SeDaTiVe: SDN-enabled Deep Learning Architecture for Network Traffic Control in Vehicular Cyber-Physical Systems

Anish Jindal, Gagangeet Singh Aujla, Neeraj Kumar, Rajat Chaudhary, Mohammad S. Obaidat, Ilsun You*

Abstract—The rapid growth in the transportation sector has led to the emergence of smart vehicles which are equipped with information and communication technologies (ICT). These modern smart vehicles are connected to the Internet to have various services such as road condition information, infotainment, and energy management. This kind of scenario can be viewed as vehicular cyber-physical system (VCPS) where the vehicles are at physical layer and services are at cyber layer. However, network traffic management is the biggest issue in the modern VCPS scenario as the mismanagement of the network resources can degrade the quality of service (QoS) for the end users. To deal with this issue, we propose a software defined network (SDN)-enabled approach, named SeDaTiVe, which used deep learning architecture to control the incoming traffic in the network in VCPS environment. The advantage of using deep learning in network traffic control is that it learns the hidden patterns in the data packets and creates an optimal route based on the learned features. Moreover, a virtual controller based scheme for flow management using SDN in VCPS is designed for effective resource utilization for providing QoS. The simulation scenario comprising of 1000 vehicles seeking various services in the network is considered to generate the dataset using SUMO. The data obtained from the simulation study is evaluated using NS-2 which proves that the proposed scheme effectively handles the real-time incoming requests in VCPS. The results also depict the improvement in performance on various evaluation metrics like delay, throughput, packet delivery ratio, and network load by using the proposed scheme over the traditional SDN and TCP/IP protocol suite.

Index Terms—Traffic control, deep learning, software-defined network, vehicular cyber-physical system.

1 Introduction

With the evolution of smart communication infrastructure and technologies, a paradigm shift has been witnessed towards provision of enhanced quality of experience (QoE) to the end users. Moreover, an exponential increase in the usage of smart devices has led to the popularity of one of the most powerful technologies of the modern era that is called as Internet of Things (IoT) [1], [2]. Using the IoT ecosystem, end users can control various smart devices across the globe during movement in their smart vehicles. With the advent of communication-enabled vehicles, users can access various services on-the-go as per their needs from anywhere which leads to the popularity of vehicular cyber-physical systems (VCPS). In VCPS, on the physical plane, smart vehicles act as the smart objects which can perform the task of computing (for example, algorithm execution) and sensing (data collection) along with interaction with different smart objects such as sensors, vehicles, or access points [3]. In the cyber plane, the underlying network provides various services (such as infotainment, itinerary management, etc.) to the end users.

However, despite their advantages, VCPS have to face various challenges which need to be tackled by designing new solutions. For example, network selection for data offloading while considering the high vehicular/user mobility is a difficult task. Secondly, smooth hand-off with respect to high mobility of the vehicles is also challenging. Security and privacy of the data gathered from different sensors is another concern. Among other issues, data analytics and forecasting to take adaptive routing decisions are the major concerns. These issues have escalated due to the existence of enormous intelligent devices and data generated from them. The data generated from these devices is heterogeneous in nature, which need an appropriate data analytical scheme to make intelligent decisions with respect to network traffic control so as to provide better QoE to the end users. Moreover, the exponential growth in the rate at which the traffic is injected in recent years has increased the overall burden on the network which degrades any designed solution. This is evident from Fig. 1, which shows the growth in the overall global traffic and the mobile to mobile traffic [4]. So, to cater these ever increasing traffic requests, an efficient traffic control strategy is required, which increases the network's quality of service (QoS) and user's QoE.

Many researchers have focused on the different aspects related to VCPS and traffic control in recent years. For instance, Wan *et al.* [5] designed an architecture to integrate the VCPS and the

- *A. Jindal is with the School of Computing & Communications, Lancaster University, Lancaster LA1 4WA, UK. E-mail: anishjindal90@gmail.com*
- *N. Kumar, and R. Chaudhary are with the Computer Science & Engineering Department, Thapar Institute of Engineering & Technology, Patiala 147004, India. E-mail: neeraj.kumar@thapar.edu, rajatlibran@gmail.com*
- *G.S. Aujla is with the Computer Science & Engineering Department, Thapar Institute of Engineering & Technology, Patiala 147004, India; also with the Computer Science Engineering Department, Chandigarh University, Mohali, India. E-mail: gagi_aujla82@yahoo.com*
- *M.S. Obaidat is with the King Abdullah II School of Information Technology, University of Jordan, Amman 11942, Jordan. E-mail: msobaidat@gmail.com*
- *I. You is with the Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, South Korea. E-mail: ilsunu@gmail.com*

