative and non-cooperative solutions to determine its superiority and adequacy in terms of efficient service composition and delivery latency, and intrusion detection.

The rest of this article is structured as follows. Section II considers related work in the literature. Section III provides an overview of the problem and the solution through an architecture. Section IV discusses the three cooperation criteria in details. Section V focuses on the reinforcement learning part used for service composition. The secure communication aspect of the solution is looked at in Section VI. Simulation results are considered in Section VII. Finally, Section VIII concludes the paper with some future work considerations.

II. Related Work

To achieve a sustainable service provisioning solution, resource distribution can be adapted to smart city frameworks [7]. Distributing resources at the edge of the network, closer to the user, provides benefits in the form of low latency and service diversity. Most of the work focuses on fog-enabled architectures and application-specific architectures using fog computing [8]. Very little work has been considered on providing sustainable, reliable and secure communication among different cloud and fog entities. In [9], the authors proposed a smart sustainable city framework that integrates cloud, fog, and agent computing together into one architecture. Distributed fog servers are used to host critical services for agents' locations and coordination that act as brokers, matchmakers and facilitators. Edge devices are modeled as autonomous agents that interact with each other and with fog units. Fog servers act as intermediate entities for activates with lower edge devices' agents and with higher-level cloud services and resources. Simulation results show that the proposed solution provides high performance in terms of service response time, if a self-regulating model is used

In [10] the authors proposed a vehicular fog service solution using a three-layered architecture. A central cloud exists on the top layer which consists of a fully trusted authority. Vehicular fogs are placed on the second layer, where each is composed of a road side unit and multiple server-vehicles. The last layer consists of vehicular clients with on-board units equipped on each client-vehicle. Mutual authentication between roadside units and on-board units is used to grant client-vehicles access to the vehicular fog. The authentication is based on the signature from the cloud trusted authority. Malicious behaviour can be detected by neighbouring vehicles in a vehicular fog, which is then reported to the trusted authority for server access revocation.

Li et al. [11] proposed a fog-assisted trustworthy packet forwarding scheme for effective and safe data transmission. The authors design a logical joint edge community model that considers fog nodes to study the influence of fog nodes on data forwarding. According to the results, nodal community activity and social similarity for fog nodes and mobile devices is redesigned to adapt a new set of forwarding rules. The self-adaptive forwarding algorithm uses improved seeds expansion and random walks algorithms to provide data forwarding security and reliable privacy protection. Direct communication between mobile devices is secured against leakage or theft of privacy on information stored on fog nodes and base stations. Contact probability and service degree are used as weight vector elements of links between node pairs to provide the trustworthiness support. Experimental results show enhanced packet delivery ratio and reduced latency.

Tang et al. [12] introduced a hierarchical distributed fog computing architecture to allow for the integration of different smart city infrastructure components and services. Moreover, intelligence through data representation and feature extraction is use to identify anomalous and hazardous events, hence, offering optimal service responsiveness and control. Both supervised and non-supervised machine learning algorithms are adapted to detect data anomalies. A prototype was implemented using smart pipeline monitoring through a four-layered fog paradigm to demonstrate the effectiveness of the solution in terms of service response time and the reduced number of service requests submitted to the cloud.

In [13] the authors proposed a load balancing technique to provide a secure and sustainable load balancing technique of edge datacenters in fog computing. The solution uses an adaptive edge datacenter secure authentication technique with the aid of a centralized cloud datacenter. The authentication process is initiated by the cloud and then all edge datacenters authenticate each other following the cloud. Each load of the edge datacenters is considered through information sharing with edge devices during the authentication process, so that no extra communication overhead is added when retrieving the load information from other edge datacenters. By adapting this authentication process, simulation results show that malicious edge devices are identified and avoided.

Shojafar et al. [14] developed a scheduler for the adaptive tuning of input/output traffic and resource reconfiguration/consolidation of virtualized fog platforms for single-hop vehicular TCP/IP links. The objective of the work is to reduce energy consumption of fog nodes for resource-intensive and delay-sensitive Infrastructure-to-Vehicle (I2V) services. The scheduler performs admission control of the input traffic to be processed by the fog nodes, dispatching of the admitted traffic, adaptive reconfiguration and consolidation of the virtual machines hosted by the fog nodes, and traffic control. System performance tests were conducted to test the solution's capabilities in terms of processing delay, rate jitter, and traffic rate under different scenarios that consider node mobility, wireless fading and resource costs.

In [15], the authors designed a distributed and adaptive cognitive resource management controller for vehicular access networks. Energy and computing-limited car smartphones are able to utilize the available V2I Wi-Fi connections and perform traffic offloading towards local and remote cloud datacenters. This process is achieved by ascending to a spectral-limited wireless backbone that is built up by multiple Roadside Units (RSUs). The resource management problem is modeled as a stochastic network utility maximization problem. As such, the controller dynamically allocates the access time-windows at the serving RSUs, as well as the access rates and traffic flows at the vehicular clients. Moreover, the solution adapts to mobility and fading-induced changes in the vehicular network. Lastly, it exploits cognitive radio support to maximize energy and bandwidth efficiency of the vehicular access network.

Yangui et al. [16] proposed a solution based on the Platform-as-a-Service (PaaS) architecture and the REpresentational State Transfer (REST) paradigm for application provisioning automation in hybrid fog/cloud environments. The enhanced architecture is composed of four layers: Application development, application deployment, application hosting and execution, and application management. The first layer allows for the composition of components and communication with IoT devices. The second layer consists of a module called the *Deployer* responsible for placing the applications at either the cloud or fog. A *controller* that interacts with the *deployer* sets the locations of the components. The third layer is responsible of orchestrating the execution flow between service containers of an application spanned across the cloud and fogs. It also coordinates the exchanges of messages. The last layer interacts with all other layers for the purposes of managing application QoS, migrating the components from the cloud to the fog and vice versa (or between different fogs). A validation scenario is given using a fire detection and fighting application. Temperature sensors and robots are used as fire detectors and fire fighters, respectively.

Resource management in the fog/cloud architecture is considered in [17]. A QoS-aware dynamic fog service provisioning framework is proposed to dynamically deploy and release new and old application services on fog nodes, respectively. Two algorithms are proposed, namely, Min-Viol and Min-Cost. The first aims at minimizing the delay violations, while the later minimizes the total cost. The Mini-Viol algorithm deploys on fog nodes services with high demands with respect to QoS requirements by continuously monitoring the traffic from different services. On the contrary, the Min-Cost algorithm checks whether deploying or releasing services will increase or decrease revenue and cost, respectively. The deployment and releasing process is achieved according to the incoming traffic rates to the fog nodes. Simulation results showed that delay and overall cost is reduced but at the cost of slower runtimes.

The authors in [18] presented a distributed fog architecture that is hierarchal in nature to support big numbers of infrastructure entities and services for smart cities. The idea behind their work is to deploy large number of sensors through large-scale geospatial sensing networks serviced by different fogs to perform big data analysis, identify anomalous and hazardous events, then offer optimal and on-time responses. Their architecture is composed of four layers. The top layer consists of the cloud and serves very high latency computing tasks that would take days to years to analyze and process. The second layer through the fourth consists of different fog entities for faster service responsiveness. For instance, the second layer contains intermediate computing nodes used to make quick responses to control the infrastructure whenever hazardous events occur. Computing nodes at the second layer are connected to the cloud data centers in layer 1 on one hand, and to edge computing nodes in layer 3 on the other. Edge nodes are responsible for a group of sensors from the first layer. The edge nodes have high performance characteristics with low-power consumption. The first layer manages the sensing networks dispatched on different smart city infrastructures. The authors constructed a working prototype for pipeline monitoring to test the proposed cloud solution with different layers of the architecture.

Yan et al. [19] proposed a fog-based data storage and processing solution for improving the smart meter infrastructure deployed in some cities. To reduce data transmission to the centralized cloud entity, certain smart meters are selected as fog nodes, such that smart meters are grouped together, forming clusters. One of the smart meters in each cluster is selected for data storage in regards to the collected information from each meter. The collected data from each fog node is then stored onto the cloud for backup purposes only. Modules on the fog nodes are used to duplicate and split the data to be distrusted onto different smart meters. The purpose of such a solution is to offload data processing from the centralized cloud to distributed fog nodes.

To the best of the authors' knowledge, the work illustrated in this article is the first to envision a node willingness-to-cooperate technique that uses three criteria to classify the IoT devices' abilities to cooperate for the purpose of service composition and delivery. The objective of the work is to distribute the work among different nodes at the IoT layer and reduce both fog and cloud workload. The proposed technique allows for most smart city services to be delivered independently from the cloud for more time-sensitive applications. The solution provides a more secure communication to the fog and cloud by integrating intrusion detection techniques.

