Ad Hoc Networks 90 (2019) 101842

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

An intrusion detection system for connected vehicles in smart cities

Moayad Aloqaily[a,*], Safa Otoum[b], Ismaeel Al Ridhawi[b], Yaser Jararweh[c]

[a] Canadian University Dubai, 1st Interchange, Sheikh Zayed Rd., Dubai, UAE
[b] University of Ottawa, School of Electrical Engineering and Computer Science, Ottawa, ON, Canada, K1N6N5
[c] Jordan University of Science and Technology (JUST), Irbid, Jordan

ARTICLE INFO

Article history:
Received 6 April 2018
Revised 31 January 2019
Accepted 1 February 2019
Available online 2 February 2019

Keywords:
Smart city
Connected vehicles
Intrusion detection
Vehicular cloud computing
Smart transportation
Service-specific clusters
QoS
QoE

ABSTRACT

In the very near future, transportation will go through a transitional period that will shape the industry beyond recognition. Smart vehicles have played a significant role in the advancement of intelligent and connected transportation systems. Continuous vehicular cloud service availability in smart cities is becoming a crucial subscriber necessity which requires improvement in the vehicular service management architecture. Moreover, as smart cities continue to deploy diversified technologies to achieve assorted and high-performance cloud services, security issues with regards to communicating entities which share personal requester information still prevails. To mitigate these concerns, we introduce an automated secure continuous cloud service availability framework for smart connected vehicles that enables an intrusion detection mechanism against security attacks and provides services that meet users' quality of service (QoS) and quality of experience (QoE) requirements. Continuous service availability is achieved by clustering smart vehicles into service-specific clusters. Cluster heads are selected for communication purposes with trusted third-party entities (TTPs) acting as mediators between service requesters and providers. The most optimal services are then delivered from the selected service providers to the requesters. Furthermore, intrusion detection is accomplished through a three-phase data traffic analysis, reduction, and classification technique used to identify positive trusted service requests against false requests that may occur during intrusion attacks. The solution adopts deep belief and decision tree machine learning mechanisms used for data reduction and classification purposes, respectively. The framework is validated through simulations to demonstrate the effectiveness of the solution in terms of intrusion attack detection. The proposed solution achieved an overall accuracy of 99.43% with 99.92% detection rate and 0.96% false positive and false negative rate of 1.53%.

1. Introduction

In the past few years, society has witnessed tremendous interest in three advanced automobile technologies, namely, Autonomous Vehicles (AVs), Connected Vehicles (CVs), and Electric Vehicles (EVs). Each one of these three technologies is projected to create significant socioeconomic values. For instance, AVs are expected to significantly reduce traffic accidents, traffic congestion, parking spaces in urban centers, and transportation costs [1,2]. On the contrary, EVs are expected to improve the resilience of power grids and to significantly reduce Greenhouse Gas (GHG) emissions. With its unique and strong capabilities, it will also provide transportation services to new communities, like children, seniors, and mobile-impaired individuals. Freight transportation is another sector that will greatly benefit from this technology [3,4].

Connected, autonomous, and electric vehicles are expected to help cities in managing issues related to public safety, services and energy. It will also enhance people's lives in terms of more efficient and secure movement. The transport electrification is also an added concept which provides real benefits in terms of oil dependency and Greenhouse Gas (GHG) emissions reduction. Connectivity and automaticity have also emerged as inseparable components of today's ITS. Fig. 1 illustrates the relation between these three technologies.

A vehicle that combines these three technologies will be so powerful that it could radically change the role of vehicles in our lives, hence, enhancing smart cities' operations [5–7]. It could play a pivotal role in public, private, or ride-hailing transportation services. It could also be part of the computing cloud and heterogeneous wireless networks [8]. Moreover, it could also serve the electric power grid. As depicted in Fig. 1, the most important and

* Corresponding author.
E-mail addresses: Moayad.Aloqaily@cud.ac.ae (M. Aloqaily), Safa.Otoum@uOttawa.ca (S. Otoum), Ismaeel.AlRidhawi@uottawa.ca (I.A. Ridhawi), YiJararweh@just.edu.jo (Y. Jararweh).

https://doi.org/10.1016/j.adhoc.2019.02.001
1570-8705/

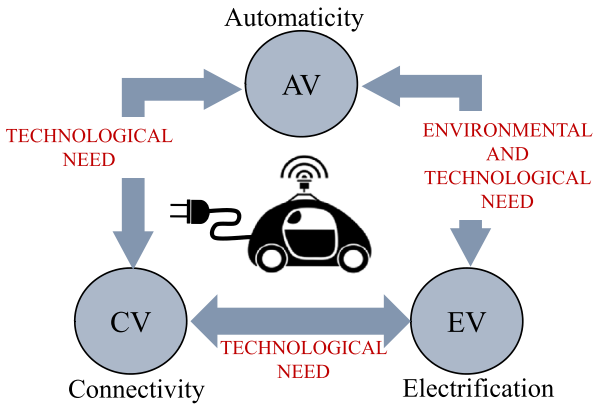


Fig. 1. Disrupting technologies that will unleash the potentials of automobile.

crucial aspect of this smart and advanced technology is connectivity, since the autonomy and electrification aspects are highly dependent on the technological needs provided by the connectivity. Fig. 1 also illustrates that these vehicles are not only providing autonomous driving but are also capable of providing autonomous electric services for stabilizing the power-grid by wirelessly charging batteries or discharging energy back (i.e. load balancing) [6,9–11]. Thus, without the proper and robust connectivity aspect, the entire concept of autonomous, connected and electric vehicle is compromised.

Smart cities necessitate methods and procedures to monitor and detect security threats. With the surge in vehicle connectivity, the susceptibility to different attacks, especially the remote cyber-attacks, has increased. Examples of such attacks have been mentioned in [12,13], which presented a scenario where a Jeep Cherokee was hacked on a highway, which led to Chrysler issuing a recall for 1.4 million vehicles.

Intrusion Detection System (IDS) is a cyber-security technique used to detect intruders and attacks in any communication system. Connected vehicles are considered one rich and critical communication environment which requires cyber security attention. IDS is employed in connected vehicles by filtering the exchanged data between vehicles. Examples of attacks that interfere with connected vehicles are the Distributed Denial of Service (DDOS), Black Hole, Sybil and Timing attacks. DDOS is considered as one of the most serious of all attacks since it prevents the user from gaining access to network services. In the connected vehicles environment, DOS attacks can manipulate with identities and broadcast fake messages to introduce a jam in the targeted network [14]. DDOS attacks target different cars at different locations and time points to carry on the same attack scenario as DOS. Employing IDS can be the best solution to detect or prevent such attacks. Black Hole attacks occur when a node drops or blocks its received packet instead of forwarding it to the receiving node, causing degradation in network efficiency. Sybil attacks occur when a malicious node defines itself as a legitimate node and starts broadcasting fake messages. Timing attacks occur in time-critical applications where malicious vehicles receive messages, where instead of broadcasting it directly, time slots are added to increase the delay [14].

Wide deployment of these vehicles will also result in many benefits to smart cities and their inhabitants such as reduced parking spaces in urban centers; reduced accidents, and significantly reducing the speed and cost of ground freight transportation. This bright future of connected vehicles in smart cities is based on the premise of seamless service management [15,16], which is a cumbersome challenge because of three elements. First, services are provided and consumed by a diverse group of individuals, enterprises, and municipalities, each having a set of expectations, preferences, and objectives. Second, different services have different cost-benefit values, which affects the supply and demand of services. Third, the high dependency on the connectivity aspect in this technology is a bottleneck and critical issue to take care of as has been shown and discussed in Fig. 1. In this article, we focus on the most critical issues in regards to connectivity and communication security.

Among the different challenges and obstacles facing connected vehicles in terms of deployment and development nowadays, securing the connectivity aspect of this technology is the most crucial of all [17–20]. Similar to any computing device connected to the Internet, connected vehicles are prone to the risk of malicious attacks. Not to overlook the fact that the automation side of these technologies is also a big reason for the increase in cyber security attacks. Such attacks can be very catastrophic and life threatening, such engine and brakes malfunction, engine overheat, control steering issues and door lock issues. Thus, the motivation of this article is to propose an Intrusion Detection System using deep belief and network decision tree to minimize any possible future security attacks against connected vehicles.

Information sharing between connected vehicles introduces security and privacy issues regarding confidentiality, message integrity, and denial of service. This article introduces an automated secure continuous cloud service availability framework for connected vehicles that enables an intrusion detection mechanism against security attacks and provides services which meet users' quality of service (QoS) and quality of experience (QoE) requirements. A Hybrid Intrusion Detection System based on monitoring using the Deep Belief Network (DBN), namely, D2H-IDS, to detect intrusions while monitoring infrastructures is proposed and evaluated. The proposed $D2H-IDS$, makes use of the Deep Belief Network (*DBN*) for data dimensionality reduction and the *ID*3 based Decision Tree (*DT*) for attack classification.

Information sharing between connected vehicles introduces security and privacy issues regarding confidentiality, message integrity, and denial of service. This article introduces an automated secure continuous cloud service availability framework for connected vehicles that enables an intrusion detection mechanism against security attacks and provides services which meet users' quality of service (QoS) and quality of experience (QoE) requirements. The main contributions of this article are summarized as follows:

- A vehicular node clustering mechanism is adapted to assure that communication to service providers is only achieved between cluster-heads, trusted-third party entities and service providers accordingly.
- A three-phase intrusion detection mechanism is proposed and evaluated. The technique is incorporated onto the cluster-heads, trusted-third parties and service providers respectively.
- The intrusion detection mechanism is a hybrid-based monitoring solution that uses Deep Belief Network (DBN), namely, D2H-IDS, for data dimensionality reduction and ID3 based Decision Tree (DT) for attacks classification.

The remainder of this article is organized as follows. Section 2 gives in-depth analysis of the related work to identify gaps that need to be taken into consideration. Section 3 presents an overview of the problem domain and solution framework. Section 4 discusses the proposed hybrid intrusion detection system and states some definitions and mathematical models that we use as base for this article. Then, we proof the soundness of the proposed system by presenting the simulation results and analysis in Section 5. We conclude the paper and illuminate the future direction in Section 6.