* *Corresponding author.*

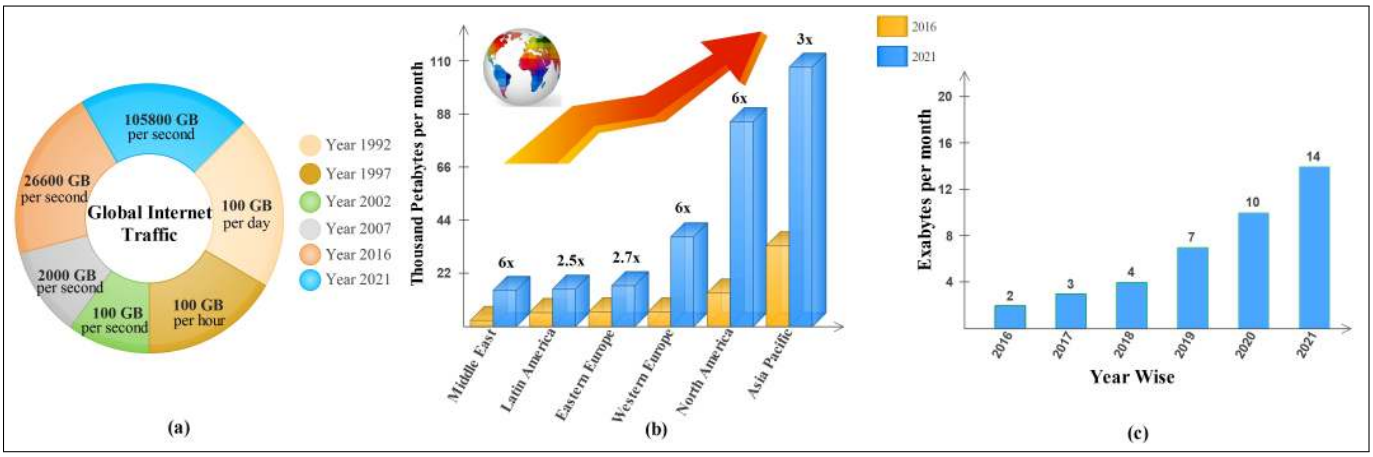

Fig. 1: Growth in (a) Global Internet traffic over the years (b) Internet traffic by regions (c) Yearwise mobile to mobile traffic [4].

mobile cloud to provide various vehicular traffic related services to the users. Similar architecture has also been presented in [6] which integrates ground vehicles and cloud computing paradigm. However, to provide the services in such environments in an efficient way, the network resources should be utilized in such a way that the overall time to access the service is minimum. To enhance QoS, Aujla *et al.* [7] proposed an offloading scheme using Stackelberg game in vehicular environment. To control the traffic in VCPS, Wang *et al.* [8] proposed a network traffic offloading strategy in VCPS to reduce the data traffic so as to improve the QoS in the network and reduce the packet loss ratio. The authors formulated a multiple-objective problem to increase the QoS in the network and used mixed integer programming for solving it to reach a global solution. However, the authors did not consider the type of services requested by the vehicles in order to assign a request priority. Moreover, finding the global solution may take more time to converge, which can increase the overall delay in providing services to the end users. To overcome all these issues, deep learning framework has been used along with SDN-enabled architecture in this paper to provide efficient services in VCPS environment.

1.1 Contributions

The major contributions of the proposed scheme are given below.

1) A SDN-enabled communication architecture is proposed for forwarding the incoming requests from vehicles to the cloud controller and *vice-versa*.
2) A virtual controller based scheme for flow management using SDN in VCPS is designed for effective resource utilization and to improve the QoS.
3) A deep learning-based traffic control mechanism is designed to find an optimal route of each data packet in the VCPS network.

2 SDN-enabled communication architecture in VCPS

VCPS is a modern system, which enables various physical objects (like sensors and actuators in the mobile vehicles) to interact with one another by providing various services such as computation, communication, and control [9]. The VCPS architecture generally comprises of two planes, namely (a) physical plane (where the data is generated), and (b) cyber plane (where the data is processed). The physical plane in VCPS consists of hardware infrastructure such as-sensors, smart vehicles, routers, and switches; while the cyber plane is the control part of the system. The VCPS communication architecture has manifold benefits. For example, it helps in building an intelligent transportation system where traffic engineering can be performed by monitoring the movement of vehicles on a road. Although VCPS allows smooth execution of distributed resources between various smart vehicles, there still persists some major challenges in VCPS. These challenges are mobility of vehicles, delay incurred for accessing the network services, and efficient network resource utilization.

In order to handle these challenges, an SDN-enabled controller is used in the proposed communication architecture of VCPS. This controller uses a virtualized control scheme to manage the network resources efficiently in dynamically changing environment. Figure 2 shows an SDN-enabled communication architecture in VCPS. SDN provides a flexible and centralized software service for the global network monitoring and control. In SDN architecture, the data plane is separated from the control plane, which reduces the computational overhead from the forwarding devices. Various layers of the SDN-enabled VCPS architecture as described below.

1. Data Plane: The communication amongst the vehicles at the data plane occurs using various communication protocols (as depicted in Table 1) [10]. Each smart vehicle has an on-board unit (OBU) for sending and receiving the data and is connected with the static control module called as road side unit (RSU). RSU is an intelligent control device, which collects the requests of vehicles in its coverage range. These requests are managed with respect to the network topology, mobility pattern of vehicles, and co-ordination of interference management of inter-RSUs. Therefore, each RSU manages a cluster of a local cloud in VCPS. This local vehicular cloud plays the role of both the cloud server (to the vehicles) and the cloud client (to the global controller). Generally, the data plane is the physical infrastructure layer comprising of OpenFlow (OF)-switches, which are deployed to perform packet forwarding. A centralized controller software is present at the control plane which is decoupled from the physical infrastructure layer. All the network policies and flow rules are implemented at the controller. The controller executes the flow rules in the form of flow tables on the forwarding devices. If the flow rule matches, then the packet is forwarded to the next OF-switch, until it reaches the