III. Problem and Solution Overview

Smart cities have benefited from cloud computing through its deviant processing, storage and networking capabilities. By distributing a plethora of sensing devices in smart cities, for different sectors such as traffic, health-care, environmental and governmental, data collected is sent to the cloud for processing and analysis. Traffic congestions in certain areas are eliminated, health-care services have become more available and user friendly, and environmental pollutions have been reduced for some cases. Although such a solution that heavily relied on a centralized entity proved to be successful at one point, with today's wide variety of smart city applications and the overabundant numbers in resource-rich sensors and mobile devices, traditional cloud solutions are no longer ideal choices. Many of today's smart city applications are time-sensitive and cannot tolerate large delays, especially in life-threatening situations such as remote surgeries or building gas leaks.

As fog computing was introduced to offload cloud tasks, the concept of Fog-to-Cloud (F2C) communication became a widely accepted architecture for smart cities [20]. Certain user requests received at the cloud are offloaded to the fog when resources are not available to assure continuous and timely service delivery. During the early stages of fog computing, nodes did not communicate directly with the fog. But with the introduction of MEC, service requests were sent directly to the fog for faster service responsiveness through the edge devices. In this article, we refer to such communication as node-to-fog (N2F). As the fog computing strategy became widely accepted, the idea of having fog nodes collaborate for resource sharing through a more distributed service provisioning solution became a practical solution through fog-to-fog (F2F) communication [21]. At first, whenever service requests received at a certain fog are determined to be not achievable due to the limited resources of the fog, such requests are shared among the neighbouring fogs to identify the fog capable of fulfilling the request. If none of the fogs are able to fulfill the request, then the request is forwarded to the cloud. Although such a solution provided QoE-enabled service delivery for certain requests, load balancing and resource sharing among cooperating fogs is not achieved, thus leading to high latencies whenever a fog is not found to deliver the service. To solve this issue, a cooperative approach that shares part of each fog's resources to compose and deliver simple and complex cloud services was proposed in [22]. Incentives in the form of profit maximization for cooperating fogs is achieved leading to not only adequate times for service delivery, but also adherence to agreed upon QoS.

With the advances in node-to-node communication using Ad Hoc networks such as Mobile Ad Hoc Networks (MANET) [23] and Vehicular Ad Hoc Networks (VANET) [24], in addition to data replication strategies [25], self-reliant distributed service composition and delivery techniques have become a promising solution for the ever-growing urban smart cities. Figure 1 provides an overview of the entity interactions at different layers of a smart city environment. It is interesting to note that public vehicular transportation systems, such as trains and buses may act as both fogs with F2C and F2F communication capabilities and/or vehicular nodes with node-to-node communication capabilities.

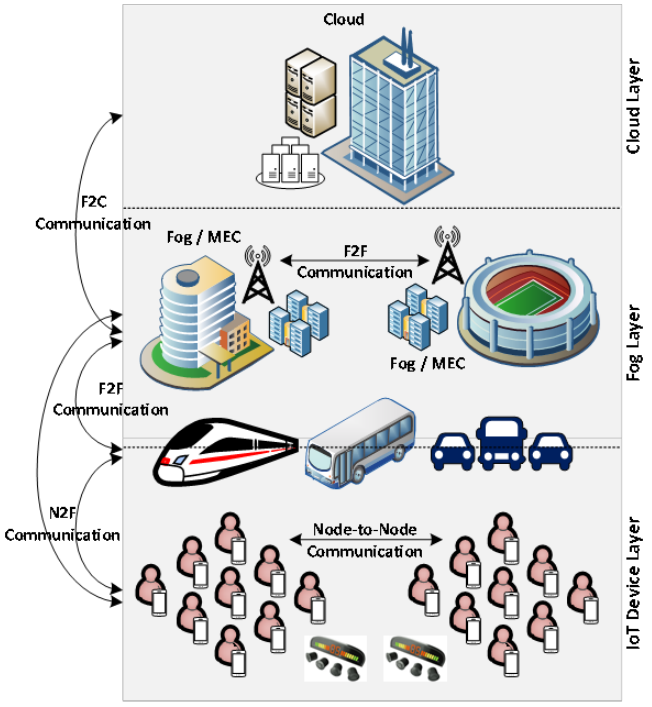

Fig.1. Types of communication involved in a typical smart city scenario. A three-layer structure is adopted, namely, the cloud layer, the fog layer, and the IoT device layer. The communication involved between the layers are: node-to-node communication, node-to-fog communication (N2F), fog-to-fog communication (F2F), and fog-to-cloud communication (F2C).

In this article an architecture is proposed and is integrated within the three layers of the smart city communication paradigm as shown in Figure 2. At the cloud layer, three modules are available, namely, 1) replica management, responsible of creating and maintaining replicas, in addition to selecting replica nodes, 2) service composition, responsible of composing both simple and complex cloud services, and 3) intrusion classification, used to avoid different types of attacks at the cloud data center and storage sites. At the fog layer, three modules are available, namely, 1) cache management, use for selecting mobile cache nodes from the IoT layer, as well as creating and maintain data for each cache, 2) Trusted-Third Party (TTP) service mediators, which are service negotiating entities responsible for selecting the most optimal set of fog resources according to user QoE preferences, and 3) intrusion detection, used to avoid different types of attacks at fog processing and storage nodes. At the IoT layer, four modules are integrated within each IoT device, namely, 1) service

discovery, for discovering neighbouring node capabilities, 2) service selection, for selecting nodes according to the needed resources at different stages of a service composition workflow, 3) service composition, the algorithm used for composing both simple and complex service, and 4) Node characteristics, which describes each IoT device's characteristics according to the three criteria to determine a node's willingness and its ability for cooperation to compose simple and complex services.

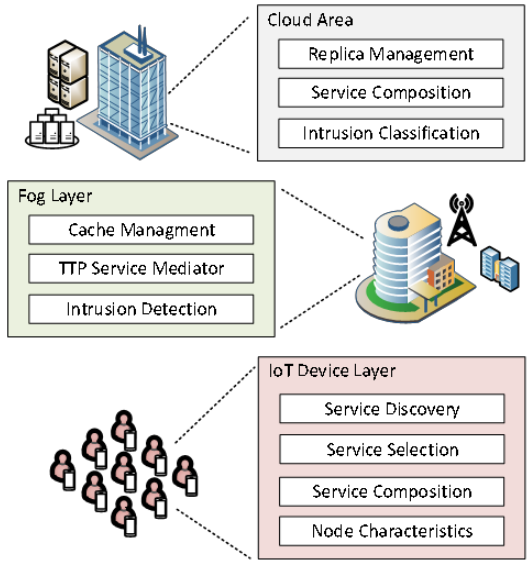

Fig.2. Smart city service provisioning architecture with three different layers: Cloud, Fog, and IoT Devices.

The work discussed in this article is built upon the work in [22] and [26], and will focus mainly on describing the node's willingness-to-cooperate characteristics, the service composition process which is dependent on the aforementioned node characteristics, and intrusion detection at the fog and cloud layers. For details in regards to replica and cache management, the reader may refer to our previous work in [27]-[30]. When services are requested from service mediators (i.e. TTPs), a QoE game theory is established. The game theory assumes that service requesters and providers play in accordance to their best interest, such that providers aim for maximized service costs and requesters aim for reduced service costs and higher service quality. TTP service mediation considers nodes' capabilities and characteristics when identifying and selecting fog resources (i.e. services) for requesters. TTPs are rewarded for their negotiation efforts in terms of the number of rounds taken to determine the optimal choice of service. For more details in regards to the TTP service mediation process, the reader may refer to [31]-[33]. Finally, for service discovery and selection, the reader may refer to [34]-[36].

IV. IoT Device Capbilites and Characteristics

IoT device cooperation for simple and complex service delivery and resource sharing can be accomplished optimally when nodes' characteristics and capabilities are known in advance by neighbouring nodes. Characteristics are determined using the three-criteria node description, while capabilities are determined using ontologies [37]. A node's cooperative score is hence determined as follows:

$$C_{MN_i} = \omega_{cap} \times CP_{MN_i} + \omega_{char} \times CR_{MN_i} \quad (1)$$

where CP_{MN_i} and CR_{MN_i} are the capability and characteristic scores for a node respectively. ω_{cap} and ω_{char} are the weights assigned for capability and characteristic scores respectively, to define the relative importance of each on the overall function. Nodes with high C_{MN_i} values indicate their willingness and capability of performing all or part of the requested service. The cooperative score is stored at the TTP and shared among other fog nodes and IoT devices.

Using the cooperative function defined in (1), highly cooperative nodes capable of providing a requested service are selected to create composition paths (i.e. cooperative nodes connected together directly or indirectly) to deliver composed services. The selection of the most optimal path is dependent on a learning method described in Section V. Given the node capabilities and characteristics, the service discovery, selection and composition process becomes more reliable and resistant to suspicious nodes.

A. Node Capabilites

Nodes' hardware and software capabilities are the main two service types for most IoT devices. Whenever a request for service is received at an IoT device, determining the node's capability is achieved by comparing the syntactic and semantic similarity of the request to the devices capabilities as described in the ontology. For instance, a request for the addition of an audio effect to a multimedia file as part of the service composition process would require that the ontology description file for the device *willing* to cooperate have its ontology description properties compared in terms of semantic similarity. Semantic similarity compares the property descriptions of the cooperative node and the requested service description to determine the number of matching features. Semantic distance is also used to determine whether disjoint properties (i.e. non-matching properties) are somewhat similar, by measuring the distance between the ontology classes of the requested object properties. An example of an ontology structure is given in Figure 3. Details in regards to ontology syntax, semantics and the mathematical functions used for measuring semantic similarity and distance can all be found in [38] and is out of the scope of this article.