2. Related work

In this section, we present the most relevant related work and discuss related research studies to identify gaps that need to be taken into consideration in the proposed study in question.

A very recent research study has described the experience of driving connected vehicles like driving on the road with sharks (referring to hackers) [21]. Usually, and in order to get into the vehicle system, attackers have to be within short communication range of the vehicle in order to be able to hack it. However, nowadays attackers have access to an advanced set of resources and have developed professional skills that enables them to perform the hacking process form far distances. Nonetheless, the development of such technology is already becoming essential to the drivers and cities and will never slow down [22]. Thus, this technology is emerging, and it will overcome the fear of the cyber security threat and touch every aspect of our lives, sooner or later. In a very detailed study, Bagloee et al. has investigated the challenges and opportunities of emerging connected vehicles' technologies [23]. They also drew a very concert conclusion that, these technologies will have far-reaching application and implication beyond all expectations.

The notion of intrusion detection has been used widely in VANETs. For instance, in [24], the authors proposed an architecture for collaborative intrusion detection through distributed machine learning. The solution uses the alternating direction method of multipliers to a class of empirical risk minimization problems. It trains the classifier to detect for intrusions in VANETs. In [25], the authors proposed an intrusion detection solution for service-oriented VANETs to protect the network against denial of service, integrity target, and false alert generation. The solution relies on a set of detection rules related to each attack to determine whether a vehicle's behavior is normal or not. Moreover, a vehicle behavior evaluation protocol is developed to evaluate the trustworthiness level of a vehicle. The authors in [26] have also developed an intrusion detection mechanism for VANETs capable of detecting false information attacks using statistical techniques reliant on traffic models. It also considers key aspects such as transmission intervals and vehicle density when determining intrusions. Although the concept of intrusion detection and the use of machine learning has been used for intrusion detection in VANETs, the presented work in this paper is the first to use both clustering and trusted third parties to provide enhanced security. Three stages of intrusion detection are adopted (i.e. at the cluster head, TTP and SP) as depicted in Fig. 3. At each stage, traffic is classified for trustworthiness, thus both reducing redundant data and speeding up the learning process.

Amazon has launched a solution for connected vehicles (i.e. Amazon Web Services (AWS) [27]) which enables automotive manufacturers to build and run their applications that gather, process, and analyze on connected vehicle data. AWS provides a secure platform with low latency and overhead to manage and connect vehicles and devices to AWS infrastructure. The solution is useful since the consumers do not need to worry about the underlying infrastructure operations. However, it has not been mentioned nor articulated on how AWS ensures communication security except that it uses the TLS protocol.

In recent years, security is gaining high focus in network deployment, especially in the connected vehicle stream [28,29].The main objective of any IDS is to distinguish normal behaviors from abnormal ones and raise an alarm when attacks are detected. Although the intrusion detection literature is extremely irritating, there is no perfect IDS which can always properly differentiate between attack and normal activities with 100% detection and accuracy rates.

Machine learning-based IDS has been an effective solution to protect networks against attacks. K-means, K-nearest neighbor, decision tree, Enhanced Density-Based Spatial Clustering of Applications with Noise (E-DBCAN) and SVM are all examples of robust and effective machine learning algorithms [30].

The researchers in [31] proposed an IDS using one-class SVM with the Radial Basis Function (RBF) kernel to study the normal behavior, and distinguish abnormalities in behaviors. The resulting classifier is valid in some environments but does not detect most anomalies as expected. Hybrid approaches based on machine learning methods have been presented in [32] and [33]. The authors combined both random forest and Enhanced Density-Based Spatial Clustering of Applications with Noise (E-DBCAN) methods to work in parallel in order to detect known and unknown intruders. In [32], they presented a hybrid architecture, namely, Clustered Hierarchal Hybrid-Intrusion Detection System (CHH-IDS) to detect sensors' intrusive behaviors for both known and unknown intruders by using anomaly and signature detection techniques, respectively. Their architecture consists of two subsystems that work through duty-cycling of random forest and E-DBSCAN methods. Their method achieved a 99.731% detection rate with 98.948% overall accuracy. Other works have been conducted in [33], where they analyzed their previous model (CHH-IDS) to investigate the mitigation of False Negative behaviors through the two-tier intrusion detection approach.

Lately, researchers began investigating deep learning methods for IDS design. Deep learning methods consist of different neural network models such as Deep Belief networks (DBN), convolutional neural networks and recurrent neural networks [34]. Deep learning procedures have been applied to IDS and achieve highly accurate results in IDS applications. The essential design of any deep learning method is the use of Restricted Boltzmann Machine (RBM) [35].

An example of employing deep learning methods in the connected vehicles environment has been presented in [36] and [37]. In [36], the researchers introduced an IDS for in-vehicle networks based on Deep Neural Networks (DNN). They started by unsupervised DBN to initiate the DNN parameters as a pre-processing phase. Then, they created the data-set by using a packet generator, and finally, anomalies' behaviors were inserted by adding some noise and manipulating the packets. Another deep learning method is the one proposed in [36], which uses the Long Short-Term Memory (LSTM) and the recurrent neural network (RNN) to detect attacks. This approach works only with raw data.

In the light of smart cities, the researchers in [38] proposed an intrusion detection framework and an attack classification schema to support smart cities' administrators to define the most sensible attacks. The authors in [39] investigated security and privacy in smart city applications and presented a smart city application and architecture that monitors different vehicle's interfaces and detects intrusions. Similarly, the authors in [40] have implemented a novel digital forensics model for smart cities automated vehicles. Their system has been tested using Autonomous Automated Vehicle (AAV) cases.

It is important to mention that some of the presented methods were tested using the very old KDD'99 dataset (1999), while few have been tested using the most recent KDD dataset (i.e. NSL-KDD). In this article, we have tested the proposed model using both datasets. Moreover, to the best of our knowledge, a hybrid IDS solution for connected vehicle-based monitoring applications that combines both machine learning and deep learning methods remains an open issue.

3. Problem and solution overview

Intelligent transportation systems deployed within smart cities play a crucial role in improving the quality and delivery perfor-

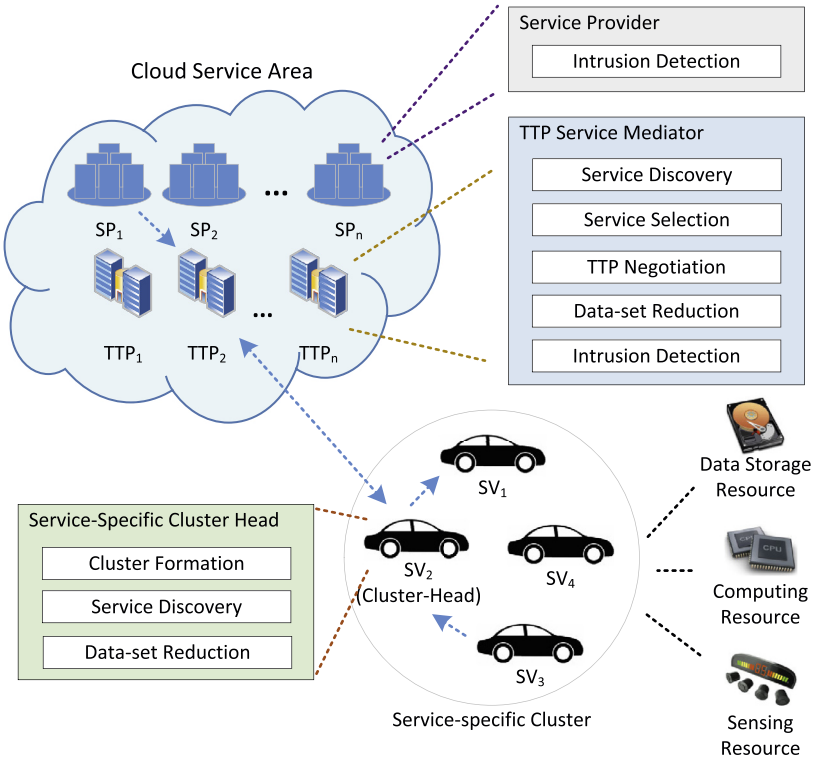


Fig. 2. Secure interconnected smart transportation system framework.

mance of diversified services. The work introduced in [41] provided a reliable service discovery and selection solution for interconnected vehicles in which smart vehicles (SV) communicate to service mediators (i.e. trusted-third parties (TTP)) through cluster-heads for service requests. Cluster-heads are selected according to a node stability and similarity identification technique as described in Section 4.1. Nodes that are identified as stable in terms of link distance and lifetime, service availability, in addition to being similar to other nearby nodes in terms of available services and relative velocity are selected as cluster-heads. Cluster-heads will serve for two purposes in return for certain profitable incentives [42–44]: i) service discovery and delivery within a cluster, and ii) request for optimal service discovery and delivery from the TTP service mediator. TTPs play the role of service buyers and sellers; buying from service providers (SPs) and selling to the SVs. A multi-agent game-theory approach is adopted by the TTP for the negotiation of services from SPs. Services are provided to SVs by TTPs at low latency and cost while revealing the minimal needed user information.

Given that cluster-heads and TTPs are crucial entities needed to select and deliver services, information sharing between such entities introduces security and privacy issues regarding confidentiality, message integrity, and denial of service. According to [45], most vehicular ad hoc networks (VANETs) use digital signatures, certificates, and timestamps to assure that exchanged messages between nodes have integrity and authenticity to prevent attacks. However, these security measures are both unfit for the adopted smart city environment and come with significant performance costs. In essence, a more robust secure solution is developed to detect intrusion attacks for intelligent transportation systems. Both cluster-heads and TTPs incorporate a security module composed of deep belief and decision tree functions to eliminate redundant data and classify incoming traffic as either malicious or normal to prevent against intrusion attacks (Fig. 2).