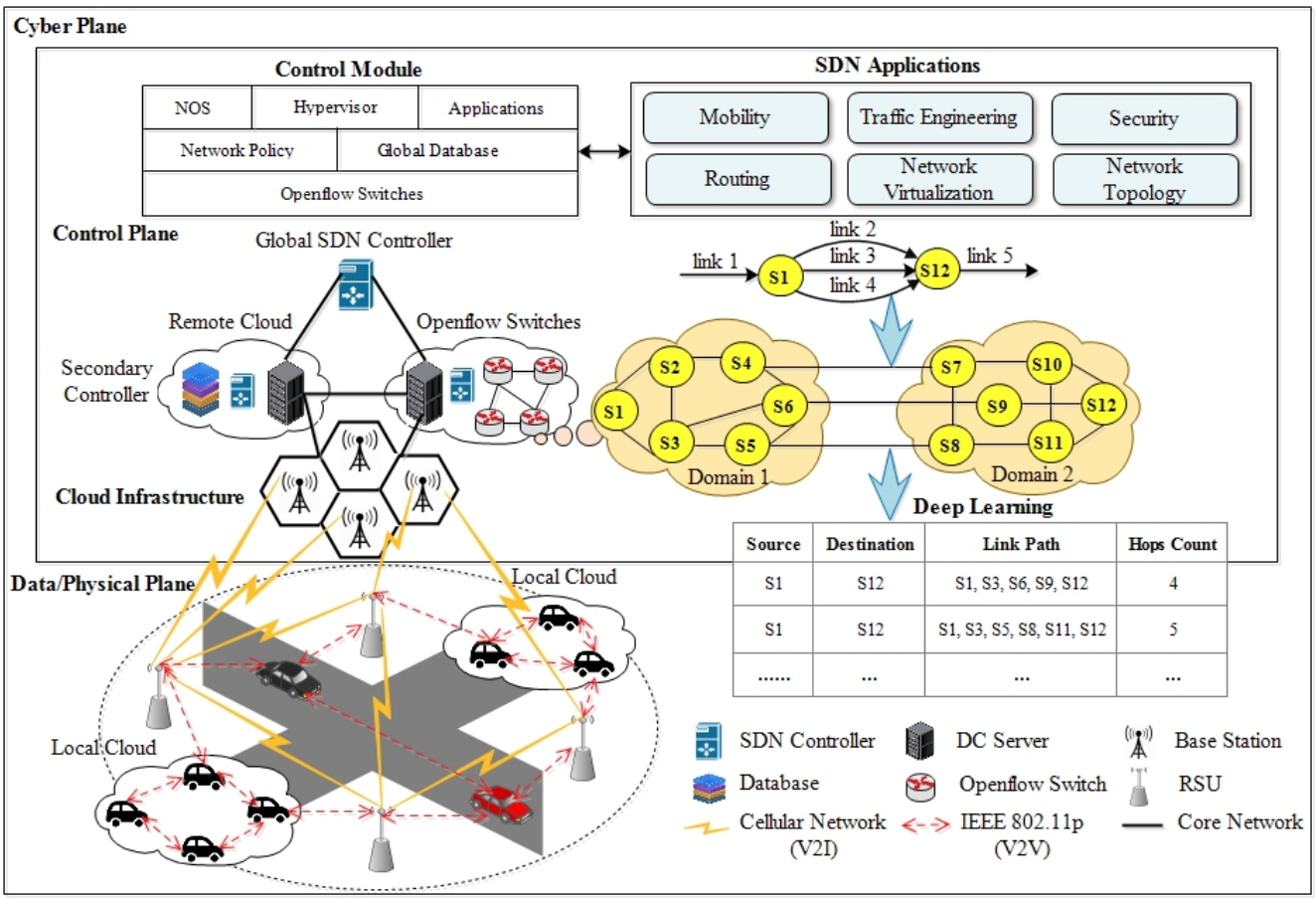

Fig. 2: Generalized communication architecture in VCPS.

TABLE 1: Communication protocols used in VCPS.

Communication	*Techniques*	*Protocols used*	*Frequency range*	*Data rate*	*Distance covered*
Vehicle to vehicle (V2V)	Bluetooth	IEEE 802.15.1	2.4 GHz	Upto 3 Mbps	upto 10 m
	Zigbee	IEEE 802.15.4	868 MHz, 915 MHz and 2.4 GHz	20-250 Kbps at 2.4 GHz	30-50 m
	UWB	IEEE 802.15.4a	3.1-10.6 GHz	53.3-480 Mbps	10-100 m
	60 GHz Millimeter Wave	IEEE 802.15.3c	57-64 GHz	>1 Gbps	upto 10 m
	DSRC/WAVE	IEEE 802.11p	5.850-5.925 GHz	3-27 Mbps	upto 1 km
Vehicle to infras-tructure (V2I)	DSA	IEEE 802.11af	476-494 MHz	1 Mbps	upto 1 km
	WAVE	IEEE 802.11p	5.850-5.925 GHz	3-27 Mbps	upto 1 km
	Wi-Fi	IEEE 802.11 a/b/g	2.4 - 5 GHz	1-54 Mbps	upto 100 m
	WiMAX	IEEE 802.16	1.25 - 20 MHz	30 Mbps - 1Gbps	upto 50 km
	Cellular (LTE/LTE-A)	-	20 - 100 MHz	300 Mbps - 3 Gbps	upto 30 km

UWB: Ultra wide band, DSRC: Dedicated short-range communication, WAVE: Wireless access in vehicular environment, DSA: Dynamic spectrum access, WiMAX: Worldwide interoperability for microwave access, LTE (-A): Long term evolution (Advanced),

destination. But, if a mismatch occurs, then the OF-switch forward the flow rule to the controller. Now, the controller builds a new flow rule, which is installed and updated on the OF-switch in order to forward the concerned packet.

2. Control Plane: At the control plane, the SDN Controller manages the control flows such as-perflow, perpacket, to perform efficient routing and flow scheduling. At the root node of the hierarchy, a global SDN controller resides, which is connected directly with the virtual controllers through core network and OF protocol. Using the network hypervisor softwares, these logically isolated virtual controllers are created on the top of a physical controller and each virtual controller is deployed in multiple domains in order to control multiple OF-switches. The virtual controllers performs two major tasks at the control plane. The first task of the virtual controllers is to create logically isolated virtual OF-switches on the top of the physical switches in each domain. The virtualization technique helps to achieve maximum utilization of network resources. Another task of the virtual controllers is to manage inter-domain routing of network nodes. The inter-domain routing helps to improve the network performance in terms of reduced latency and high throughput efficiency. In the proposed scheme, to automatically plan the optimal route for the data packets to reach the destination, a deep learning technique is used for routing decisions. For example, consider a scenario

of routing between inter-domains of network nodes as illustrated in Fig. 2. Suppose, domain 1 and domain 2 have six openflow switches each. Using link 1, source node (say S1) has requested a service from the destination node (say S12). From S1 to S12, the link path can be reached via three communication links. Now, initially, the request from S1 is floated in the network where the SDN control logic uses deep learning to find out the optimal path from S1 to S12.