A node's capability score is derived by comparing ontological property descriptions of node MN_i and service request SR_j as follows:

$$CP_{MN_i} = \frac{\left|F_{MN_i} \cap F_{SR_j}\right|}{\left|F_{MN_i} \cap F_{SR_j}\right| + \omega_{F_{MN} \setminus F_{SR}} \left|F_{MN_i} \setminus F_{SR_j}\right| + \omega_{F_{SR} \setminus F_{MN}} \left|F_{SR_j} \setminus F_{MN_i}\right|} \quad (2)$$

where F_{MN_i} and F_{SR_j} correspond to the ontology features for the service being requested, $F_{MN_i} \cap F_{SR_j}$ indicates the set of matching ontology features, $F_{MN_i} \setminus F_{SR_j}$ is the set of features that were found in MN_i but not in the requested service, in addition to its overall effect on the capability score $\omega_{F_{MN} \setminus F_{SR}}$, and $F_{SR_j} \setminus F_{MN_i}$ is the set of features that were not found in MN_i but are needed in the requested service, in addition to its overall

effect on the capability score $\omega_{F_{SR} \setminus F_{MN}}$. A capability score of one indicates that the nodes is fully capable of providing the requested service according to all described features. On the contrary, a score of zero indicates that there is no match and that the node is incapable of performing the service without a single matching service feature.

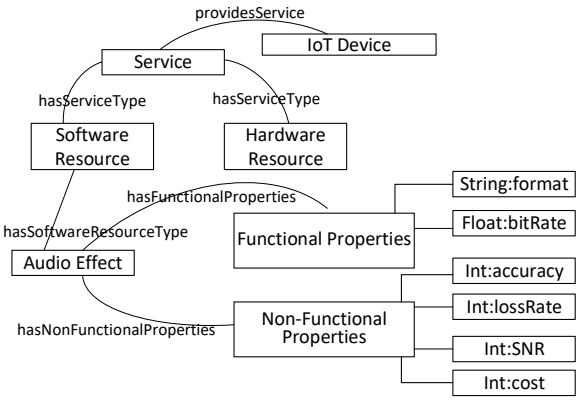

Fig.3. An example of different ontology classes and object properties used to describe an IoT device's hardware and software resources.

B. Node Characteristics

A node's willingness to cooperate or collaborate with other nodes in a smart city environment is determined according to certain node characteristics. Since IoT devices (e.g. smart phones) belong to end-users, which in reality are human beings, devices tend to carry over similar characteristics of the user. People in reality differ in their characteristics, for example, one person might be open to others, while another is more conservative. A person's willingness to share something might be motivated by their nature. Such human characteristics is the driving force behind the proposed cooperative solution. By applying three personality criteria to IoT devices, namely, *awareness*, *attitude* and *nature*, a new technique is developed to determine the *willingness* (i.e. probability) of a node to join other nodes in order to compose and deliver smart city services without reliance on the cloud. Figure 4 provides a visualization of the three criteria and the mutual relationship between each other.

Given the three characteristics criteria, the overall characteristics score for a node is calculated according to (3) and is normalized to be in the range $0 \leq CR_{MN_i} \leq 1$, where 0 is a node characterized as unwilling to cooperate, while 1 is a node characterized to be highly willing to cooperate.

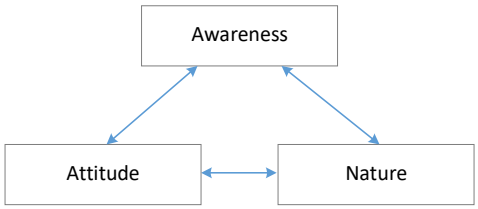

Fig.4. The three node characteristics criteria and the inter-relationship between each other.

$$CR_{MN_i} = \omega_{aware} \times aware\_rate_{MN_i} + \omega_{attitude} \times attitude\_rate_{MN_i} + \omega_{nature} \times nature\_rate_{MN_i} \quad (3)$$

A node's characteristics is defined as follows:

$$\mathbb{C} = \langle \Psi, \Omega, \Phi \rangle \quad (4)$$

where Ψ is the node's awareness, Ω is attitude, and Φ is nature. Each criterion has a short- and long-term impact on a node's cooperative sate, and is dependent on the rest of the criteria. Moreover, each criterion is classified into different weighted behavior classes using fuzzy logic. The use of fuzzy logic has been used in many different disciplines to manage and model imprecise concepts when definitive values cannot be determined. A fuzzy system can dynamically adjust and manage a node's characteristics descriptive class during the node's ongoing operation.

The fuzzification function adopted in this article uses a generalized bell-shape membership function due to its adjustability and capability to fit the behavior of node characteristics [39]. The membership function is defined according to (5), where a, b, and c are characteristics configuration parameters that are adjusted to fit the desired membership data. Moreover, c represents the exact desired value for that fuzzified region, a represents the fuzzified region's range of values, and b reflects the slope of the curve.

$$f(x; a, b, c) = \begin{cases} 0 & for\ x < a \\ \dfrac{2(x-a)^2}{(c-a)^2} & for\ a \leq x < b \\ 1 - \dfrac{2(x-c)^2}{(c-a)^2} & for\ b \leq x \leq c \\ 1 & for\ x > c \end{cases} \quad (5)$$

Figure 5 provides a generalized illustration of the fuzzification membership function.

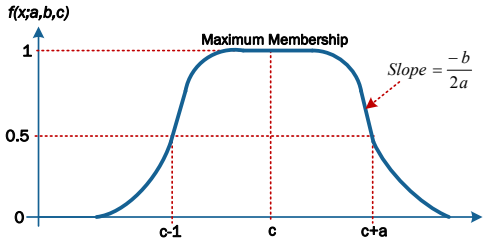

Fig.5. Illustration of the bell-shaped fuzzification membership curve.

i) Awareness

Awareness Ψ is defined as the node's ability to evaluate different situations (i.e. network behavior and surrounding contextual state), then as a result take an action. For example, if a node's cognitive state determines that certain nodes in the surrounding environment are malicious, then the node informs the fog of the current state for actions to be taken. We rely on the definition of awareness described in [40] to divide awareness levels into five different states, namely, $\Psi = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$:

- **Primitive Awareness (α_1)**: nodes of this state have nearly no awareness of the surrounding environment.

Nodes cannot see beyond the problem, such that no solutions can be identified.

- **Binary Awareness (α_2)**: nodes can identify two solutions to every problem, such as yes or no, cooperative or non-cooperative, right or left, etc.
- **Divergent Awareness (α_3)**: nodes of this state can provide an advanced solution. Such a solution is not simple or binary but rather provides more intelligent features.
- **Unconditioned Awareness (α_4)**: nodes can provide multiple advanced solutions to a given problem.
- **Transcended Awareness (α_5)**: nodes provide evolving solutions through reinforced learning methods. Such solutions advance with time and experience.

Using the five different states of awareness, Figure 6 illustrates the fuzzified membership function for awareness. Parameter c, which identifies the exact desired value, is dependent upon the fuzzified regions, identified in the figure. Parameter a, determines the width of the membership curve and is represented as the range of values that fall within the specified region, such that $aware\_rate_{min} < aware\_rate < aware\_rate_{max}$. For instance, assuming a service request which requires cooperation tolerates a node awareness rate of about 3.2 (i.e. divergent awareness), then the midpoint c would be set directly at that toleration level, with an upper and lower end points for an acceptable awareness rate, namely, $aware\_rate_{min}$ and $aware\_rate_{max}$.

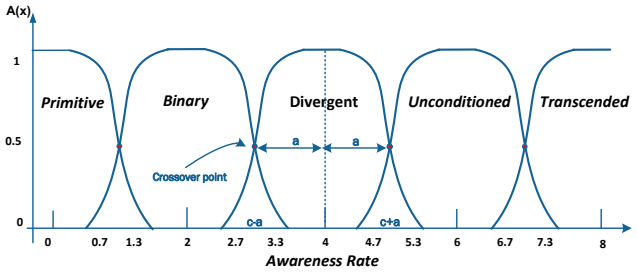

Fig.6. Illustration of the fuzzification membership function used for awareness.