The introduced solution adopts a three-phase security feature incorporated within the participating entities (i.e. cluster-heads, TTPs, and SPs) as depicted in Fig. 3. Vehicles requesting cloud services from service-specific cluster heads will first have to go through the first phase of the intrusion detection mechanism. The first phase involves analyzing the received requests and nodes' descriptions to eliminate redundant data for further processing. Additionally, a vehicular node trust score is used to determine the

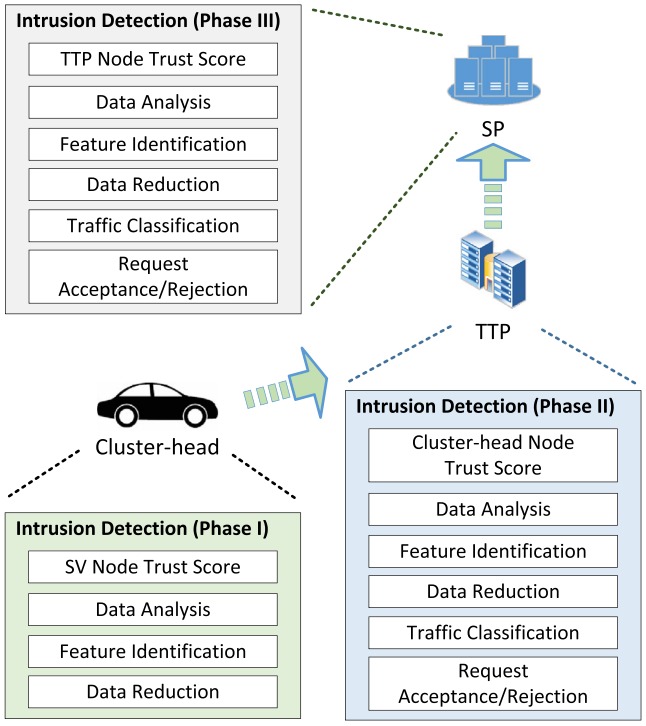


Fig. 3. Intrusion detection phases for service requests originating from SVs.

trustworthiness of the received service requests. The set of reduced trustworthy data from phase 1 (i.e. service requests not available in the cluster) are forwarded to the TTP serving the cluster head for further analysis. TTPs repeat the same process from all cluster-heads' requests, in which TTPs are assigned a node trust score. Data is then analyzed to identify the feature it belongs to. Data is further reduced by eliminating redundant data. The incoming suspicious traffic is classified according to the type of intrusion such as denial-of-service (DoS), probing, user to root (U2R), and remote to user attacks (R2L). The final stage of this phase is to either accept the service request for further processing at the SP or reject it and then add it to the set of intrusive traffic for future traffic analysis. The final phase repeats the same process involved in phase II but applied to incoming traffic from TTPs.

By adopting such a multi-stage intrusion detection mechanism, traffic is assured against intrusive security attacks. User data shared among the cloud entities will only be accessible with the communicating nodes. For instance, if an SV is requesting a particular service that is available at one of the SP sites in a cloud service area, then SV data is only shared with the cluster-head if the service is available within the same service-specific cluster. In the case where a service is not available within the same cluster, then SV data is communicated with the serving TTP, which in essence is shared with the serving SP. At each communication level, data is checked against intrusion attacks as depicted in Fig. 3. It is also important to note that since TTP is a trusted service mediator responsible of buying and selling services, user information such as personal data and interests are not disclosed with any participating cloud entity other than the serving cluster-head. TTP uses a fair secure game-theory approach with the service requesters and providers in which SP offers are not based on personal data and interests, nor purchase history [46].

Given that information sharing between vehicular nodes and other network entities (i.e. cluster-heads, TTPs, and SPs) introduces security and privacy issues such as confidentiality, message integrity, and denial of service, an intrusion detection mechanism must be incorporated within cloud vehicular networking environments to assure continuous and safe communication. According to [24–26] and [45], most vehicular ad hoc networks (VANETs) use digital signatures, certificates, and timestamps to assure that exchanged messages between nodes have integrity and authenticity to prevent attacks. However, these security measures are both unfit for the adopted smart city environment and come with significant performance costs.

4. Hybrid intrusion detection system for ITS-based monitoring

IDS is one of the key mechanisms to guarantee network security. The goal of any Intrusion Detection (ID) is to identify or detect variant attack behaviors, monitor and analyze network traffic to classify normal and abnormal behaviors, analyze system vulnerabilities and configurations, detect protection systems' weaknesses, recognize threats that arise in networks and identify the faults of the protection system administrator and the occurred abnormal activities. In this paper we study a clustered network that consists of *X* clusters each containing *Y* vehicle nodes and *Z* trusted third party nodes. In each cluster, the cluster head (CH) assumes the responsibility of data collection. Each vehicle node forwards its data to its matching CH which collects data and forwards it to the corresponding TTP. Our proposed framework is responsible for detection of intrusive nodes through a Deep belief and Decision Tree-based Hybrid Intrusion Detection System (D2H-IDS) as shown in Fig. 4.

As seen in Fig. 4, the NS3 collected traffic along with the NSL-KDD dataset pass through a three phase IDS. We have clustered and featured the NS-3 traffic before passing it to the pre-processing phase in order to make the NS-3 data format similar to

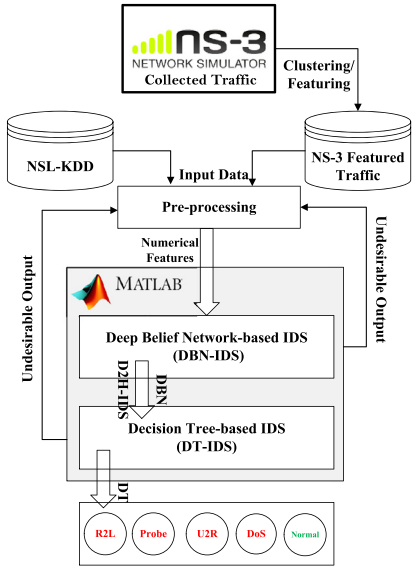


Fig. 4. Hybrid Intrusion Detection System (D2H-IDS) Architecture.

our selected NSL-KDD dataset. Our NS3 traffic is in DUMP format while our selected NSL-KDD dataset is in text format. Since the NSL-KDD dataset is in.txt format and in order to merge the input data, we have converted the NS3 trace/output (dump/tcpdump) to text format by using the tshark command. On the other hand, we used the tcpdump in Wireshark plugin in NS-3 to find our targeted features (ex. num-conn, startTime, duration, protocol, src-bytes, dst-bytes, flag,etc.) for each connection to be able to match it correctly with NSL-KDD feature format. An example is shown below:

Tcpdump record: 0, 0, "0", "0", "0", "0", 0, TCP,OTH, "0", "0",0,0,0,0,0,0,0,0,0,0,0,0.

The matching NSL-KDD record: 0,tcp,oth,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0,0.

We also used the following command in order to change our DUMP traffic file to.txt. The command takes the dump as input, filters the tcp from it and writes the features to a.txt as output. tshark -r < INPUT (DUMP) > -q -z conv,tcp > < OUTPUT (TXT) >

The pre-processing phase will process the NS-3 collected featured traffic along with the NSL-KDD dataset and pass it through three phases (i.e. D2H-IDS, DBN, and DT) of intrusion detection. It is worth mentioning here that the NS-3 collected traffic has been used in order to construct the network's normal behavior in order to differentiate between normal and malicious behaviors since it is a valuable source of information for the network's ongoing traffic. In other words, this information consists of the network's incoming and outgoing packets. On the other hand, NSL-KDD is an intrusion detection dataset and has been used for intrusion detection purposes. "Both datasets (NS-3 and NSL-KDD) have underwent the pre-processing phase such that the string features will be encoded to numerical ones. NS-3 collected traffic helps in building the normal behavior to enable IDS to differentiate between the real user from intruders. On the contrary, NSL-KDD consists of connections that represent either normal or malicious traffic (Normal, Denial of Service (DoS) attacks, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Probing attacks (Probe)).

The pre-processing output, which consists of the merged dataset, will undergo the Deep Belief Network (DBN) in order to reduce the data dimensionality, select needed features and detect whether there is an attack or not. The reduced dataset then undergoes the Decision Tree (DT) phase which is used to classify the attacks and signal alerts.Among these, the pre-processing

phase aims to encode the string features into a numerical representation. Processed traffic passes through the Deep Belief Network (DBN) which is the second phase and it aims to reduce data dimensionality as well as select needed features, whereas Decision Tree (DT) represents the third phase which focuses on classifying the type of intrusion. Undesirable outputs are directed again to the pre-processing phase whereas desirable outputs could be normal or abnormal behaviors. Abnormal behaviors raise a system alert. The Deep belief and Decision tree-based Hybrid Intrusion Detection System (D2H-IDS) phases are presented in detail in the following sub-sections.

4.1. Cluster-head selection for data collection

A Cluster-Head (CH) selection mechanism has been adopted for data collection and processing. CHs act as both service directories and service providers. Cluster participants register their services with the CH, in essence, the CH broadcasts the services that are available in the cluster to the TTPs as shown in Fig. 2. CHs may also act as SPs if the requested service is available in its cluster. CHs are also given incentives in return to the communication and routing services they provide [42]. The selection process of CHs is associated with node stability constraints which include node link lifetime, node link distance, node neighbors, and node service availability. Nodes producing the highest scores according to Eq. (1) are selected as CHs.

$$CH_i = \sum_{n=0}^{n} \left(LLT_n + \frac{1}{LD_n} \right) + SA_i \tag{1}$$

Link lifetime (LLT) is the duration of time in which two vehicles remain connected. The work in [47] introduced the concept of link lifetime which considers relative speed and distance between two nodes when calculating the link lifetime. In addition, Link distance (LD) is used to determine the average relative distance between neighboring nodes for a candidate cluster head, such that, the shorter the distance to its neighboring nodes, the higher the cluster head score. The number of neighboring nodes is also considered in the cluster head selection process. Nodes with more neighboring nodes derive higher LLT and LD scores.

Additionally, cluster heads are also selected according to the number of services they offer and the duration of which these offered services are available (SA_j). More significance is given for recently offered services using a forgetting weight factor as shown in Eq. (2).

$$SA_j = \sum_{s_j=0}^{J} \left(S_j * B^g \right) \tag{2}$$

where B^g is the forgetting weight factor for the offered service S_j and is in the range:

$$0 \leq B^g \leq 1$$

with 1 being the most recent available service and 0 being the oldest available service.