3 VIRTUALIZED CONTROL SCHEME FOR FLOW MANAGEMENT IN SDN

In VCPS environment, the network resource sharing among multiple vehicles running various applications in parallel can lead to a reduction in the operational expenses. However, this may end up in various challenges such as, network traffic congestion and high latency [11]. Therefore, to handle these challenges in VCPS, an SDN-based control scheme based on virtualization is designed for traffic flow management. In this scheme, three types of data flows are considered for evaluation [12]. The first type is the active data flow, wherein flows matched successfully are included. The active data flows are served with highest priority. The second type is the waiting/queued data flows in which the flows with lower priority are considered. Such data flows are processed only after the active data flows are served. The third type is the suspended data flow, wherein the flow rule mismatch entries are included. Such cases are temporarily suspended and the notification of the mismatched entry is send to the controller for further action.

In this scheme, a single physical OFcontroller (*pOFC*) is deployed in the network. Now, multiple virtual OF controllers (*vOFCs*) are created over *pOFC* to distribute the network resources in the geographical area. Each *vOFC* is connected to a group of smart vehicles forming an exclusive cluster or local vehicular cloud. Each cluster is isolated from other clusters and *vOFC* performs its functionalities as *pOFC* does. This approach is used to divide the network resources effectively in different clusters to improve the network resource utilization. The applications running at each vehicle are served locally, closer to their location in each cluster thereby reducing latency. Similarly, the dedicated and isolated resource sharing among all the clusters lead to reduction in the network congestion. Hence, each cluster in independent to implement its own flow controls to run various vehicular applications over a single *pOFC* concurrently. Whenever required, the *vOFC* located in each cluster can communicate with each other through the *pOFC*.

Using the above discussed approach, all the forwarding devices located in each cluster use an *OF* protocol to schedule their traffic flows. In the proposed architecture, the flow table management of each isolated cluster is managed by the *vOFC*. For this purpose, each *vOFC* defines control decisions and network policies to manage the flow entries (*FEs*) in the flow tables (*FTs*) of each OF-switch. Each OF-switch comprises of multiple *FTs* stored in its memory and connected with each other via a pipeline. The flow table entries consists of three parameters namely; prefix, action, and statistics. The prefix is the matching rule field which contains a list of tuples. The matching rule contains the tuples such as-ingress port, and the packet source and destination addresses. The action parameter specifies; the action be forward to the specific port, forward to the controller, drop the packet, or forward normal to the next *FT*. The third parameter is the statistics, which consists of a counter and a timeout field. The counter field maintains the record of the number of packet matched or mismatched and the timeout field record the expiry time of a packet for a fixed amount of time or a period of inactivity. Based on these three parameters, the *FTs* are regularly updated at the *vOFC* and OF-switches.

However, updating the flow rules forwarding information generates the network overhead at the control plane. Moreover, the reconciliation of flow rules by controller in the case of mismatch and suspended entries lead to additional network overhead at the controller. The centralized *pOFC* having a global database maintains the data consistency by receiving the updated flow rules from the *vOFC*. The *vOFC* performs the task of data replication as multiple copies is quite helpful in failure recovery. In VCPS, instead of vOFCs if a single controller is being used to take efficient routing decisions regarding managing different *FTs* then the complexities may arise. The overall complexity for managing the different *FTs* in this case comes out to be $O(n.k)$, where k is the number of pOF-switches present in the network controlled by a single controller and n be the maximum capacity of number of *FEs* handled by each pOF-switch. The major complexity is related with mapping of billions of flow rules between the pOF-switch and the vOF-switch. The mapping of the flow rules at the various switches is explained with the help of example.

Example: An OF-switch OFS_j (j = 1, 2, . . . ,n) consists of FE_{jk} (k = 1, 2, . . . ,m) in the FT_j. These *FEs* in each *FT* is managed by *vOFC* using *OF* protocol. The pOF-switches contain all the *pFEs* in their *FTs* and vOF-switches consist of all the *vFEs*. The *vOFC* creates a virtual flow entry (*vFE*) corresponding to each physical flow entry (*pFE*). The entire flow mapping is performed using *vFEs*, but actually data travels on the basis of *pFEs* corresponding to each *vFE*. For flow scheduling, each *FT* must contain a matching fields MF_{jk} (an ingress switch port number and the header value) and an action field (*AF*). An *MF* is a field which is used to match the incoming traffic packet with the *FE* in the *FT* of a particular switch. Similarly, *AF* is the field which consists of the corresponding action whenever a successful matching happens. Now, a vFE^a_{jk} is created for cluster *a* having a virtual matching field (vMF^a_{jk}). In the similar way, pFE^a_{jk} for cluster *a* consists of physical matching fields (pMF^a_{jk}). Similarly, a virtual header is created as vH_i and a physical header is termed as pH_i. In order to provide isolation to each cluster, the value of header or *MF* is uniquely assigned. For example, let us consider two clusters *a* and *b*, then the vMF^a_{jk} and vMF^b_{jk} created for pMF^a_{jk} and pMF^b_{jk} are different if $a \neq b$. In the same way, this condition of inequality ($a \neq b$) exists for pH^a_i and pH^b_i. Using this approach, Fig. 3 shows the flow of a data packet from source end host (EH_{src}) located in the local vehicular cloud to the destination end host (EH_{dst}) located in another local vehicular cloud through various OF-switches. The mapping of virtual SDN layer and physical SDN layer is also shown in the Fig. 3. The flow of steps involved in the transmission process of a packet shown in Fig. 3 is elaborated as below.