The awareness rate is dependent on different factors, including the other two node characteristics criteria, namely, attitude and nature. For instance, a node described as aggressive (i.e. aggressive nature) would have a low awareness rate due to the minimal interaction with other nearby cooperative nodes. Moreover, as the user's schedule continues to be busy over time, the awareness rate would also decrease due to the node's unwillingness to cooperate with other nodes. The main factors affecting a node's awareness rate are: hardware capabilities, software capabilities, wired and wireless network connections, and training history. Each factor has a weight that affects the overall node's awareness rate according to (6).

$$aware\_rate_{MN_i} = \omega_{HW} \times HW_{MN_i} + \omega_{SW} \times SW_{MN_i} + \omega_{NW} \times NW_{MN_i} + \omega_{TR} \times TR_{MN_i} + \omega_{attitude} \times attitude_{MN_i} + \omega_{nature} \times nature_{MN_i} \quad (6)$$

The value of $aware\_rate_{MN_i}$ is normalized to be in the range $0 \leq aware\_rate_{MN_i} \leq 1$, where 0 is a node characterized as purely primitive, while 1 is a node characterized to be highly transcended. For instance, nodes with high hardware and software computing power, having a variety of network connections (e.g. Wi-Fi, Bluetooth, etc.), and having an exhaustive training history, would result in a high awareness rate. Having a high awareness rate would result in a higher node characteristic score in accordance to the fuzzification function, resulting in a higher probability of cooperation.

ii) Attitude

Attitude Ω is the way in which a node tends to act at different states. Such actions are derived from the current personal and surrounding state and are highly dependent on the other four criteria. For instance, a node that has very limited resources at a certain time point, tends to disconnect from all device-to-device connections to free up certain resources, and hence would have an affect of its cooperative state. On the contrary, if the user had a busy work schedule, then according to the gathered history, it is determined that the node's resources are free of use and it is highly probable that the node is willing to cooperate or share its resources due to having the user rest for a certain time according to his schedule. According to [41], human emotions or attitude are classified into four categories, namely: happy, sad, fear, and anger. Using such characteristics, we adapt the four categories $\Omega = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ to characterize node attitude which is mainly derived from human emotions:

- **Enraged Attitude (β_1)**: nodes of this state are highly non-cooperative and non-trustworthy towards other nodes of unknown characteristics. Such behaviour may lead to a failure in service delivery if chosen for cooperation.
- **Distressed Attitude (β_2)**: nodes of this state have experienced a set of negative outcomes leading to unsatisfaction. The probability of node cooperation under this state is considered low. For example, not receiving the requested services according to the agreed upon Service Level Agreement (SLA) leads to node distress. This state is also dependent on current and previous experiences and environmental settings.
- **Concerned Attitude (β_3)**: nodes of this state possess fear towards initiating solutions that involve cooperation or trust. Such state is a result of previous cooperation results leading to negative outcomes (e.g. security concerns, incentive concerns, etc.).
- **Satisfied Attitude (β_4)**: nodes of this state have experienced a set of positive outcomes leading to satisfaction. Such nodes are more likely to cooperate and can be considered more trustworthy. For instance, a node that received requested services according to the agreed upon SLA is considered satisfied. This state is dependent upon current and previous experiences and environmental settings.

Figure 7 provides an overview of the fuzzy logic membership function used for the attitude node characteristic. A node's attitude state is dependent on different factors, namely: network status, cooperative experience, and SLA

abidance. Moreover, attitude is also highly dependent on the awareness and node characteristics. Therefore, the overall node's attitude score is derived as follows:

$$attitude\_rate_{MN_i} = \omega_{NW} \times NW_{MN_i} + \omega_{CP} \times CP_{MN_i} + \omega_{SLA} \times SLA_{MN_i} + \omega_{aware} \times aware_{MN_i} + \omega_{nature} \times nature_{MN_i} \quad (7)$$

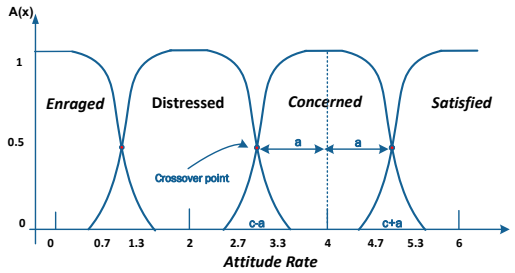

Fig.7. Illustration of the fuzzification membership function used for attitude.

The value of $attitude\_rate_{MN_i}$ is normalized to be in the range $0 \leq attitude\_rate_{MN_i} \leq 1$, where 0 is a node characterized as highly enraged, while 1 is a node characterized to be highly satisfied. For example, nodes experiencing a network described as safe with high throughput and low delay experience, in addition to successful cooperation results and SLA abidance, would result in a high attitude rate. As stated earlier, the nodes awareness and nature characteristics also play an important role on the overall attitude score. The weight of each factor is determined in accordance to the type of service request. High attitude scores result in a higher node characteristic score.

iii) Nature

Nature Φ is highly related to a user's personal characteristics. For instance, a person who is highly aggressive in nature would result also in having the node act aggressively. Such aggressive nature might be in terms of non-cooperation, malicious attacks, and many other aggressive behaviours. The tendency of a node to communicate with other nodes can also be determined based on the nature of the node. Such characteristic is not only dependent on the other four criteria, but is also dependent on the node's capability. A node that does not have device-to-device communication capabilities, will not be able to join compositions in a direct connection, but rather would require data to be transmitted to the base station and then routed to other nodes which may cause increased latency. We categorize a node's nature characteristic into four categories $\Phi = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$: cooperative, peaceful, non-cooperative, and aggressive.

- **Aggressive Nature (γ_1)**: nodes of this state are considered non-cooperative. Moreover, such nodes pose security and privacy threats towards the network.
- **Non-Cooperative Nature (γ_2)**: nodes of this state are considered rarely cooperative or non-cooperative. Although such nodes are considered non-cooperative, they do not pose any security nor privacy issues towards other nodes.
- **Peaceful Nature (γ_3)**: nodes of this state are considered somewhat cooperative depending on different factors such as user schedule and network settings. Similarly, nodes of this state do not pose any security and privacy threats.
- **Cooperative Nature (γ_4)**: nodes of this state are considered the most willing to cooperate and do not pose security nor privacy threats towards other nearby and cooperating nodes.

A node's nature characteristic is determined using different factors such as network ranking (i.e. a rank that is given by a trusted third party which uses service negotiations ratings to determine a node's behavior [33]), hardware and software capabilities as defined in (8).

$$nature\_rate_{MN_i} = \omega_{TTP} \times TTP_{MN_i} + \omega_{HW} \times HW_{MN_i} + \omega_{SW} \times SW_{MN_i} + \omega_{aware} \times aware_{MN_i} + \omega_{attitude} \times attitude_{MN_i} \quad (8)$$

The state of the node is determined using a fuzzy membership function similar to that of awareness and attitude and is depicted in Figure 8. The value of $nature\_rate_{MN_i}$ is normalized to be in the range $0 \leq nature\_rate_{MN_i} \leq 1$, where 0 is a node characterized as highly aggressive, while 1 is a node characterized to be purely cooperative.

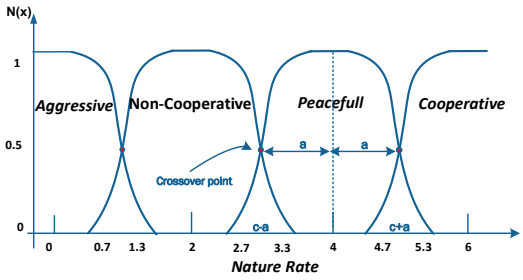

Fig.8. Illustration of the fuzzification membership function used for nature.

V. Service Composition

Composite services are a result of a cooperative process between different entities in the environment. Most composite services require different nodes to collaborate and share part of their resources and capabilities (i.e. hardware/software). Composition may occur on three different layers: using mobile nodes only, using a mix of mobile and fog nodes, and using fog nodes only. The choice of the composition process is dependent on the type of service requested and the service requirements. Service requests that are determined to be simple and time-sensitive are composed and delivered through the cooperation of fog nodes only. For instance, cloud services that have been decomposed and replicated to fog and edge sites will require multiple fogs to cooperate and recompose the service whenever requested [26]. Such compositions are referred to as simple composite service requests. The selection of the fog nodes is dependent on the sub-service (i.e. service unit) availability and service delivery requirements. Such a technique is usually simple and may not require optimization unless cost of performing the service and profit maximization are mandatory. This problem is out of the scope of this article and is being investigated by the authors in another work.

In this article we focus on complex service requests that involve the use of mobile nodes in the smart city environment. Requests that cannot be composed due to it not being available

at the cloud nor fog and edge devices (at least partially) are considered complex and requires the use of the capability and characteristics mechanism introduced in the previous section to identify capable and willing nodes to be used for the composition process. Since the process of determining node willingness and capability using the fuzzification function may be time-consuming, a learning technique is adopted to select composition patterns that have proved to provide adequate results in terms of service quality and time-constraints. Those composition patterns are constructed using nodes with high cooperative scores determined with the aid of the TTPs.