4.2. Pre-processing phase

Since network performance is tested using NS-3 traffic and NSL-KDD datasets, some of their features are represented as string values such as protocols and flags' names. The numerical encoding process of the dataset is presented since protocols' names are represented with a string. Protocols' names are encoded as TCP-001, ICMP-010 and UDP-011. REJ, RSTO and SH are examples of NSL-KDD flags where their corresponding given numerical values are 001, 010 and 011 respectively. The pre-processing phase analyzes the network records as well as the connection description to obtain the characteristics field values of the connection.

4.3. DBN-based data reduction

DBN is one of the deep learning methods composed of a number of Restricted Boltzmann Machines (RBM). RBM is a type of energy model, assuming that RBM has v visible nodes and h hidden nodes, with VI and HI respectively representing visible and hidden layer units. v_x represents x unit's state, and h_y represents y unit's state. For a given state (VI, HI), the energy is defined as follows [48]:

$$E(VI, HI|\omega) = -\sum_{x=1}^{v} a_x v_x - \sum_{y=1}^{h} b_y h_y - \sum_{x=1}^{v}\sum_{y=1}^{h} v_x h_y W_{xy} \tag{3}$$

From Eq. (3), $\omega = (W_{xy}, a_x, b_y)$, are RBM parameters where a_x, b_y are the bias of visible and hidden units respectively, W_{xy} represents the weights between x visible and y hidden units.

Description of v and h joint probability is shown in Eq. (4) below which allocates a probability to every possible pair of a visible and a hidden layer [48]:

$$P(v, h) = \frac{1}{Z} e^{-E(v,h)} \tag{4}$$

where Z is named the partition function which represents summing up all pairs of visible and hidden layers. The partition function is shown in Eq. (5).

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{5}$$

From Eqs. (4) and (5) the probability that the network assigns a visible layer is obtained as in Eq. (6) which is represented by summing up all possible hidden layers.

$$P(v) = \frac{1}{Z} \sum_{h} e^{-E(v,h)} \tag{6}$$

DBN is introduced for feature selection and data dimension reduction such as reducing the dimension of the training and testing data where the features of the raw data (input data) are mapped to the low dimension space.

Feature selection is the process of selecting the most appropriate feature-set by eliminating redundant or extraneous features. The objectives of feature selection techniques are to reduce the data dimensionality, speed-up the classification process, reduce data size, reduce storage capacity, and improve data quality. By minimizing the feature-set layer by layer, the original feature space is minimized into new feature-set which is easier to classify/predict [49].

The following lines are examples of the trained records from the KDDTrain+ dataset:

- 0, *tcp, private, S0*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 199, 3, 1.00, 1.00, 0.00, 0.00, 0.02, 0.06, 0.00, 255, 13, 0.05, 0.07, 0.00, 0.00, 1.00, 1.00, 0.00, 0.00, *neptune*, 21
- 0, *tcp, http, SF*, 287, 2251, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 7, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.43, 8, 219, 1.00, 0.00, 0.12, 0.03, 0.00, 0.00, 0.00, 0.00, *normal*, 21

It is clear from the aforementioned examples that the normal and attack records are labeled as features, so we used the DBN in order to reduce the record's number of features to prepare them for classification, as follows:

Example 1 after dimensionality reduction: 001,0,0.06,0.00,neptune.

Example 2 after dimensionality reduction: 001,0,0.00,0.03,normal.

In this paper, the constructed DBN is composed of two Restricted Boltzmann Machines (RBMs). The number of visible nodes of lower RBM is the tested attribute number and the number of

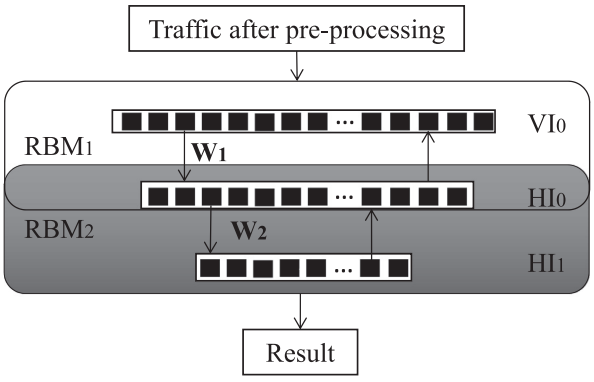


Fig. 5. Proposed DBN phase for features selection.

hidden nodes of the higher RBM is the targeted output, while the number of hidden nodes in the lower RBM layer and number of visible nodes in the higher RBM layer are the same. The proposed DBN is shown in Fig. 5, where *VI* and *HI* refer to visible and hidden layers respectively. *W* refers to the weights between both layers.

As Fig. 5 shows, the DBN is composed of layers of RBM. The first layer of the first RBM represents the traffic after the pre-processing phase. The output is fed to the second RBM as input. This process is repeated. The output of the last RBM represents the minimized features which are directed to the third phase, namely, DT classification for intrusion detection and attacks classification process.

4.4. DT based intruder's classification

Data classification is a learning technique which is used to classify the data based on various methods and algorithms. In this paper, Decision Tree (DT) is introduced to classify the minimized dataset from DBN into normal, U2R, probe, DoS and R2L. Before classification, data pre-processing and feature selection are applied to improve the classification algorithm efficiency.

DT is based on the fact that each data feature can be used to make a decision by splitting it into smaller sets, in other words, it consists of nodes that form a rooted tree. The tree consists of three sorts of nodes: The root node which has no incoming edges, Internal nodes (branches) and terminal nodes (leaves) [50]. The DT-based training phase builds both regular and intrusive patterns by using the training dataset. A labeled training dataset is delivered into the intrusion pattern builder, as a result it builds the detection module that needs the intrusive patterns.

The classification phase detects intrusions based on the generated patterns from the previous training phase. In the classification phase, the network traffic is captured by the network aggregators and altered by pre-processing operations and written into a network features database. At the end, the DT classifier decomposes the features into intrusive features that identifies the type of attack and normal features by using the patterns generated in the training phase. An alarm is raised if an intrusion is detected.

In this paper, Iterative Dichotomiser 3 (ID3) is used to construct the proposed DT. In ID3, each node corresponds to a feature whereas each arc represents the attribute's possible value. At each node, the feature is nominated to the most informative among the features not measured in the path from the root [51]. In ID3, entropy is used to measure the predictability of an event which is the amount of uncertainty or randomness in data, whereas information gain criteria is used to determine the goodness of feature split. The feature with the highest information gain is reserved as the splitting feature [51]. Entropy which is denoted by $H(E)$ for a set E is represented in Eq. (7) below [51].

$$H(E) = \sum_{x \varepsilon X} p(x) \log_2 \frac{1}{p(x)} \tag{7}$$

Lower $H(E)$ values indicate less uncertainty whereas higher values indicate high uncertainty. Information gain is denoted by $IG(E, F)$ for a set E which represents the actual change in entropy after a decision is made on a feature F $IG(E, F)$ and is represented in Eq. (8).

$$IG(E, F) = H(E) - H(E, F) = H(E) - \sum_{i=0}^{n} p(x) H(F) \tag{8}$$

where $IG(E, F)$ is the information gain by applying feature F.$H(E)$ is the entropy of set E, whereas the second term computes the entropy after applying feature F. $P(x)$ refers to an event x probability. The ID3 classification algorithm starts with creating the tree root node, computing the entropy of a state $H(E)$, computing the entropy with respect to feature F denoted by $H(E, F)$, selecting the feature with the maximum value of $IG(E, F)$, removing the feature which offers the highest IG from features set and finally, repeating the algorithm until all features are processed [52].

5. System setup and performance evaluation

The proposed D2H-IDS was implemented in Matlab 2017b [53]. As shown previously in Fig. 4, the proposed D2H-IDS takes two datasets as input: ns-3 trace for the network's normal behavior and NSL-KDD for intrusion specific behavior. Details in regards to each dataset is described in the following two subsections.

5.1. NS-3 dataset

The NS-3 [54] trace output was measured in a network of 40 vehicles attached to ITS network. The vehicles make 8 clusters that spread out in a 200 m × 200 m area. We have tested each simulation scenario 10 times. Simulation tests incorporate the *random waypoint* model [55] to simplify the problem and focus on intrusion detection issues. The simulation settings are summarized in Table 1.

5.2. NSL-KDD Dataset

KDD refers to (Knowledge Discovery in Data mining) which is a dataset that was generated in 1999 in a collaboration by DARPA and MIT Lincoln Lab. NSL-KDD is a refined version of the KDD'99 dataset proposed to solve some of KDD'99 inherent problems. Moreover, the number of records in its training and testing sets are more realistic compared to the ones in KDD'99. The NSL-KDD dataset has some advantages over the KDD'99 dataset such

Table 1
Simulation settings.

Simulation parameter	Value
Number of vehicle	40
Number of clusters	8
Number of TTPs	4
Packet size	250 Bytes
Communication range	200 m
Simulation time	600 s
Operational area	200 m × 200 m
Attack types	DDoS,DoS, Probe, U2R, L2R
Transmission rate	8 packets/sec
Traffic type	CBR
Maximum Speed (m/s)	20 m/s
Node movement	random waypoint
Type of traffic connections	TCP/UDP
Communication Protocol	IEEE 802.11.p

Table 2
Attacks in NSL-KDD training dataset.

Main attack classes	Attacks distribution
DoS	neptune, pod, smurt, teardrop, back, land
Probe	portsweep, satan, ipsweep, nmap
R2L	multihop, phf, spy, warezclient, warezmaster, ftp_write, guess_passwd, imap
U2R	loadmodule, rootkit, buffer_overflow, perl

as: First, it does not include redundant or duplicate records in the train set and test set respectively. Second, the number of records in the train and test sets are reasonable, which makes it easy to run the experiments on the whole set without the need to select a small part [56].