1) Create *FEs* in *FTs*: The *vOFC* creates all the *FEs* for each OF-switch (S_i) located in each cluster/local cloud.
2) Create *vFEs*: All the *vFEs* at S_i in each cluster are created by the *vOFC* using OF protocol.
3) Store all *FEs* in *FT*: A flow table controller stores all the *FEs* created at each switch in different clusters.
4) Receive ad inspect the incoming packet (*p*): An incoming packet *p* is received at input port (*I*) and inspected for the MF and header value. Then, it is guided through the flow

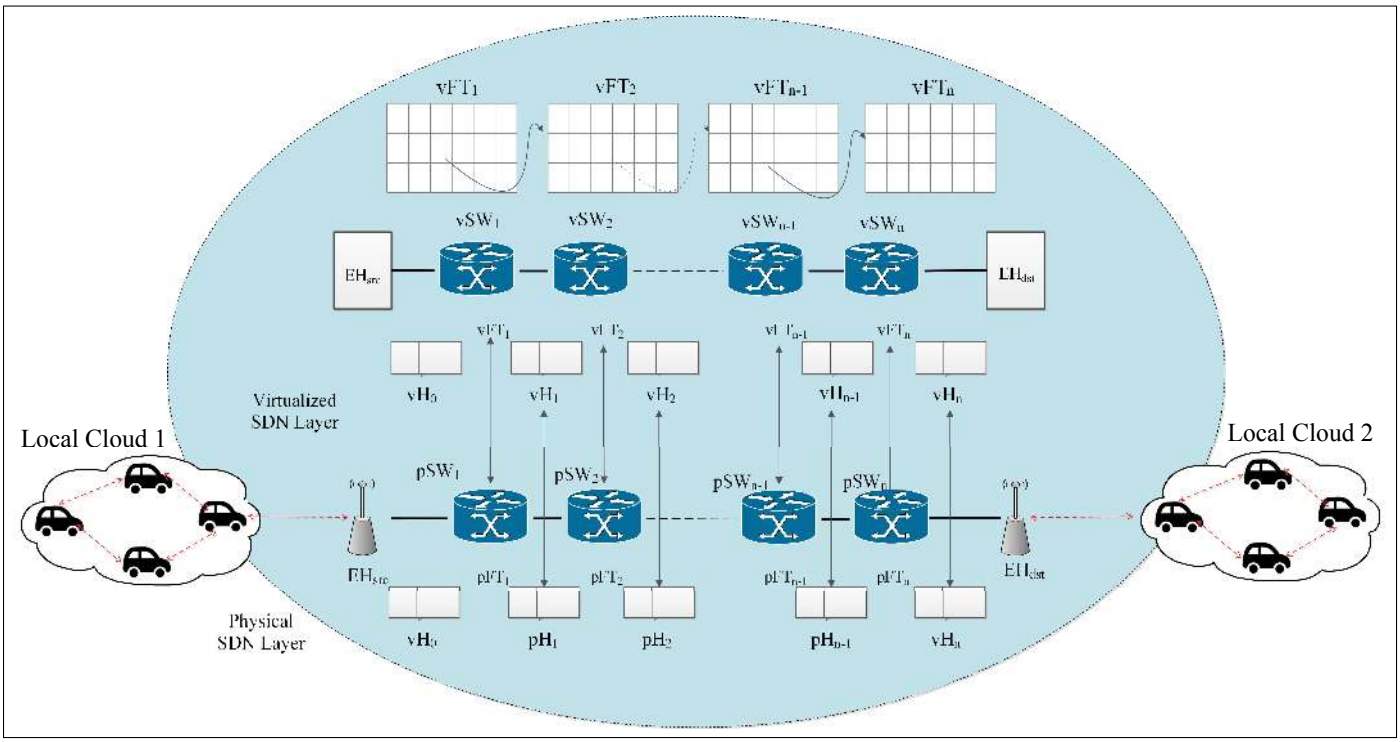

Fig. 3: Network traffic flow management using virtual SDN layer in VCPS.

process starting from EH_{src} and ending up at EH_{dst} through subsequent OF-switches (S_1 to S_n).

5) Matching process: When *p* arrives at S_1, then its *vFE* is matched. If a *vFE* exists for *p*, then the corresponding *pFE* is set at S_1 to obtain pFE_1.
6) Match priority: Sometimes, there can be a case where multiple *vFEs* are matched. For such a case, a wild-card match is performed on the basis of priority. For this purpose, the priority field (*PF*) in the *FT* is used to select the *FE*. A *FE* corresponding to the higher priority is opted and the corresponding action in the *AF* is performed.
7) Once the *pFEs* are set, then it is checked whether the switch is edge port or not. If the concerned switch is an edge port, then *p* is sent out; otherwise, it is passed to the subsequent switch where *vFE* is again searched and matched until the entire flow path is completed.
8) Once the *FE* for *p* matches, then the corresponding action in *AF* is performed on *p*. This may end up in modifying the header values or updating a switch port.
9) Now, *p* is sent from EH_{src} with a header value (H_0). Once *p* is received at S_1 with H_0, then it is forwarded to S_2 by updating its header value to H_1, such that $H_0 \neq H_1$. In this way, the header value and switch ID is incremented by *1* for each subsequent switch until *p* arrives at EH_{dst}.
10) Sometimes there may exist a case when a *FE* does not match with the *MF* at *FT*, then the cluster ID and switch ID are sent to the pOFC. After receiving the information about mismatch and the corresponding details, pOFC creates new *FE* and stores it *FT* for the concerned switch.
11) In the above discussed manner, the MF is searched and matched for S_1 to S_n in order to send *p* to EH_{dst}.
12) Once all the *FEs* are matched along the flow path, then *p* is sent out through EH_{dst}.