Using the nodes' cooperative scores defined in (1), a reward matrix is derived for a service request as shown in the example below, where each element in the matrix indicates the current reward for executing a service s_j according to the cooperative score C_{MN_i}.

$$\mathbb{R}_S = \begin{array}{c} \\ MN_1 \\ MN_2 \\ MN_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} \mathbb{R}_s(C_{MN_1}(s_1(t))) & \mathbb{R}_s(C_{MN_1}(s_2(t))) & \mathbb{R}_s(C_{MN_1}(s_3(t))) \\ \mathbb{R}_s(C_{MN_2}(s_1(t))) & \mathbb{R}_s(C_{MN_2}(s_2(t))) & \mathbb{R}_s(C_{MN_2}(s_3(t))) \\ \mathbb{R}_s(C_{MN_3}(s_1(t))) & \mathbb{R}_s(C_{MN_3}(s_2(t))) & \mathbb{R}_s(C_{MN_3}(s_3(t))) \end{bmatrix} \end{array}$$

The reward $\mathbb{R}_s(C_{MN_i}(s_j(t)))$ for executing service unit $s_j \in S$ according to the capability score C_{MN_i} at time t is dependent on a history record of executions. Accordingly, the average error is calculated as follows:

$$E = \left(\int_{t=1}^{T} \frac{(\mathbb{R}_{expected}(C_{MN_i}(s_j(t))) - \mathbb{R}_{actual}(C_{MN_i}(s_j(t))))^2}{2} \partial t \right) / T \quad (9)$$

where T is the length of the history record, $\mathbb{R}_{expected}(C_{MN_i}(s_j(t)))$ is the expected reward at time point t, and $\mathbb{R}_{actual}(C_{MN_i}(s_j(t)))$ is the actual reward at time point t.

Given that the reward range is $\mathbb{R}_s\left(C_{MN_i}\left(s_j(t)\right)\right) - E \leq \mathbb{R}_{actual}(C_{MN_i}(s_j(t)))) \leq \mathbb{R}_s\left(C_{MN_i}\left(s_j(t)\right)\right) + E$, the value function for performing s_j at time t using mobile node MN_i according to its capability score C_{MN_i} is:

$$V_{C_{MN_i}(s_j(t))} = V_{C_{MN_i}(s_j(t-1))} + \varphi\left(V_{C_{MN_i}(s_j)} - V_{C_{MN_i}(s_j(t-1))}\right) \quad (10)$$

where $V_{C_{MN_i}(s_j(t-1))}$ is the value function at the previous time point, $V_{C_{MN_i}(s_j)} = \mathbb{R}_{actual}(C_{MN_i}(s_j(t))))$, and φ is the learning rate.

Accordingly, the composition pattern is selected such that the higher reward values are achieved using (10). The pattern is simply the set of mobile nodes used leading to max $(V_{C_{MN_i}(s_j)})$. Below is an example of a pattern selected according to the previous matrix example, where using the resources and capabilities of mobile node MN_2 to fulfil service unit s_1, MN_1 for service unit s_1, and MN_3 for service unit s_3, to compose the overall requested service S, leads to the highest reward value.

$$\mathbb{R}_S = \begin{array}{c} \\ MN_1 \\ MN_2 \\ MN_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} \mathbb{R}_s(C_{MN_1}(s_1(t))) & \mathbb{R}_s(\boldsymbol{C_{MN_1}(s_2(t))}) & \mathbb{R}_s(C_{MN_1}(s_3(t))) \\ \mathbb{R}_s(\boldsymbol{C_{MN_2}(s_1(t))}) & \mathbb{R}_s(C_{MN_2}(s_2(t))) & \mathbb{R}_s(C_{MN_2}(s_3(t))) \\ \mathbb{R}_s(C_{MN_3}(s_1(t))) & \mathbb{R}_s(C_{MN_3}(s_2(t))) & \mathbb{R}_s(\boldsymbol{C_{MN_3}(s_3(t))}) \end{bmatrix} \end{array}$$

This technique is adaptable to both cooperation among mobile nodes only and a mix of mobile nodes and fog nodes. Whenever fog nodes are used, the capability score and reward function are used only to identify the mobile nodes needed for services not available at the fog nearby the service requesting node. On the contrary, when mobile nodes are only used for the cooperation, then the cooperation strategy is purely based on the use of node capability score and reward functions.

VI. Intrusion Prevention and Detection

Many researchers have been working to solve the intrusion detection issue in different networks and fields by using supervised learning, unsupervised learning, and hybrid learning mechanisms [42]-[44][47][48]. A novel classifier-ensemble for intrusion detection by adopting Particle swarm optimization (PSO) weights has been proposed in [49]. Furthermore, the authors in [50] proposed a new hybrid detection system by adopting the Support Vector Machine (SVM) and Radial Basis Function (RBF). Their work proved the effectiveness of heterogeneous models in comparison to homogeneous solutions. In [51], the authors presented a hybrid approach that uses resampling to maximize the flow in a minority class. It also applies the ensemble method to improve the classifier generalization. Moreover, the authors in [52] proposed an ensemble intrusion detection mechanism to mitigate malicious behaviors. Their AdaBoost-based ensemble algorithm was built using three machine learning techniques, namely, Decision Tree (DT), Artificial Neural Network (ANN), and Naive Bayes (NB).

In our model, we split the proposed ensemble-based intrusion detection system (E-IDS) into two layers, namely, intrusion detection at the fog and intrusion classification at the cloud. At the fog layer, computing resources are limited, hence, intrusion detection is a binary procedure and has two outcomes, either normal or abnormal. Therefore, intrusion detection does not require complex computation resources. Moreover, intrusion detection is considered critical, such a task requires low latency, making the fog layer an appropriate environment for its deployment. Intrusion classification on the other hand is considered a complex model where multi-class classification tasks are performed. The cloud layer is the optimal solution for such complex computations. In summary, the proposed E-IDS is deployed in the fog layer as the first classifier. Once an intrusion is identified, an alert is deployed. Then, the collected traffic will be directed to the cloud to perform the second task (i.e. the attacks classification task). Attacks classification is conducted using the proposed ensemble technique as well.

A. Proposed Ensemble Method

The proposed ensemble technique uses three base classifiers, namely, Support Vector Machine (SVM), Deep Belief Network (DBN), and eXtreme Gradient Boost (XGBoost). All the selected classifier techniques are capable of providing a decision boundary of the collected traffic instead of assigning the collected traffic to normal or malicious categories. The NSL-KDD dataset has been adopted for training and testing. In order to maximize the ensemble diversity, we split the NSL-KDD dataset that contains 41 features into two partial feature subsets. Such that, the basic and content features totaling 22 features represent the fist subset and

the 19 traffic features represent the second subset. The features undergo two parallel base classifiers, such that each contains three classifiers. All the collected decisions from the classifiers are combined using combiner 1 and combiner 2 as shown in Figure 9. The results of both combiners are integrated to produce the overall conclusion of the ensemble technique. The majority voting rule method is adopted as the fusion technique.

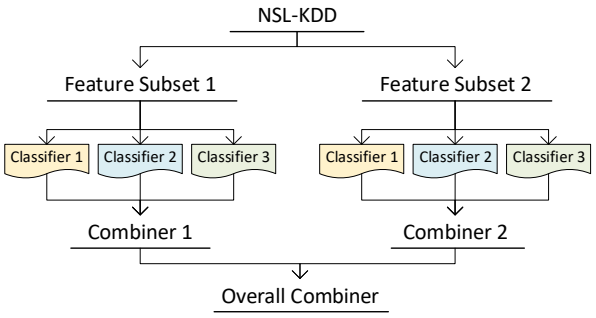

Fig.9. The proposed ensemble model.

i) Support Vector Machine

SVM is a supervised machine learning algorithm specialized in pattern recognition which is among the best supervised learning algorithm techniques [53]. SVM performs classification using a linear of non-linear separating surface that depends on a subset of the whole dataset. This chosen subset is the needed data to formulate the separating surface which as a result generates the support vector set. In non-linear SVM, the input vector is mapped into a high dimensional feature space, such that SVM generates a linear boundary in between various classes and maximizes the margin by adjusting the generated boundary [54]. It also maximizes the classification by sub-dividing the feature space into sub-spaces. On the contrary, in case of linear separated data, SVM aims to find the suitable vector between hyper planes. Then, an adjustment for boundary is performed by maximizing the distance between the nearest data points and the boundary.

In our proposed binary classification technique, the data points P are represented as shown in (11) [54]:

$$P = \{(j^1, k^1), (j^2, k^2), \dots, (j^n, k^n)\} \quad (11)$$

where $j \in D^m$ and $k \in \{-1, +1\}$, k refers to the binary value representing the classes, j refers to the input vector, D^m refers to the whole dataset and n is the number of training sets.

Using SVM, there will be numerous hyper-planes that separate the two data sets. The objective is to figure out the hyper-plane that achieves the maximum margin. All training data falls under some restrictions as shown below [54].

$$\begin{cases} \text{for } k^i = +1, j + b \geq +1 \\ \text{for } k^i = -1, j + b \leq -1 \end{cases}$$

where w refers to the boundary, j is the input vector, and b is the bias.