In this dataset, each network connection contains a total of 41 features which are categorized into three sets: 1) basic features such as duration, 2) content, and 3) the statistical features which are computed by a time window [56]. In the NSL-KDD dataset, each record represents feature values of a class in the network data flow, and each class is labeled as either attack or normal. NSL-KDD is categorized into five main classes (one normal class and four attack classes: DoS, probe, R2L, and U2R). Attacks are defined as follows [56–58]:

1. **Denial of Service (DoS)** such as apache2 which targeted the machines computing power or memory and prevented users from utilizing a service.
2. **Probe** such as ipsweep which scans a network to gather private information and find known vulnerabilities.
3. **Remote to User (R2L)** such as ftp write which represents attacks that do not have access to machines but try to gain access by sending packets over network communications.
4. **User to Root (U2R)** such as buffer overflow which are attacks that begin with machine access but attempt to have additional privileges and exploit various vulnerabilities of the system.

NSL-KDD is distributed into many files in order to facilitate the training and testing procedures [56] such as:

1. KDDTrain+.ARFF which represents the full NSL-KDD train set with binary labels in ARFF format.
2. KDDTrain+.TXT which is the full NSL-KDD train set including attack-type labels and difficulty level in CSV format.
3. KDDTrain+20%.ARFF which represents 20% subset of the KDDTrain+.arff file.
4. KDDTest-21.TXT as a subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21.

Attacks in the NSL-KDD dataset are divided into 22 different attacks, which are organized in Table 2. The total number of records in the NSL-KDD dataset is 125973, 67,343 records are considered as normal data while the rest is considered as attacks [36]. Our used test set contains 22,544 records. Table 3 presents the number of incidences for each class.

Table 3
Number of attacks and normal incidences.

Classes	Number of incidences
DoS	7460
Probe	2421
U2R	67
R2L	2885
Normal	9711

5.3. D2H-IDS training phase

Since DBN is composed of RBMs that are trained in an unsupervised fashion, its training can be simplified. The D2H-IDS training mechanism involves starting with training the first layer, gathering the data generated from the trained RBM (as learned RBM). This data will be used as the training dataset for the second RBM. This process is then repeated. The used NSL-KDD dataset consists of KDDTrain+ as the training dataset and KDDTest+ as the testing dataset composed of 125,973 and 22,544 samples, respectively. The dataset has 41 features in each data line, consisting of 38 numerical and 3 non-numerical features. These 41 features are mapped into 122 features which represent the 38 numerical features and the 84 binary encoded features from the three non-numerical ones.

We started by training the DBN model by adopting the KDDTrain+. After going through the whole DBN, the resulting aggregated data is considered as the training dataset to the ID3-based DT. The contrastive divergence algorithm has been adopted to train our DBN by training the RBMs, by learning the weights of a layer and then repeating this process for all layers. It is based on estimating the log-likelihood gradient using the j Gibbs sampling steps [59]. The generated samples form a Markov chain-based Eq. (3) which represents the energy function derived from the Gibbs distribution is calculated using Eq. (4). The learning and testing description is represented in Fig. 6. It is worth mentioning that NS3 collected traffic has been used along with the NSL-KDD dataset in all phases presented in Fig. 6.

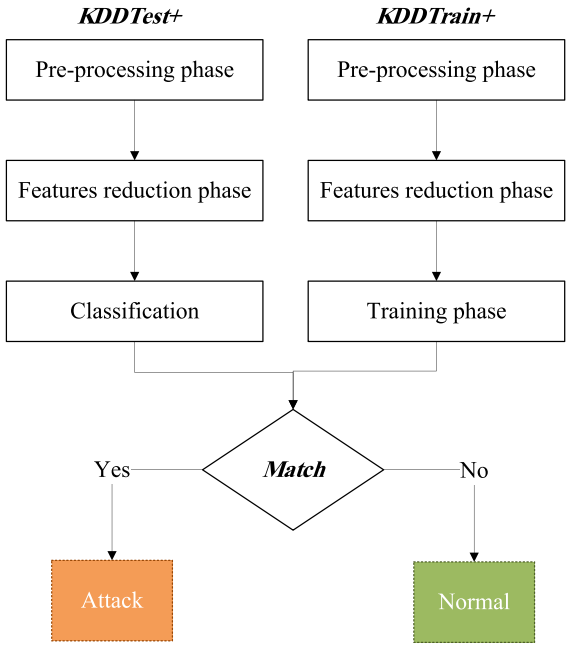


Fig. 6. Training and testing phases.

Algorithm 1 Restricted Boltzmann Machine (RBM) Fast Learning Procedure.

procedure RBM LEARNING
Initialize: W_{xy}, a_x, b_y, v.
▷ W_{xy} is the combined weight of visible v_x and hidden h_y
▷ a_x is the visible bias
▷ b_y is the hidden bias
▷ x is the number of visible nodes
▷ y is the number of hidden nodes
Set: $v_1 \leftarrow$ training input
for $y = 1, 2, 3, \ldots, h$ (hidden layers) **do**
Compute $P(h_{1y}) = (1|v_1)$
▷ $h_{1y} \in 0, 1$ as Gipps sampling from $P(h_{1y}|v_1)$
for $x = 1, 2, 3, \ldots, v$ (visible layers) **do**
Compute $P(v_{2x}) = (1|h_1)$
▷ $v_{2x} \in 0, 1$ as Gipps sampling from $P(v_{2x}|h_1)$
▷ $P(v)$ is the probability allocated to a visible element v
for $y = 1, 2, 3, \ldots, h$ (hidden layers) **do**
Compute $P(h_{2y}) = (1|v_2)$
▷ $P(h)$ is the probability allocated to a hidden element h

$$\textit{Where: } E(VI, HI|\omega) = -\sum_{x=1}^{v} a_x v_x - \sum_{y=1}^{h} b_y h_y - \sum_{x=1}^{v}\sum_{y=1}^{h} v_x h_y W_{xy}$$

▷ ω represents the RBM parameters (W_{xy}, a_x, b_y)

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

▷ $\sum_{v,h} e^{-E(v,h)}$ is the normalization factor(all possible configurations including the visible and hidden elements)

EndFor
EndFor

Moreover, the RBM fast learning procedure using Restricted Boltzmann Machine approach is shown in Algorithm 1 which highlights the steps taken during the proposed DBN learning procedure.

5.4. Evaluation measures

In order to evaluate the performance of D2H-IDS, we performed a 5-class classification procedure by adopting the NSL-KDD dataset for attacks detection purposes. DBN was applied for feature-selection whereas DT was applied for the attacks classification process. The presented D2H-IDS is evaluated based on the following criteria:

- True Positive (TP): Are the anomalous cases that were correctly classified as abnormal.
- False Positive (FP): Are the normal cases that were incorrectly classified as anomalous.
- True Negative (TN): Are the normal cases that were classified correctly.
- False Negative (FN): Are the anomalous cases that were incorrectly classified as normal.

The performance metrics used to evaluate the proposed model performance are as follows:

(i) **Accuracy Rate**
It refers to the percentage of correct predictions compared to all predictions as shown in Eq. (9) [32]:

$$Accuracy\_Rate = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

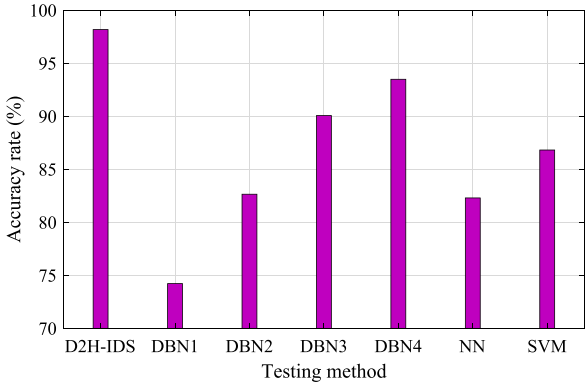


Fig. 7. Accuracy rates comparison with the ones presented in [60].

Accuracy rates represent the vital factor for any IDS performance evaluation, thus, we have compared the achieved rates with the ones presented in [60] as shown in Fig. 7. The authors in [60] presented accuracy rates comparisons between different methods such as SVM, NN and DBN with different settings. The hybrid D2H-IDS accuracy rate outperforms the other models since it incorporate machine and deep learning methods within the same model which is not the case in their other models.
Other accuracy rate comparisons are presented in Figs. 8 and 9, where the authors in [61] presented accuracy rate comparison between their deep belief network

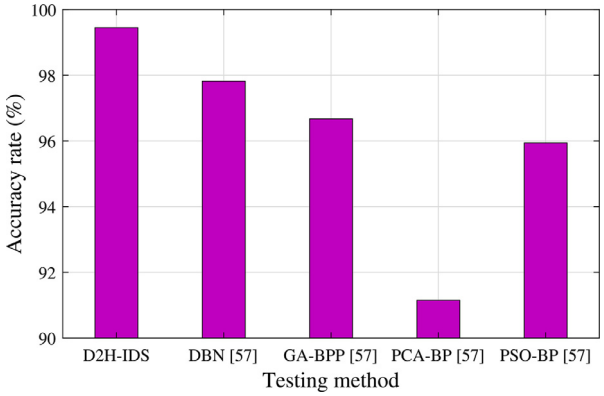


Fig. 8. D2H-IDS's Accuracy rates comparison with the work presented in [61].

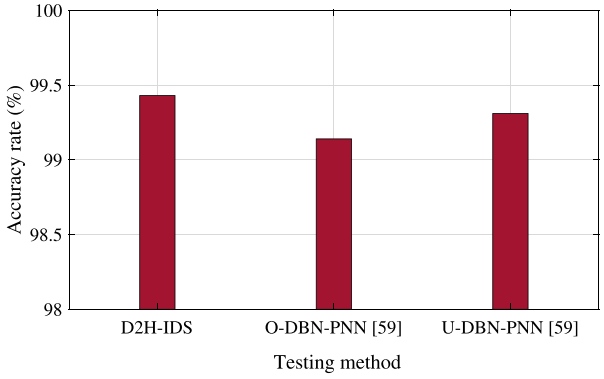


Fig. 9. D2H-IDS's Accuracy rates comparison with the work presented in [62].

Table 4
Attacks detection rates for training dataset.