In the proposed scheme, the traffic flow management is handled using virtual controller in VCPS is implemented. But still, network overload

4 Deep Learning for Traffic Control

With the advances in technology, the deep learning is paving its way to solve all the complex problems of modern era. These problems range from automatic image classification to intrusion detection [13], [14]. In this paper, the application of deep learning for automatic traffic control in VCPS is described. For this purpose, the convolution neural network (CNN) model of the deep learning is used to learn the hidden patterns in the incoming data requests to assign an optimal route to the data packets.

The basic system architecture of the CNN model for traffic control and route assignment is as shown in Fig. 4 This model consists of layers viz. input, hidden, and output layers. The hidden layers comprise of convolution and pooling layers to give a final fully-connected layer as an output. So, the first step in the CNN model is to provide an input to the model. As the traffic data is labeled and structured, it can easily be converted into the two dimensional vector data where the rows depict various parameters of the traffic data and columns depict values of these parameters. This two dimensional vector forms an input layer of the CNN model. After creating the initial layer, the feature vector is passed in parts by forming groups (also known as the receptive fields) to the hidden layers. The grouping is done in order to connect the feature vector to the hidden neurons locally for a particular region. In this way, the traditional weights used in CNN for different locations are overlapped to share the same amount of information (as in MLP with full connectivity) with less connectivity. This also decreases the overall complexity of the CNN, which is very important considering the large training set.

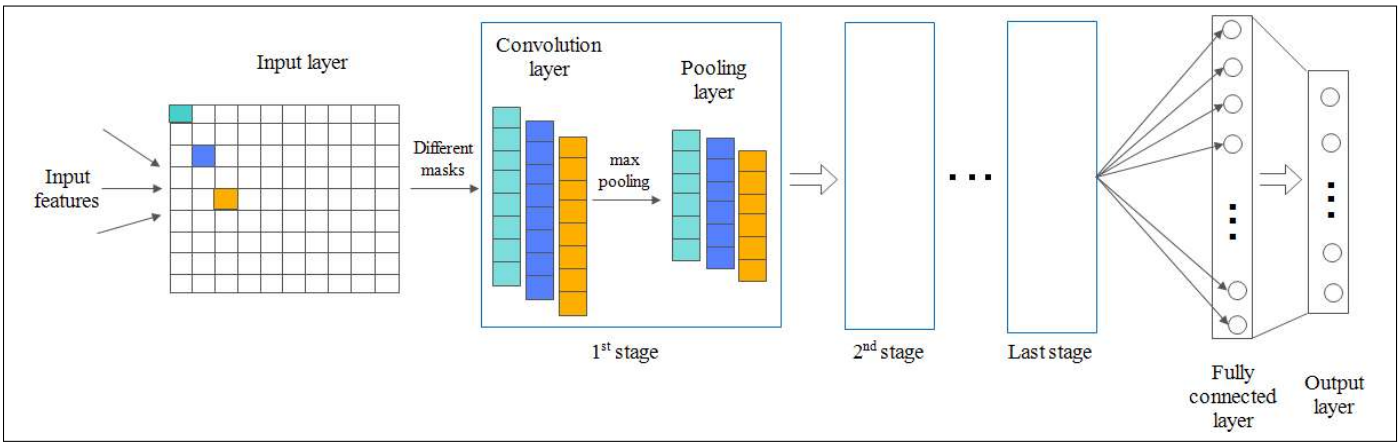

Fig. 4: CNN architecture for traffic control in VCPS.

In the hidden module of CNN model, the first hidden layer is the convolution layer. In the layer, a mask (also called function or kernel) is applied on the various receptive fields of the input feature vector. Once the mask is applied on the receptive field, the average weighted values are computed from the receptive fields, which are passed onto the neurons of next hidden layer, i.e., pooling layer. The convolution operation performed by applying this mask can be mathematically represented as:

$$conv_{xy}^{w} = \sigma\left(b + \sum_{p=0}^{n}\sum_{q=0}^{n} w_{(p,q)}\alpha_{(x+p,y+q)}\right) \quad (1)$$

where σ and b represent the activation function and the value of bias, respectively. $w_{(p,q)}$ is the weighted mask of dimension $n \times n$, and $\alpha_{(x,y)}$ is the input at $(x, y)^{th}$ position. The activation function σ used in the convolution operation can be of various types like sigmoid or *tanh*. However, rectified linear unit (ReLU) has been very successfully used in the past for deep learning applications as it is easy to implement and takes less training time [15]. ReLU is a simple discrete operation, which outputs the maximum value of the input to the neuron a [i.e, $max(0, a)$]. After applying this activation function, the output of the hidden convolution layer (also known as feature map) is passed to the pooling layer. It is to be noted here that the different masks can be applied to the input layer to create different feature maps. The task of the pooling layer is to simply the output of these feature maps so that hidden features from the input patterns can be learned from them. To perform this task, max-pooling function is used in the proposed scheme, which gives the maximum value of the considered region as an output. The major benefit of this function is that it helps to identify the data values, which are most likely to influence the outcome.

The process of convolution and pooling layers is repeated in various stages so that a final fully connected neuron layer can be created which is easy to handle. The fully connected layer is further mapped to the neurons of the final output layer (which is responsible for assigning the optimal route for the data packets). Once the output layer is created for the CNN model using the training dataset, this model is able to learn the hidden features from unknown data packets and assign the optimal route based on the learned features. This assignment is fast and optimal as it results in less delay and packet loss, and high network throughput.

Deep learning is used to manage the flow control of traffic in VCPS as follows. Initially, the data is gathered to train the deep learning model from the vehicles present in the network. Then, the network controller sends network information about the available bandwidth, resource utilization, capacity, type of service requested, protocol used, number of switches between source & destination, and path from one switch to other at different time instances. These parameters are passed onto the input layer of the deep learning which uses convolution and pooling layers to learn about the hidden patterns in the data to traverse the path from source switch to destination switch. It then repeats this process in various stages so as to learn new patterns in every stage. Once the learning phase is complete, it is able to specify the new path from one switch to another.