The used decision function for data classification is represented in (12), such that the disjoint hyper-plane must achieve the constraints represented in (13) [54].

$$f(k) = \pm\big((w, j) + b\big) \quad (12)$$

$$k^i[(w, j^i) + b] \geq 1 \quad (13)$$

ii) Deep Belief Network

DBN is a deep learning method which consists of numerous Restricted Boltzmann Machines (RBM). RBM is an energy model that consists of V visible nodes in the input layer and H hidden nodes in the hidden layers. v_i represents the i unit's state, and h_j represents the j unit's state. For a given state (V, H), the energy function is formulated as shown in (14) [55].

$$E = (V, H|w) = -\sum_{i=1}^{v} a_i v_i - \sum_{j=1}^{h} b_j h_j - \sum_{i=1}^{v}\sum_{j=1}^{h} v_i h_j W_{ij} \quad (14)$$

where $w = (W_{ij}, a_i, b_j)$ refers to RBM parameters, a_i refers to the visible bias, b_j refers to the hidden bias, and W_{ij} represents the $i - j$ weights.

The probability to every possible pair of a visible and a hidden layer is shown in (15) [55], where Z is the partition function that is shown in (16).

$$P(v, h) = \frac{1}{Z} e^{-E(v,h)} \quad (15)$$

$$Z = \sum_{v,h} e^{-E(v,h)} \quad (16)$$

From (15) and (16), the probability that the network assigns to a visible layer is shown in (17) [55].

$$P(v) = \frac{1}{Z} \sum_{h} e^{-E(v,h)} \quad (17)$$

iii) eXtreme Gradient Boost

To speed up the detection procedure, the XGBoost technique is used. XGBoost is considered a machine boosting technique and was designed for speed performance purposes using gradient-boosted decision trees [56]. XGBoost belongs to the Distributed Machine Learning Community (DMLC) which benefits from taking advantage of resources and memory for tree boosting algorithms as well as accomplishing main gradient mechanisms such as Regularized Boosting, Stochastic Boosting, and Gradient Boosting. XGBoost is considered an effective technique to reduce computation time. It applies the decision tree algorithm to the dataset and classifies the data.

The mathematical algorithm makes predictions p_i based on the trained data T_i. For example, in a linear model, the prediction is based on a combination of weighted input features such as $p'_i = \sum_j \theta_j p_{ij}$ [57]. Where θ refers to the parameters and p_{ij} refers to the predicted value used for classification purposes. XGBoost adds the predictions of all formed trees and optimizes the result. The objective function contains two parts: the first part represents the *regularization* (R) which helps in keeping the model complexity in the desired boundaries as well as eliminating the over-fitting of data, the second part represents the *training loss* (L) as shown in (18) [57].

$$Objective(\theta) = R(\theta) + L(\theta) \quad (18)$$

iv) The Combiner Method

In our model, we adopt the Simple Majority Voting (SMV) rule as the fusion technique to combine outputs together. The majority voting rule aims to direct the collected traffic to the majority class between the classifiers' outputs. Consider **d** is a set of D records in the adopted dataset and **c** is a set of C classes. An algorithm set $A = \{A_1, A_2, \dots, A_F\}$ is defined, which contains the F classifiers used for voting [58]. Each example $d \in \mathbf{d}$ is assigned to have one of the C classes. Each classifier will have its prediction for each example. The final class assigned to each example is the class predicted by the majority of classifiers (gaining the majority votes) for this example. This can be formulated as follows [58].

Let $c_l \in \mathbf{d}$ denote the class of an example x predicted by a classifier A_l, and let a counting function G_l be defined as:

$$\begin{cases} \text{if } c_l = c_k, G_k(c_l) = 1 \\ \text{if } c_l \neq c_k, G_k(c_l) = 0 \end{cases}$$

where c_l and c_k refer to the c classes. The class c_k total vote counts can be represented as in (19) [58].

$$V_k = \sum_{l=1}^{F} G_k(c_l) \tag{19}$$

The predicted class c for an example x using the algorithm set S is defined to be a class that gains the majority vote as shown in (20), where $k \in \{1, \dots, C\}$ [58].

$$c = A(x) = \max(V_k) \tag{20}$$

VII. Simulation Results

Simulations were conducted using NS-3 to test the performance of the proposed method against traditional fog-based solutions. Up to 100 mobile nodes are connected to fog nodes using an LTE network through enhanced NodeB (eNB) base stations. The coverage of each LTE network is 7 km with 30 dBm transmission power. Fog nodes are connected to the cloud using a number of gateways. Mobile nodes communicate directly with each other for cooperation purposes using WLAN access points with a data rate of 54 Mbps. Mobile node movement speed was set at 1-2 m/s, in which each is equipped with WLAN and LTE interfaces. Fog nodes communicate together through gateways. Table I summarizes the settings and configurations adapted in the simulator. The simulations conducted targeted testing the effectiveness of the cooperation and intrusion detection strategies. The simulation performance metrics used to test the proposed solution are: *i)* service hit ratio – defined as the success rate of discovering and retrieving the requested services, *ii)* service composition delay – defined as the time taken to compose a service from the time a service is requested, *iii)* the number of suspicious nodes identified given the trustworthiness criteria discussed in Section IV, and *iv)* the accuracy rate (AR), false negative rate (FNR), and detection rate (DR) of the proposed intrusion detection algorithm.

TABLE I. Simulation Settings and Configurations

Simulation parameter	Value
Simulator	NS-3
Operational area	1500 x 1500 meters
Density	5 – 100 mobile nodes
Mobility model	Random Way-point
Mobile node speed	1-2 m/s
Access points	LTE eNB (30 dBm), IEEE 802.11g (54 Mbps)
Message exchange frequency	10 ms
Packet size	1500 Bytes

A. Service Hit Ratio

Service hit ratio is one of the main advantages achieved through cooperation. As more nodes collaborate, the probability of composing the requested service increases. Figure 10 depicts the service hit ratio for three solutions, namely, the proposed mobile node and fog cooperative solution (CP-MN), the cooperative fog solution (CP-FG), and the traditional F2C solution (F2C). From the figure, it can be seen that as the number of cooperative nodes increases, the overall service hit ratio remains high. On the contrary, the other two solutions experience a sharp decrease in the service hit ratio as the number of nodes join the network. This is due to having more complex service requests that cannot be composed without the reliance on a plethora of hardware and software capabilities found using the cooperative mobile node and fog solution (i.e. CP-MN).

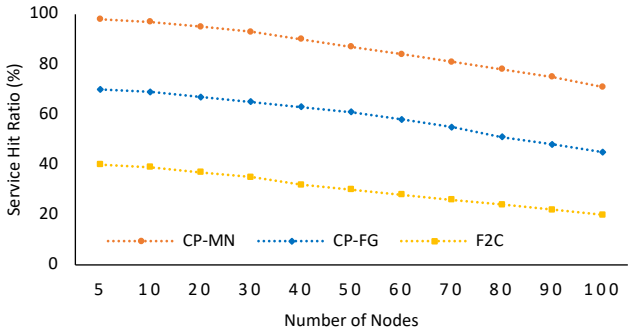

Fig.10. Overall service hit ratio for the cooperative solutions (CP-MN and CP-FG) and the traditional F2C solution.

B. Service Composition Delay

The time taken to compose a service as more nodes join the environment and request services was also considered in the simulation tests. Figure 11 depicts the results, highlighting the significance in delay reduction using CP-MN. As more nodes join the network, relying on the traditional F2C solution results in significantly high delays due to resource unavailability. The CP-FG solution provides a better alternative with reduced delays. Fogs are cooperative and thus can share resources whenever needed. The CP-MN solution on the other hand provides the best solution. By having multiple cooperative nodes share resources and capabilities, composite services can be delivered in a fraction of time when compared to the CP-FG and F2C solutions. We assume that all service units are available using the three solutions. Realistically speaking, certain service units might not be available using the traditional cloud solution.

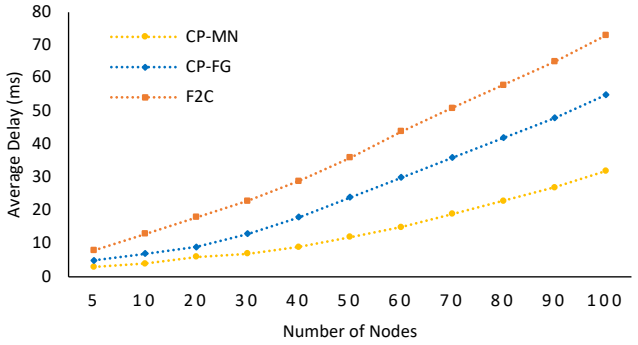

Fig.11. Experienced delay for service retrieval (composed services) using cooperative and non-cooperative solutions.