Tested method	Detection rate (%)				
	DoS	Probe	U2R	R2L	Normal
DT (*DTR%*)	95.91	95.48	95.27	95.21	96.23
DBN (*DTR%*)	97.52	97.10	96.84	96.32	97.93
D2H-IDS(*DTR%*)	99.93	99.90	99.83	99.80	99.99

(DBN) and different learning methods including particle swarm optimization-back propagation (PSO-BP), Genetic Algorithms-Biophysical Profile (GA- BPP) and principal component analysis - back propagation (PCA-BP). As shown in Fig. 8, the hybrid D2H-IDS achieves an accuracy rate of 99.43% which outperforms the other models.

The authors in [62] presented accuracy rates' comparisons between an optimized DBN with a Probabilistic Neural Network ($O-DBN-PNN$) and an Unoptimized DBN with a Probabilistic Neural Network ($U-DBN-PNN$). It is clear in Fig. 9 that our proposed hybrid D2H-IDS achieves the highest accuracy rate compared to the other models (O-DBN-PNN and U-DBN-PNN).

(ii) **Detection Rate**

It refers to the percentage of behaviors that are correctly classified as attacks. It represents the true positive ratio as shown in Eq. (10) where *TP* and *FP* are the True Positive and False Positive cases respectively [63].

$$Detection\_Rate = \frac{TP}{TP + FP} \tag{10}$$

Table 4 presents the detection rates for detected main attack classes in the NSL-KDD dataset against standalone DT, standalone DBN, and the hybrid D2H-IDS. It can be seen that the proposed model achieves the highest detection percentages over all peers. It detects 99.93% of DoS attacks, which is a high value compared to the standalone DT and DBN which have achieved detection percentages of 95.91% and 97.52%, respectively. It is also clear that the hybrid model detected almost all the normal behaviors. Moreover, Table 4 shows *R2L* achieved the least detected attacks.

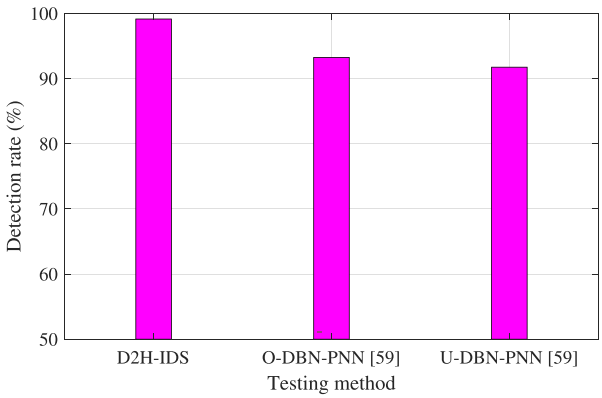


Fig. 10. Comparison of detection rates with the ones presented in [62].

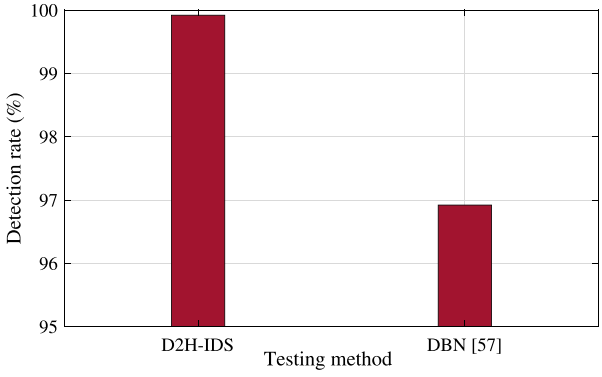


Fig. 11. D2H-IDS detection rate comparison with the one achieved in [61].

In order to ensure the originality of the proposed model, we have also compared the achieved detection rates with the ones presented in [62] as shown in Fig. 10. The authors in [62] presented detection rates comparison between an optimized DBN with Probabilistic Neural Network ($O-DBN-PNN$) and an Unoptimized DBN with Probabilistic Neural Network ($U-DBN-PNN$). The proposed hybrid D2H-IDS achieved the highest detection rate compared to the $O-DBN-PNN$ and the $U-DBN-PNN$ models with an approximate of 6% difference.

Another detection rate comparison is presented in Fig. 11 where we compared our achieved detection rate with the ones presented in [61]. We have re-simulated the work in [61] using MATLAB 2017b in order to get their model's detection rate since the authors did not consider it in their results. We used the authors DBN's settings which consists of two RBMs with 41-22-12 nodes number and with iteration of 1000 times. It is clear that D2H-IDS outperforms the model presented in [61].

(iii) **False Positive Rate (FP%)**

False Positive (*FP*) refers to the percentage of normal instances, which has inaccurately been classified as attacks, as shown in Eq. (11) where *FP, FN, TP* and *TN* are the False Positive, False Negative, True Positive and True Negative cases

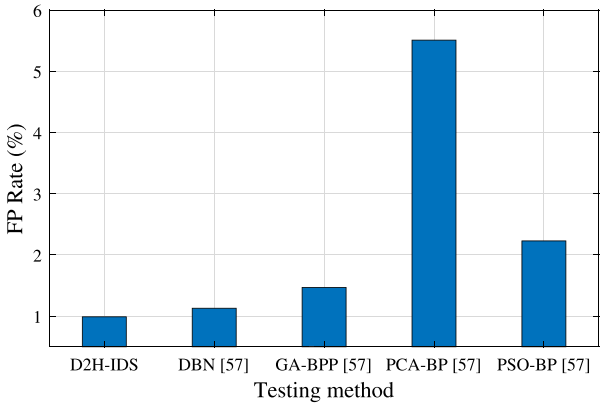


Fig. 12. D2H-IDS's FP rates comparison with the work presented in [61].

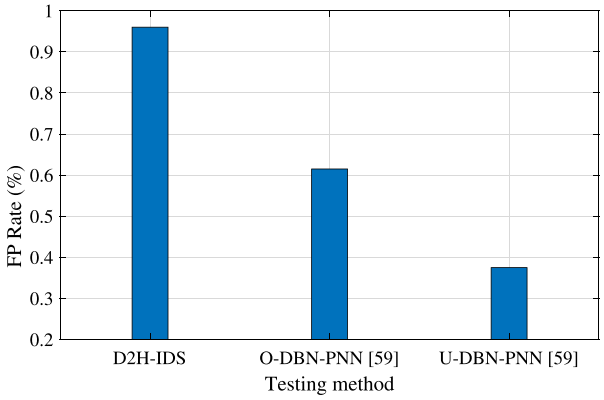


Fig. 13. D2H-IDS's FP rates comparison with the work presented in [62].

respectively.

$$FP\% = \frac{FP}{TP + FP + FN + TN} \tag{11}$$

FP% is used to describe the failure in detecting normal behaviors. In other words, an alarm has been raised. FP% is used to describe the failure in detecting normal behaviors. Fig. 12 presents a comparison of the D2H-IDS FP against the work presented in [61]. The D2H-IDS technique achieves the least FP% of 0.96% when compared to the work presented in [61].

Another FP rate comparison is the one presented in Fig. 13 where we have compared our FP rate with the one presented in [62] which the authors describe in their results as *false alarm rate*.

In Fig. 13, we have compared our proposed D2H-IDS with the results mentioned in [62], where the authors compared their proposed optimized DBN with a Probabilistic Neural Network ($O - DBN - PNN$) and an Unoptimized DBN with a Probabilistic Neural Network ($U - DBN - PNN$). It is clear that U-DBN-PNN outperforms our D2H-IDS in terms of FP rate, but our solution achieves the highest detection, accuracy, FN, and latency rates as shown in Figs. 9, 10, 15 and 16, respectively.

(iv) **False Negative Rate (FN%)**

False Negative rate (*FN*) refers to the ratio of positive cases such as a malicious node, which is incorrectly classified as negative, as shown in Eq. (12).

$$FN\% = \frac{FN}{FN + TN + TP + FP} \tag{12}$$

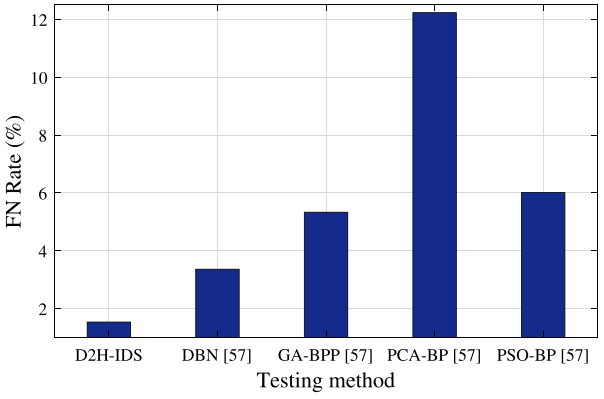


Fig. 14. D2H-IDS's FN rates comparison with the work presented in [61].

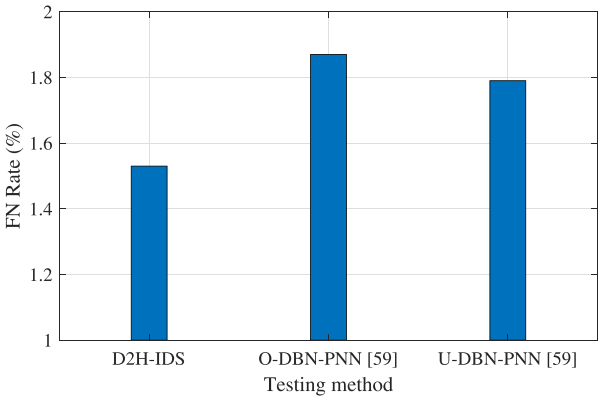


Fig. 15. D2H-IDS's FN rates comparison with the work presented in [62].