5 Results and Discussions

5.1 Simulation scenario

The simulation scenario consists of 1000 vehicles, which communicates with a centralized controller to access various services. This scenario is simulated and tested in NS-2 and SUMO. To train the proposed CNN model, data requests from vehicles are considered for accessing various services. All these data requests along with the network capabilities such as bandwidth, resource utilization, capacity, number of switches, etc. are given as an input to the proposed CNN model to predict the optimal flow. The deep learning model is run for 1000 epochs using the feature set mentioned above. The input layer comprises of 1000×360 entries in each epoch having the values of different feature for each vehicles for every minute. It is to be noted that the feature set is inter-dependent where the final solution depends on number of hidden layers. The number of layers is taken to be 20 so as to converge at a faster pace. Once the route is decided, the proposed scheme is evaluated on the basis of performance metrics such as-delay, throughput, packet delivery ratio and load.

5.2 Performance evaluation

5.2.1 Impact on delay

Fig 5a shows the impact on delay with increasing number of vehicles in the VCPS. It can be inferred from Fig. 5a that the delay for the proposed scheme is minimum as compared to the scenarios where only TCP/IP and SDN protocols are used. This

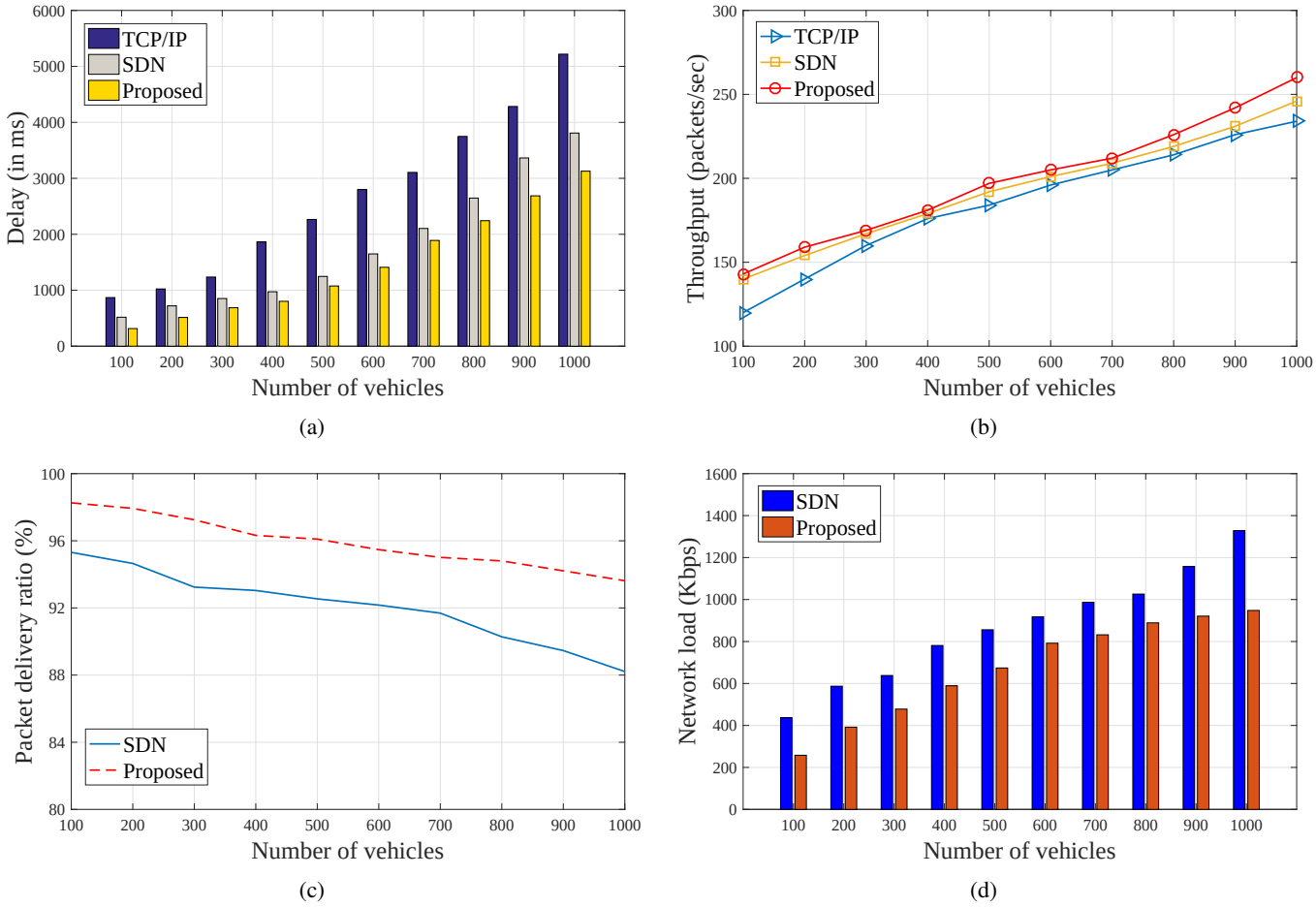

Fig. 5: Analysis of: (a) Delay incurred (b) Packet delivery ratio (c) Network throughput and (d) Load on network, with respect to number of vehicles.

is because the TCP/IP follows only the defined set of protocols, which incurs more delay as compared to the SDN protocol that is more adaptive. The proposed scheme assigns the routes using CNN model in such a way that it decreases the delay.

5.2.2 *Impact on network throughput*

Figure 5b shows the variations of the overall throughput in the underlying network with respect the number of requests processed from the vehicles in the network. This figure indicates that the throughput of the network increases in all the three cases, which is because of the increase in incoming packets. It can also be inferred from Fig. 5b that the throughput is maximum in the case of proposed scheme which proves the efficacy of our scheme. The reason behind this is the optimal route planning in an adaptive way such that the requests can be processed in consideration with the network resources.