C. Suspicious Nodes

The adopted cooperative node characteristics solution provides high accuracy in terms of identifying potential suspicious nodes. Using the proposed three criteria node characteristics, a significant number of threats are identifiable. As seen from Figure 12, using the proposed CP-MN solution, the number of suspicious nodes identified is significantly higher than the traditional fog and cloud solutions. For instance, in a network environment of 100 mobile nodes, up to 20 nodes are identified as either suspicious or malicious (i.e. non-cooperative and may harm the cooperation process), as opposed

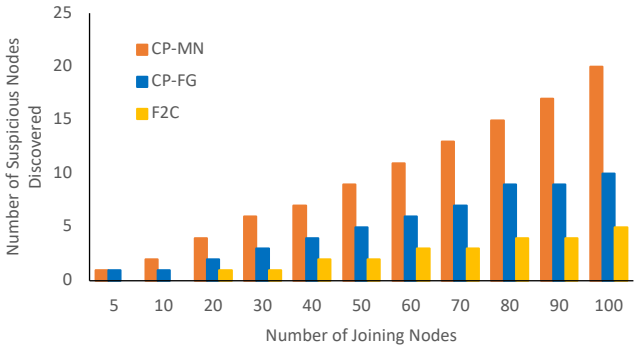

Fig.12. Identifying suspicious and malicious nodes using three different techniques.

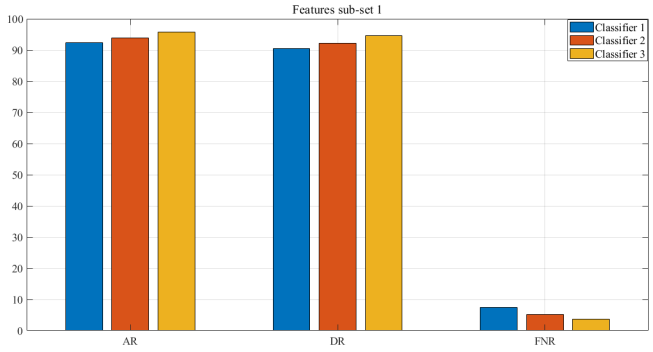

Fig.13. AR, DR, and FNR comparison between three classifiers in the first features sub-set.

to the cooperative fog solution (10 suspicious nodes) and the traditional F2C solutions (5 suspicious nodes).

D. Intrusion Detection

The Knowledge Discovery in Data mining CUP 1999 (KDDCup99) dataset is a subset of the Defense Advanced Research Projects Agency (DARPA) data-set [59]. The NSL-KDD dataset is an improved version of the KDD'99 dataset which was introduced to tackle KDD'99 issues such as omitting all redundant records in the training and testing datasets, and the number of records in training and testing datasets are reasonable compared to the ones in KDD'99 [60]. In the NSL-KDD dataset, each network connection contains a total of 41 features and each data record represents feature values of a class in the network data flow, where each class is labeled as either *attack* or *normal*. Attack types are classified into the following pre-defined groups of intrusive behavior: Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe [59]. The NSL-KDD dataset is used to test the proposed intrusion detection solution, such that each connection record contains 41 features and is labeled as normal or attack.

In each trial, we trained and tested the model. In the training phase, the three base classifiers, namely, SVM classifier, DBN classifier and, the XGBoost classifier, are built using the training dataset, namely, KDDTrain+. The testing dataset then undergo into each classifier in order to recognize malicious behaviors in the fog layer as well as to classify the intrusion types in the cloud layer. The solution is evaluated using standard intrusion detection measurements such as: accuracy rate (AR), false negative rate (FNR), and detection rate (DR).

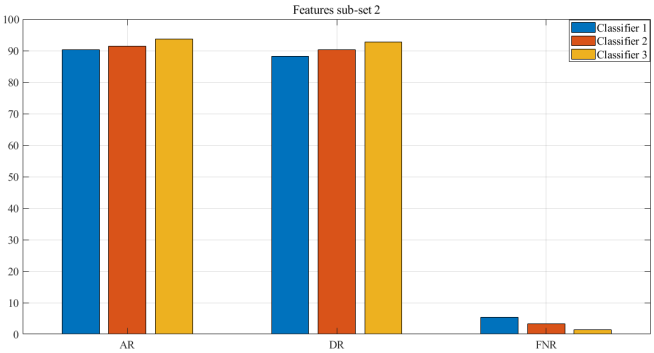

Fig.14. AR, DR, and FNR comparison between three classifiers in the second features sub-set.

Each run is performed for 10 trials and the average is recorded in order to mitigate the inaccuracy and variation factor. Figure 13 shows the AR, DR and FNR comparison between the three adopted classifiers (SVM, DBN and XGBoost) for the first NSL-KDD features subset. It is clear that XGBoost outperforms the other two classifiers. A similar comparison of AR, DR and FNR between the three classifiers for the second NSL-KDD features sub-set was conducted. Results are shown in Figure 14. Figure 15 represents the AR, DR, and FNR comparison between the combiners (SMV).

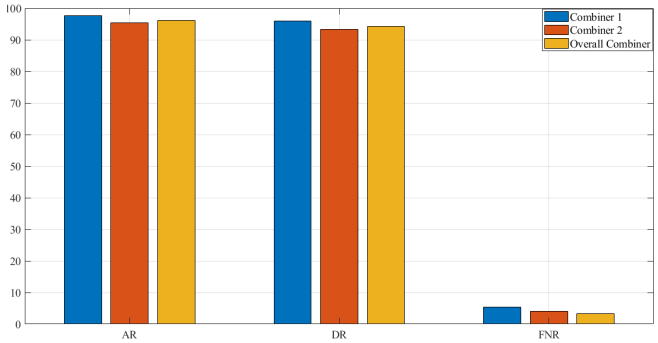

Fig.15. AR, DR, and FNR of the two combiners compared to the overall combiner.

VIII. Conclusion and Future Work

This paper introduced a solution to compose complex services with the aid of a fuzzification and reinforcement learning technique which depends on node "willingness" and "capability" characteristics. Node capabilities are determined through ontological property descriptions, while node willingness uses the node's level of intelligent awareness, attitude and nature towards task distribution and collaboration. Moreover, an ensemble-based intrusion detection system is developed that uses three base classifiers, namely, SVM, DBN, and XGBoost. Intrusion detection and classification are conducted separately at both the fog and cloud layers. Simulation tests conducted on the cooperative service composition solution against other cooperative and non-cooperative solutions revealed its superiority and adequacy in terms of efficient service composition and delivery latency, and intrusion detection. For future work, we plan to develop a decentralized service composition solution using block chain technology. The solution will not rely on service mediators nor intermediate network provider entities. Participants will be rewarded by cloud and fog entities for solving complex composition processes. Such a solution will reduce power usage of fog and cloud datacenters and provide even more diversified and sustainable smart city services.

References

[1] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[2] W. Tu, "Data-Driven QoS and QoE Management in Smart Cities: A Tutorial Study," in IEEE Communications Magazine, vol. 56, no. 12, pp. 126-133, December 2018.

[3] R. Bruno, M. Conti and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," in IEEE Communications Magazine, vol. 43, no. 3, pp. 123-131, March 2005.

[4] Y. A. Ridhawi and A. Karmouch, "Decentralized Plan-Free Semantic-Based Service Composition in Mobile Networks," in IEEE Transactions on Services Computing, vol. 8, no. 1, pp. 17-31, Jan.-Feb. 2015.

[5] I. Al Ridhawi, Y. Al Ridhawi, "QoS-Aware Service Composition in Cloud Mobile Networks", in Proc. 7th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 30 Nov. – 2 Dec. 2015.

[6] I. A. Ridhawi, M. Aloqaily and A. Boukerche, "Comparing Fog Solutions for Energy Efficiency in Wireless Networks: Challenges and Opportunities," in IEEE Wireless Communications, vol. 26, no. 6, pp. 80-86, December 2019.

[7] F. Al-Turjman, A. Malekloo, "Smart parking in IoT-enabled cities: A survey," Sustainable Cities and Society, vol. 49, pp. 101608, Aug. 2019.

[8] B. N. Silva, M. Khan, K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," Sustainable Cities and Society, vol. 38, pp. 697-713, Apr. 2018.

[9] H. Abbas, S. Shaheen, M. Elhoseny, A. K. Singh, M. Alkhambashi, "Systems thinking for developing sustainable complex smart cities based on self-regulated agent systems and fog computing," Sustainable Computing: Informatics and Systems, vol. 19, pp. 204-213, Sep. 2018.

[10] Y. Yao, X. Chang, J. Mišić and V. Mišić, "Reliable and Secure Vehicular Fog Service Provision," in IEEE Internet of Things Journal, vol. 6, no. 1, pp. 734-743, Feb. 2019.

[11] J. Li, X. Li, J. Yuan, R. Zhang and B. Fang, "Fog Computing-assisted Trustworthy Forwarding Scheme in Mobile Internet of Things," in IEEE Internet of Things Journal.

[12] B. Tang et al., "Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities," in IEEE Transactions on Industrial Informatics, vol. 13, no. 5, pp. 2140-2150, Oct. 2017.

[13] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty and A. Y. Zomaya, "Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing," in IEEE Communications Magazine, vol. 56, no. 5, pp. 60-65, May 2018.

[14] M. Shojafar, N. Cordeschi and E. Baccarelli, "Energy-Efficient Adaptive Resource Management for Real-Time Vehicular Cloud Services," in IEEE Transactions on Cloud Computing, vol. 7, no. 1, pp. 196-209, 1 Jan.-March 2019.