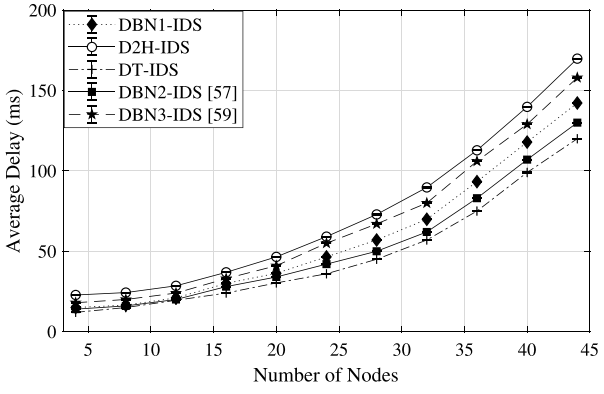


Fig. 16. Average service retrieval delay experienced by vehicular nodes using five different intrusion detection methods.

where *FN, TN, TP*, and *FP* are the False Negative, True Negative, True Positive and False Positive cases respectively. Fig. 14 presents a comparison of the D2H-IDS FN against the work presented in [61]. The D2H-IDS technique achieves the least FN% of 1.53% when compared to the work presented in [61].

The model along with its settings that were presented in [62] have been implemented and re-simulated using MATLAB 2017b in order to get its FN rate. By adopting a Particle Swarm Optimization Algorithm (PSO) of three-hidden layers of 90, 21 and 17 nodes and entering the reduced KDD traffic into a Probabilistic Neural Network (PNN) for the training and testing purposes, Fig. 15 represents the FN rate comparison between our D2H-IDS and the one adopted in [62].

(v) **Service Retrieval Delay**

An experiment was conducted to test the average delay incurred to retrieve a service from a cluster-head, TTP, or SP after traffic is detected against intrusions. A service request is first sent to the cluster-head to determine whether the requested service is available. The request is checked for any security threat using the first phase of the intrusion detection method. The service is delivered to the requester if available. Otherwise, the request is sent to the serving TTP and is checked against any security threats using the second phase of the intrusion detection method. Once the request is determined to be safe, a modified TTP request according to the game-theory service negotiation technique described in [46] is forwarded to one or more SPs. The request is finally checked against any threats using the third phase of the intrusion detection method. Safe requests are processed and the service is delivered to the requester.

We compared the hybrid D2H-IDS technique against both the DBN and DT intrusion detection mechanisms. Such that, three DBN techniques were adopted: 1) DBN1-IDS, which is the normal DBN technique adopted in this article, 2) DBN2-IDS, which is the enhanced DBN technique adopted from [61], and 3) DBN3-IDS, which is the enhanced DBN technique adopted form [62]. Moreover, the normal DT technique adopted in this article was also used to compare it against the proposed D2H-IDS technique.

Evidently, the hybrid D2H-IDS technique incurs the most delay to process and deliver a service as depicted in Fig. 16. However, when compared with the three DBN mechanisms and the DT intrusion detection mechanism alone, it is evident that the delay incurred is minimal. Results show that the D2H-IDS technique incurs a maximum average delay of 165 ms compared with 138 ms for DBN1-IDS, 126 ms for DBN2-IDS, 154 ms for DBN3-IDS and 115 ms for the DT technique, when 40 SV nodes and 4 TTP nodes are available in the environment. This translates to an increase of 39 ms in delay when compared against the most optimal DBN technique (i.e. DBN2-IDS) and 50 ms delay increase when compared against the DT technique. This amount is almost negligible given the number of nodes available in the environment and the intrusion detection advantages achieved using the D2H-IDS method.

6. Conclusion and future work

Connected vehicles provide a promising future for smart cities. With today's advances in smart vehicular technology, self-driving and autonomous service delivery mechanisms have become part of the connected vehicle vision. These capabilities coupled with other technological advances such as software-defined communication, cloud computing and storage, and IoT devices play a decisive role in increasing the welfare for smart city users. However, the high reliance of smart vehicles on communication makes it prone to various types of cyber-security attacks. Motivated by the need to address security issues of connected vehicles, and to prevent such attacks, the paper proposed a new hybrid method called D2H-IDS used for intrusion detection in smart connected vehicle cloud environments. A deep belief function is used for data dimensionality reduction, while an ID3-based decision tree technique is used for feature selection and attacks classification purposes. Through a set of ten simulations, we have shown the effectiveness of the proposed system through real cyber-security attack scenarios. The proposed solution achieved an overall accuracy of 99.43% with 99.92% detection rate and 0.96% false positive and a false negative rate of 1.53%.

Smart and connected vehicles provide a promising future for many areas such as energy harvesting [64], wireless power transfer [65] and multimedia cognitive radio networks [66]. For instance, in addition to having vehicles serve as service providers for other surrounding vehicles, smart vehicles can share power wirelessly when requested. This opens a new path for power sharing to assure that power is available continuously and on-demand to prolong an electric vehicle's driving range. Moreover, enhanced communication between vehicular nodes provides a promising future for on-demand complex composite service availability tailored according to user QoS preferences through cooperation [67,68]. For the near future, we plan to extend the proposed model to utilize big data collected from vehicular communication. Artificial intelligence models and techniques will be incorporated within the system for data analysis which will be used to improve the security of vehicles' traffic flow and road safety conditions.

References

[1] P. Godsmark, B. Kirk, V. Gill, B. Flemming, Automated Vehicles: The Coming of the Next Disruptive Technology, 2015.

[2] N. Lu, N. Cheng, N. Zhang, X. Shen, J.W. Mark, Connected vehicles: solutions and challenges, IEEE Internet Things J. 1 (4) (2014) 289–299.

[3] S.R. Narla, The evolution of connected vehicle technology: from smart drivers to smart cars to...self-driving cars, ITE J. 83 (7) (2013) 22.

[4] M. Hawes, Connected and Autonomous Vehicles-The UK Economic Opportunity, 2015. En ligne http://www.smmt.co.uk/wp-content/uploads/sites/2/CRT036586F-Connected-and-Autonomous-Vehicles—The-UK-Economic-Opportu 1.

[5] L.A. Maglaras, A.H. Al-Bayatti, Y. He, I. Wagner, H. Janicke, Social internet of vehicles for smart cities, J. Sens. Actuator Netw. 5 (1) (2016) 3.

[6] F. Mwasilu, J.J. Justo, E.-K. Kim, T.D. Do, J.-W. Jung, Electric vehicles and smart grid interaction: a review on vehicle to grid and renewable energy sources integration, Renewable Sustainable Energy Rev. 34 (2014) 501–516.

[7] M. Aloqaily, I.A. Ridhawi, B. Kantraci, H.T. Mouftah, Vehicle as a resource for continuous service availability in smart cities, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–6.

[8] M. Aloqaily, V. Balasubramanian, F. Zaman, I. Al Ridhawi, Y. Jararweh, Congestion mitigation in densely crowded environments for augmenting QoS in vehicular clouds, in: Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, in: DIVANet'18, ACM, 2018, pp. 49–56.

[9] S. Khan, D. Paul, P. Momtahan, M. Aloqaily, Artificial intelligence framework for smart city microgrids: State of the art challenges and opportunities, in: The third IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2018), 2018, pp. 1–6.

[10] A. Bilh, K. Naik, R. El-Shatshat, Evaluating electric vehicles response time to regulation signals in smart grids, IEEE Trans. Ind. Inf. 14 (3) (2018) 1210–1219.

[11] M. Amjad, A. Ahmad, M.H. Rehmani, T. Umer, A review of evs charging: from the perspective of energy optimization, optimization approaches, and charging techniques, Transp. Res. Part D 62 (2018) 386–417.

[12] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, Black Hat USA 2015 (2015) 91.

[13] A. Greenberg, Hackers remotely kill a jeep on the highway-with me in it, Wired 7 (2015) 21.

[14] R.E. Haas, D.P.F. Möller, P. Bansal, R. Ghosh, S.S. Bhat, Intrusion detection in connected cars, in: 2017 IEEE International Conference on Electro Information Technology (EIT), 2017, pp. 516–519, doi:10.1109/EIT.2017.8053416.

[15] M. Aloqaily, I. Al Ridhawi, H.B. Salameh, Y. Jararweh, Data and service management in densely crowded environments: challenges, opportunities, and recent developments, IEEE Commun. Mag. (2019).

[16] A.A. Alkheir, M. Aloqaily, H.T. Mouftah, Connected and autonomous electric vehicles (caevs), IT Prof. 20 (6) (2018) 54–61.

[17] S. Parkinson, P. Ward, K. Wilson, J. Miller, Cyber threats facing autonomous and connected vehicles: future challenges, IEEE Trans. Intell. Transp. Syst. 18 (11) (2017) 2898–2915.

[18] P. Kleberger, T. Olovsson, E. Jonsson, Security aspects of the in-vehicle network in the connected car, in: 2011 IEEE Intelligent Vehicles Symposium (IV), 2011, pp. 528–533.

[19] M. Raya, P. Papadimitratos, J.-P. Hubaux, Securing vehicular communications, IEEE Wireless Commun. 13 (5) (2006).

[20] J. Contreras, S. Zeadally, J.A. Guerrero-Ibanez, Internet of vehicles: architecture, protocols, and security, IEEE Internet Things J. (2017).

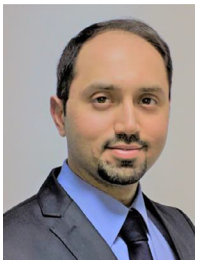
[21] M.H. Eiza, Q. Ni, Driving with sharks: rethinking connected vehicles with vehicle cybersecurity, IEEE Veh. Technol. Mag. 12 (2) (2017) 45–51.