5.2.3 *Impact on packet delivery ratio (PDR)*

The variations in PDR with respect to number of vehicles are shown in 5c. As seen in this figure, the PDR reduces with the rise in number of vehicles. It is because the network congestion increases, which leads to the frequent collisions and hence increase in the packet drop.

5.2.4 *Impact on load on network*

The load on the entire network is shown in Fig. 5d. This figure suggests that the load on the network grows with increase in the number of vehicles. It is because when more vehicles request for the network services, the consumption of network resources such as bandwidth and utilization increases, which puts more load on the underlying network. However, this load is considerably less in case of the proposed approach when compared to the other cases.

6 Conclusion

An SDN-based network architecture is used in the proposed scheme, which uses a virtualized control scheme to manage the flow in SDN. To control the network traffic, a deep learning framework is used to learn the hidden patterns in the data packets for planning their optimal routes. The effectiveness of the proposed scheme is tested using various evaluation metrics such as delay, network throughput, packet delivery ratio, and network load. The results prove the superiority of proposed scheme in comparison to the traditional variants. Although, the proposed scheme performs well in context to faster flow forwarding and optimal utilization of resources in VCPS environment, however, there are some limitations with respect to network overhead, complexity and data security. In future, security, data imbalance, overhead, complexity analysis and fault tolerance aspects in VCPS would be explored.

ACKNOWLEDGMENT

The work presented in this paper is sponsored by the Soonchunhyang University Research Fund. This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2016R1D1A1B03935619).

REFERENCES

[1] M. S. Obaidat and P. Nicopolitidis, *Smart cities and homes: Key enabling technologies*. Morgan Kaufmann, 2016.

[2] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined internet of things for smart urban sensing," *IEEE communications magazine*, vol. 53, no. 9, pp. 55–63, 2015.

[3] A. Dua, N. Kumar, and S. Bawa, "A systematic review on routing protocols for vehicular ad hoc networks," *Vehicular Communications*, vol. 1, no. 1, pp. 33–52, 2014.

[4] Cisco Visual Networking Index, "The Zettabyte Era: Trends and Analysis." CICSO, 2017. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf. [Last accessed:] Mar. 2018.

[5] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, "VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 153–160, 2014.

[6] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-Cloud: vehicular cyber-physical systems and cloud computing," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, 2011, pp. 165:1–165:5.

[7] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. Rodrigues, and A. Vinel, "Data offloading in 5g-enabled software-defined vehicular networks: A stackelberg-game-based approach," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 100–108, 2017.

[8] S. Wang, T. Lei, L. Zhang, C.-H. Hsu, and F. Yang, "Offloading mobile data traffic for qos-aware service provision in vehicular cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 118–127, 2016.

[9] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE communications surveys & tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[10] N. Kumar, K. Kaur, A. Jindal, and J. J. Rodrigues, "Providing healthcare services on-the-fly using multi-player cooperation game theory in internet of vehicles (IoV) environment," *Digital Communications and Networks*, vol. 1, no. 3, pp. 191–203, 2015.

[11] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, 2018.

[12] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, 2018.

[13] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[14] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*. IEEE, 2014, pp. 247–252.

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

BIOGRAPHIES

Anish Jindal [S'15] (anishjindal90@gmail.com) received his B.Tech. degree from Punjab Technical University, India, in 2012; M.E. degree from UIET, Panjab University, India, in 2014; and Ph.D. from Thapar Institute of Engineering & Technology, India, in 2018, all in Computer Science & Engineering. He is presently working as a postdoctoral researcher in School of Computing and Communications, Lancaster University, UK. His research interests include data analytics, smart grid, vehicular cyber-physical systems, wireless networks, Internet of things, and security.

Gagangeet Singh Aujla [S'15] (gagi_aujla82@yahoo.com) received the B.Tech and the M.Tech degrees from Punjab Technical University, Jalandhar, Punjab, India, in 2003 and 2013, respectively, and Ph.D. from Thapar Institute of Engineering & Technology, India in 2018, in Computer Science & Engineering. He has many research contributions in the area of smart grid, cloud computing, vehicular ad hoc networks. Some of his research findings are published in top cited journals such as IEEE TII, IEEE TCC, IEEE Communication Magazine, IEEE CE Magazine, FGCS, and JPDC.

Neeraj Kumar [M'16, SM'17] (neeraj.kumar@thapar.edu) is working as an associate professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering & Technology. He received his M.Tech. from Kurukshetra University, India, followed by his Ph.D. from SMVD University, Katra, in CSE. He was a postdoctoral research fellow at Coventry University, UK. He has more than 150 research papers in leading journals and conferences of repute. He is an Associate Editor of IJCS, Wiley, JNCA, Elsevier, and Security and Communication, Wiley.

Rajat Chaudhary [S'17] (rajatlibran@gmail.com) is pursuing a Ph.D. at Thapar University. He received his B.Tech degree in computer science and engineering from UPTU, Lucknow, India, in 2010, and his M.Tech degree from UTU, Dehradun, India, in 2012. He is currently a junior research fellow in Indo-Polland joint research project funded by Indian and Polish governments. His main research interests include networking, SDN, NFV, cloud computing, fog computing, and security.

Ilsun You [SM'13] received his M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, Korea, in 1997 and 2002, respectively. He received his second Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at THIN Multimedia, Internet Security, and Hanjo Engineering as a research engineer. Now, he is an associate professor in the Information Security Engineering Department, Soonchunhyang University. He is a Fellow of the IET.

Mohammad S. Obaidat [F'05] received Ph.D. and M.S. degrees in computer engineering with a minor in computer science from Ohio State University. He is a well-known worldwide academic and scientist. He is currently a full professor at King Abdullah II School of Information Technology, University of Jordan. He has published about 55 books, over 55 book chapters, and over 700 refereed technical journal and conference articles.