[15] N. Cordeschi, D. Amendola, M. Shojafar, E. Baccarelli, "Distributed and adaptive resource management in Cloud-assisted Cognitive Radio Vehicular Networks with hard reliability guarantees," in Vehicular Communications, vol. 2, no. 1, pp. 1-12, January 2015.

[16] S. Yangui et al., "A platform as-a-service for hybrid cloud/fog environments," 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Rome, 2016, pp. 1-7.

[17] A. Yousefpour et al., "FogPlan: A Lightweight QoS-aware Dynamic Fog Service Provisioning Framework," in IEEE Internet of Things Journal.

[18] B. Tang et al., "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in Proc. ASE BigData Soc. Informat., Kaohsiung, Taiwan, 2015, pp. 1–6.

[19] Y. Yan and W. Su, "A fog computing solution for advanced metering infrastructure," 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), Dallas, TX, 2016, pp. 1-4.

[20] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan and G. J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," in IEEE Wireless Communications, vol. 23, no. 5, pp. 120-128, October 2016.

[21] W. Masri, I. Al Ridhawi, N. Mostafa and P. Pourghomi, "Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication," 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, 2017, pp. 1005-1010.

[22] I. Al Ridhawi, Y. Kotb and Y. Al Ridhawi, "Workflow-Net Based Service Composition using Mobile Edge Nodes", in IEEE Access, vol. 5, pp. 23719-23735, 2017.

[23] L. Junhai, Y. Danxia, X. Liu and F. Mingyu, "A survey of multicast routing protocols for mobile Ad-Hoc networks," in IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 78-91, First Quarter 2009.

[24] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, “A comparative survey of vanet clustering techniques,” in IEEE Commun. Surveys Tuts., vol. 19, no. 1, pp. 657–681, 1st Quart., 2017.

[25] N. Moustafa, I. Al Ridhawi, and A. Hamza, “An Intelligent Dynamic Replica Selection Model within Grid Systems,” in Proc. 8th IEEE GCC conference on Towards Smart Sustainable Solutions, pp. 1-6, 1-4 February 2015.

[26] I. Al Ridhawi, M. Aloqaily, Y, Kotb, Y. Al Ridhawi, and Y. Jararweh, “A collaborative mobile edge computing and user solution for service composition in 5G systems,” in Transactions on Emerging Telecommunication Technologies, June 2018.

[27] I. Al Ridhawi, N. Moustafa, Y. Kotb, M. Aloqaily, I. Abualhaol, “Data Caching and Selection in 5G Networks Using F2F Communication,” in Proc. 28th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2017), Montreal, Canada, 2017.

[28] I. Al Ridhawi and Y. A. Ridhawi, "A cache-node selection mechanism for data replication and service composition within cloud-based systems," in Proc. 9th International Conference on Ubiquitous and Future Networks (ICUFN), Milan, 2017, pp. 726-731.

[29] I. Al Ridhawi, N. Moustafa, and W. Masri, “Client-Side Partial File Caching for Cloud-Based Systems,” in Proc. 2016 IEEE Smart World Congress, 18-21 July 2016.

[30] I. Al Ridhawi, N. Moustafa, and W. Masri, “Location-aware data replication in cloud computing systems," in Proc. 11th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp.20-27, 19-21 Oct. 2015.

[31] I. Al Ridhawi, M. Aloqaily, B. Kantarci, Y. Jararweh, and H. T. Mouftah, “A Continuous Diversified Vehicular Cloud Service Availability Framework for Smart Cities,” Computer Networks, vol. 145, pp. 207-218, 2018.

[32] M. Aloqaily, I. Al Ridhawi, B. Kantarci, H. Mouftah, “Vehicle as a Resource for Continuous Service Availability in Smart Cities,” in Proc. 28th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2017), Montreal, Canada, 2017.

[33] M. Aloqaily, B. Kantarci and H. T. Mouftah, "Multiagent/multiobjective interaction game system for service provisioning in vehicular cloud," in IEEE Access, vol. 4, pp. 3153-3168, 2016.

[34] N. Moustafa, I. Al Ridhawi, M. Aloqaily, “Fog Resource Selection Using Historical Executions,” in Proc. IEEE 3rd International conference on Fog and Mobile Edge Computing (FMEC 2018), Barcelona, Spain, 2018.

[35] I. Al Ridhawi and Y. Kotb, “A Secure Service-Specific Overlay Composition Model for Cloud Networks,” in Journal of Software Networking, vol. 2017, no. 1, pp. 221-240, 2017.

[36] Y. Al Ridhawi and A. Karmouch, "QoS-Based Composition of Service Specific Overlay Networks," in IEEE Transactions on Computers, vol. 64, no. 3, pp. 832-846, March 2015.

[37] Y. Al Ridhawi and A. Karmouch, "Ontology-based negotiation protocol and context-level agreements," 2008 IET 4th International Conference on Intelligent Environments, Seattle, WA, 2008, pp. 1-8.

[38] I. Al Ridhawi, N. Samaan and A. Karmouch, "A QoS Monitor Selection Mechanism for Cellular Data Networks," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-7.

[39] Y. Al Ridhawi, “Dynamic Composition of Service Specific Overlay Networks,” University of Ottawa, 2013.

[40] S. Ravindran, “Accessing Higer States of Awareness: The Next Paradigm of Innovation,” Internet: https://www.beingatfullpotential.com/community/accessing-higher-states-of-awareness-the-next-paradigm-of-innovation-by-sujith-ravindran, March 2017 [Online].

[41] R. E. Jack, O. G.B. Garrod, P. G. Schyns, “Dynamic Facial Expressions of Emotion Transmit an Evolving Hierarchy of Signals over Time,” Current Biology, vol. 4, pp. 187-192, 2014.

[42] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomalydetection: Methods, systems and tools,” IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 303–336, First 2014.

[43] M. Z. Alom and T. M. Taha, “Network intrusion detection for cybersecurity using unsupervised deep learning approaches,” in 2017 IEEE National Aerospace and Electronics Conference (NAECON), June 2017,pp. 63–69.

[44] S. Soheily-Khah, P. Marteau, and N. B´echet, “Intrusion detection innetwork systems through hybrid supervised and unsupervised machinelearning process: A case study on the iscx dataset,” in 2018 1st International Conference on Data Intelligence and Security (ICDIS),April 2018, pp. 219–226.

[45] S. Otoum, B. Kantarci and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," in IEEE Networking Letters.

[46] S. Otoum, B. Kantarci and H. T. Mouftah, "Detection of Known and Unknown Intrusive Sensor Behavior in Critical Applications," in IEEE Sensors Letters, vol. 1, no. 5, pp. 1-4, Oct. 2017, Art no. 7500804.

[47] S. Otoum, B. Kantarci and H. Mouftah, "Adaptively Supervised and Intrusion-Aware Data Aggregation for Wireless Sensor Clusters in Critical Infrastructures," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.

[48] T. Ahmad, H. Chen, “A review on machine learning forecasting growth trends and their real-time applications in different energy systems,” Sustainable Cities and Society, vol. 54, pp. 102010, Mar. 2020.

[49] A. A. Aburomman and M. B. I. Reaz, “A novel svm-knn-psoensemble method for intrusion detection system,” Applied Soft Computing, vol. 38, pp. 360-372, 2016.

[50] T. M.Govindarajan, “Evaluation of ensemble classifiers for intrusiondetection,” 2016.

[51] Z. Liu, R. Wang, and M. Tao, “Smoteadanl: a learning method fornetwork traffic classification,” Journal of Ambient Intelligence andHumanized Computing, vol. 7, no. 1, pp. 121–130, Feb 2016.

[52] N. Moustafa, B. Turnbull, and K. R. Choo, “An ensemble intrusiondetection technique based on proposed statistical flow features forprotecting network traffic of internet of things,” IEEE Internet of Things Journal, pp. 1–1, 2019.

[53] Y. Guo, X. Jia, and D. Paull, “Effective sequential classifier trainingfor svm-based multitemporal remote sensing image classification,” IEEE Transactions on Image Processing, vol. 27, no. 6, pp. 3036–3048, June 2018.

[54] M. Govindarajan, “Evaluation of ensemble classifiers for intrusiondetection,” Semantic Scholar, vol. 10, no. 6, pp. 1045-1053, 2016.

[55] G. E. Hinton, “Training products of experts by minimizing contrastivedivergence,” Neural Comput., vol. 14, no. 8, pp. 1771–1800, 2002.

[56] S. Dhaliwal, A. Nahid, and R. Abbas, “Effective intrusion detection system using xgboost,” Information, vol. 9, no. 7, 2018.

[57] xgboost developers, Introduction to Boosted Trees, [online] available at https://xgboost.readthedocs.io/en/latest/index.html.

[58] H. Bouziane, B. Messabih, and A. Chouarfia, “Profiles and majorityvoting-based ensemble method for protein secondary structure prediction,” Evolutionary Bioinformatics, vol. 7, pp. 171–89, 2011.

[59] “Kdd cup 1999 data.” [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[60] C. I. for Cybersecurity. (2018) Nsl-kdd dataset. [Online]. Available:http://www.unb.ca/cic/datasets/nsl.html