[22] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, D. Gruyer, Chain-branch-leaf: a clustering scheme for vehicular networks using only V2V communications, Ad Hoc Netw. 68 (2018) 70–84. Advances in Wireless Communication and Networking for Cooperating Autonomous Systems

[23] S.A. Bagloee, M. Sarvi, M. Patriksson, A. Rajabifard, A mixed user-equilibrium and system-optimal traffic flow for connected vehicles stated as a complementarity problem, Comput.-Aided Civ. Infrastruct. Eng. 32 (7) (2017) 562–580.
[24] T. Zhang, Q. Zhu, Distributed privacy-preserving collaborative intrusion detection systems for VANETs, IEEE Trans. Signal Inf. Process. Networks 4 (1) (2018) 148–161.
[25] K. Zaidi, M.B. Milojevic, V. Rakocevic, A. Nallanathan, M. Rajarajan, Host-based intrusion detection for VANETs: a statistical approach to rogue node detection, IEEE Trans. Veh. Technol. 65 (8) (2016) 6703–6714.
[26] H. Sedjelmaci, S.M. Senouci, M.A. Abu-Rgheff, An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks, IEEE Internet Things J. 1 (6) (2014) 570–577.
[27] AWS, Aws connected vehicle solution, 2018, URL https://aws.amazon.com/answers/iot/connected-vehicle-solution/.
[28] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H.M. Zhang, J. Rowe, K. Levitt, Security vulnerabilities of connected vehicle streams and their impact on cooperative driving, IEEE Commun. Mag. 53 (6) (2015) 126–132.
[29] P. Sharma, H. Liu, H. Wang, S. Zhang, Securing wireless communications of connected vehicles with artificial intelligence, in: 2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1–7, doi:10.1109/THS.2017.7943477.
[30] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutorials 18 (2) (2016) 1153–1176.
[31] A. Theissler, Anomaly detection in recordings from in-vehicle net-works, Big Data Appl. 23 (2014).
[32] S. Otoum, B. Kantarci, H.T. Mouftah, Detection of known and unknown intrusive sensor behavior in critical applications, IEEE Sens. Lett. 1 (5) (2017) 1–4, doi:10.1109/LSENS.2017.2752719.
[33] S. Otoum, B. Kantarci, H.T. Mouftah, Mitigating false negative intruder decisions in wsn-based smart grid monitoring, in: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp. 153–158, doi:10.1109/IWCMC.2017.7986278.
[34] S. Ludwig, Intrusion detection of multiple attack classes using a deep neural net ensemble, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–7.
[35] K. Alrawashdeh, C. Purdy, Toward an online anomaly intrusion detection system based on deep learning, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 195–200, doi:10.1109/ICMLA.2016.0040.
[36] M.R. Parsaei, S.M. Rostami, R. Javidan, A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset, Int. J. Adv. Comput. Sci. Appl. 7 (6) (2016) 20–25.
[37] L. Dhanabal, D.S.P. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, 2015.
[38] V. Garcia-Font, C. Garrigues, H. Rifà-Pous, Attack classification schema for smart city WSNs, Sensors 17 (4) (2017) 771.
[39] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, X.S. Shen, Security and privacy in smart city applications: challenges and solutions, IEEE Commun. Mag. 55 (1) (2017) 122–129.
[40] X. Feng, E.S. Dawam, S. Amin, A new digital forensics model of smart city automated vehicles, in: Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on, IEEE, 2017, pp. 274–279.
[41] I.A. Ridhawi, M. Aloqaily, B. Kantarci, Y. Jararweh, H.T. Mouftah, A continuous diversified vehicular cloud service availability framework for smart cities, Comput. Netw. 145 (2018) 207–218.
[42] I. Al Ridhawi, N. Samaan, A. Karmouch, Simulator-assisted joint service-level-agreement and vertical-handover adaptation for profit maximization, in: Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on, IEEE, 2012, pp. 74–82.
[43] M.X. Goemans, L. Li, V.S. Mirrokni, M. Thottan, Market sharing games applied to content distribution in ad hoc networks, IEEE J. Sel. Areas Commun. 24 (5) (2006) 1020–1033.
[44] H. Susanto, B. Kaushik, B. Liu, B. Kim, Pricing and revenue sharing mechanism for secondary redistribution of data service for mobile devices, in: Performance Computing and Communications Conference (IPCCC), 2014 IEEE International, IEEE, 2014, pp. 1–8.
[45] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.-J. Nordlund, Particle filters for positioning, navigation, and tracking, IEEE Trans. Signal Process. 50 (2) (2002) 425–437.
[46] M. Aloqaily, B. Kantarci, H.T. Mouftah, Multiagent/multiobjective interaction game system for service provisioning in vehicular cloud, IEEE Access 4 (2016) 3153–3168.
[47] E. Natsheh, T.-C. Wan, Links lifetime estimation based on nodes affinity in wireless ad-hoc networks, in: High Capacity Optical Networks and Enabling Technologies, 2008. HONET 2008. International Symposium on, IEEE, 2008, pp. 39–45.
[48] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800, doi:10.1162/089976602760128018.
[49] M.Z. Alom, V. Bontupalli, T.M. Taha, Intrusion detection using deep belief networks, in: 2015 National Aerospace and Electronics Conference (NAECON), 2015, pp. 339–344, doi:10.1109/NAECON.2015.7443094.
[50] D.M. Farid, N. Harbi, M.Z. Rahman, Combining naive Bayes and decision tree for adaptive intrusion detection, CoRR (2010) arXiv:1005.4496.
[51] A. Pujari, Data Mining Techniques, Universities Press, 2001. URL https://books.google.ca/books?id=dH2KQhJboSYC.
[52] Decision trees for classification: a machine learning algorithm, 2018URL https://www.xoriant.com/blog/product-engineering/.
[53] MATLAB URL https://www.mathworks.com/.
[54] ns-3 URL https://www.nsnam.org/.
[55] J. Harri, F. Filali, C. Bonnet, Mobility models for vehicular ad hoc networks: a survey and taxonomy, IEEE Commun. Surv. Tutorials 11 (4) (2009).
[56] C.I. for Cybersecurity, NSL-KDD dataset, 2018, URL http://www.unb.ca/cic/datasets/nsl.html.
[57] KDD cup 1999 data,. URL http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[58] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, in: CISDA'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 53–58. URL http://dl.acm.org/citation.cfm?id=1736481.1736489.
[59] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[60] N. Gao, L. Gao, Q. Gao, H. Wang, An intrusion detection model based on deep belief networks, in: 2014 Second International Conference on Advanced Cloud and Big Data, 2014, pp. 247–252, doi:10.1109/CBD.2014.41.
[61] B. Wang, S. Sun, S. Zhang, Research on feature selection method of intrusion detection based on deep belief network, in: The 3rd International Conference on Machinery, Materials and Information Technology Applications (ICMMITA 2015), Atlantis Press, 2015, pp. 556–561.
[62] G. Zhao, C. Zhang, L. Zheng, Intrusion detection using deep belief network and probabilistic neural network, in: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 1, 2017, pp. 639–642, doi:10.1109/CSE-EUC.2017.119.
[63] S. Otoum, B. Kantarci, H.T. Mouftah, Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures, in: IEEE International Conference on Communications (ICC'18), 2018, pp. 1–6.
[64] A. Kariminezhad, S. Gherekhloo, A. Sezgin, Optimal power splitting for simultaneous information detection and energy harvesting, IEEE Signal Process. Lett. 24 (7) (2017) 963–967.
[65] D. Bavastro, A. Canova, V. Cirimele, F. Freschi, L. Giaccone, P. Guglielmi, M. Repetto, Design of wireless power transmission for a charge while driving system, IEEE Trans. Magn. 50 (2) (2014) 965–968.
[66] M. Amjad, M.H. Rehmani, S. Mao, Wireless multimedia cognitive radio networks: a comprehensive survey, IEEE Commun. Surv. Tutorials 20 (2) (2018) 1056–1103.
[67] I. Al Ridhawi, M. Aloqaily, Y. Kotb, Y. Al Ridhawi, Y. Jararweh, A collaborative mobile edge computing and user solution for service composition in 5G systems, Trans. Emerging Tel. Tech. 29 (2018) e3446, doi:10.1002/ett.3446.
[68] I.A. Ridhawi, Y. Kotb, Y.A. Ridhawi, Workflow-net based service composition using mobile edge nodes, IEEE Access 5 (2017) 23719–23735.

Moayad Aloqaily received the M.Sc. degree in Electrical and Computer Engineering from Concordia University, Montreal, QC, Canada, in 2012, and the Ph.D degree in Electrical and Computer Engineering from the University of Ottawa in 2016. He was an instructor in the Systems and Computer Engineering Department at Carleton University, Ottawa in 2017, Canada. He works with Gnowit Inc. since 2016. From 2017 to 2018, he was an assistant professor with the Computer Engineering department at the College of Engineering and Technology, American University of the Middle East (AUM), Kuwait. He is currently an Assistant Professor of computer engineering at the Faculty of Engineering, Canadian University Dubai, UAE. His current research interests include Connected Vehicles, Intelligent Transportation Systems, Cloud and Edge Computing, Vehicular Cloud Computing, 5G Networks, Wireless Communications/Networks. He is an IEEE member and actively working on different IEEE events. He was the president of Electrical Engineering Graduate Student Association (EEGSA) at the University of Ottawa 2014-2016 and the IEEE Photonics Society chair for the year 2016/2017. He is a Professional Engineer Ontario (P.Eng.).

Safa Otoum received her Master degree in electrical and computer engineering from the University of Ottawa (Canada) in 2014. She is currently a Ph.D. candidate in school of electrical engineering and computer science at the University of Ottawa, Canada. Her research interests include networks security issues, intrusion detection and prevention, wireless sensor networks, and smart grid. She is an IEEE student member.



Ismaeel Al Ridhawi received his BASc, MASc, and Ph.D degrees in Electrical and Computer Engineering from the University of Ottawa, Canada, in 2007, 2009, and 2014 respectively. He was with the College of Engineering and Technology, American University of the Middle East, Kuwait, as an Assistant Professor, from 2014 to 2019. His current research interests include quality of service monitoring, network service management, overlay networks, fog computing, and mobile edge computing.

Yaser Jararweh received his Ph.D. in computer engineering from University of Arizona in 2010. He is currently an associate professor of computer science at Jordan University of Science and Tech. He has co-authored several technical papers in established journals and conferences in fields related to cloud computing, edge computing, SDN and Big Data. He is a steering committee member and co-chair for CCSNA 2018 with Infocom. He is the General Co-Chair in IEEE International conference on Software Defined Systems SDS-2016 and SDS 2017. He is also chairing many IEEE events such as ICICS, SNAMS, BDSN, IoTSMS and many others. Dr. Jararweh served as a guest editor for many special issues in different established journals. Also, he is the steering committee chair of the IBM Cloud Academy Conference. He is associate editor in the Cluster Computing Journal (Springer), Information Processing & Management (Elsevier) and others.