IET Networks

Review Article

Artificial intelligence enabled software-defined networking: a comprehensive overview

ISSN 2047-4954
Received on 3rd June 2018
Revised 28th September 2018
Accepted on 5th November 2018
E-First on 10th December 2018
doi: 10.1049/iet-net.2018.5082
www.ietdl.org

Majd Latah[1] ✉, *Levent Toker*[2]

[1]Department of Computer Science, Ozyegin University, 34794 Istanbul, Turkey
[2]Department of Computer Engineering, Ege University, 35100 Izmir, Turkey
✉ *E-mail: majd.latah@ozu.edu.tr*

Abstract: Software-defined networking (SDN) represents a promising networking architecture that combines central management and network programmability. SDN separates the control plane from the data plane and moves the network management to a central point, called the controller that can be programmed and used as the brain of the network. Recently, the research community has shown an increased tendency to benefit from the recent advancements in the artificial intelligence (AI) field to provide learning abilities and better decision making in SDN. In this study, the authors provide a detailed overview of the recent efforts to include AI in SDN. The study showed that the research efforts focused on three main sub-fields of AI namely: machine learning, meta-heuristics and fuzzy inference systems. Accordingly, in this work, the authors investigate their different application areas and potential use, as well as the improvements achieved by including AI-based techniques in the SDN paradigm.

1 Introduction

Software-defined networking (SDN) adopts the concept of programmable networks by using a logically centralised management, which represents a simplified solution for complex tasks such as traffic engineering [1], network optimisation [2] and orchestration [3]. Furthermore, dealing with modern network applications requires more scalable architecture which should be able to provide reliable and sufficient services based on a specific traffic type [4]. This can be achieved with the SDN architecture, which maintains a global view of network states and provides a flow-level control of the underlying layers [4]. This idea caused a dramatical change in the way how networks are designed and managed [5, 6]. In addition, the SDN architecture allows the involvement of third parties in the design and deployment of modern network applications [6]. Ethane project [7] formed the foundation of today's SDN by presenting a new networking paradigm, in which a centralised controller is used for flow-level policy management and security purposes in enterprise networks.

The SDN paradigm separates the control plane from the data plane, which results in achieving much faster and dynamic approach compared with a conventional network architecture [8]. The control plane can be split into several virtual networks where each one implements a different policy [8]. As a result, this paradigm can be viewed as a tool allows addressing various issues in networking from another perspective [5] and can be used also to fulfil the requirements of new technologies such as internet of things (IoT) and 5G [9]. The adoption of SDN paradigm, however, strongly depends on its success in reaching an appropriate solution for the problems, which cannot be solved by the traditional networking protocols and architectures [10]. Some large companies such as Microsoft and Google have already started using the SDN paradigm for their own data centres [11, 12]. Artificial intelligence (AI), on the other hand, reveals a huge potential in SDN innovation. Our previous study [13] focused on highlighting the first efforts to integrate AI in SDN. In this work, however, we provide a thorough overview of the research efforts in this area to gain a deeper insight into the significant role of AI in SDN paradigm.

2 SDN architecture

As mentioned previously, SDN promotes innovation by introducing the concept of centralised programmable control of the data plane, which facilitates the development of new network services and protocols [4]. The SDN architecture is designed based on the idea of the separation between control and data planes (see Fig. 1).

The first attempt was network control point [14], which employed the separation concept to enhance the control of AT&T's telephone network, whereas recent contributions such as Ethane [7] and SANE [15] applied the same concept for Ethernet networks [16]. The applications of SDN reside in the application plane of SDN architecture where the northbound application programming interface (API) provides the commutation between the application and control planes [8], which allows implementing a set of network services such as traffic engineering, intrusion detection, quality of service (QoS), firewall and monitoring applications [4]. Northbound API allows developers to write their own applications without the need for a detailed knowledge of the controller functions or understanding how the data plane works. It is worth mentioning that several SDN controllers provide their own northbound APIs [16].

The communication between control and data planes is provided using a southbound API such as forwarding and control element separation [17], open vSwitch database [18], protocol oblivious forwarding [19], OpenState [20], OpenFlow (OF) [21] and OpFlex [22], which enables exchanging control messages with forwarding elements (e.g. OF-enabled switches). As shown in Fig. 1, each OF-enabled switch adopts a flow-based decision making logic determined by the so-called SDN controller, which is responsible for preparing the forwarding tables of each switch [8]. A typical OF-enabled switch has a pipeline of flow tables, which consist of flow entries, each of which has three parts: (i) matching rules, which are used for matching incoming packets; (ii) counters that maintain statistics of matched flows; and (iii) actions or instructions, which can be configured proactively or reactively to be executed upon a match [6, 14]. The forwarding elements (i.e. OF-enabled switches) can be implemented in either software or hardware. Some software switches such as Open vSwitch have a great potential for providing a solution for data centres and virtual networks [16]. On the other hand, other APIs [10, 23] are proposed for a specific purpose (e.g. VOIP applications and inter-domain routing), not to mention various SDN programming languages such

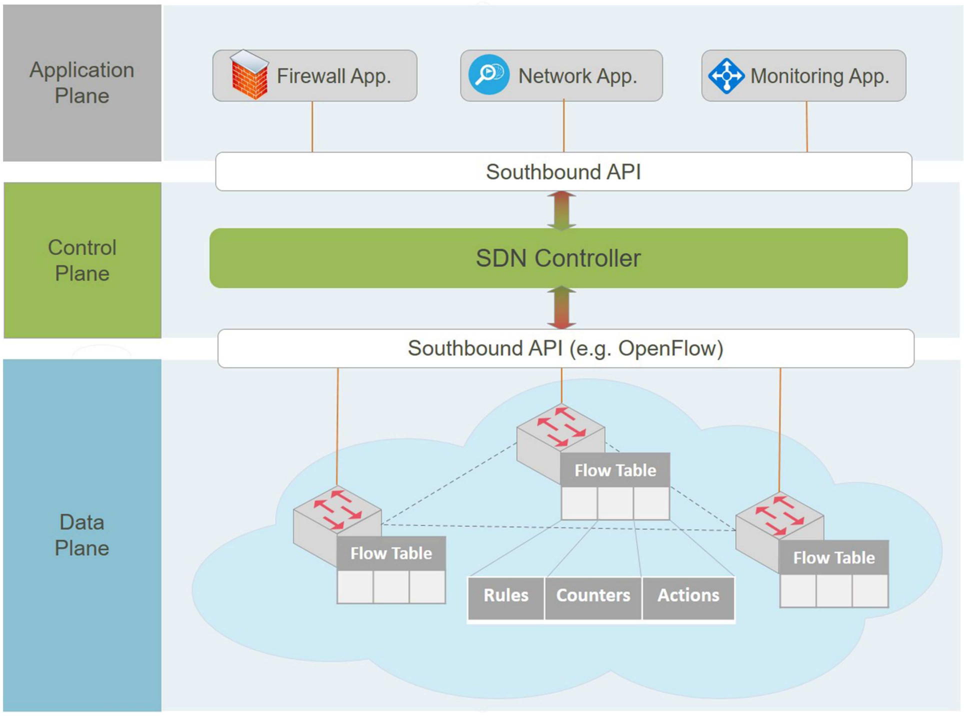

Fig. 1 *Basic SDN architecture*

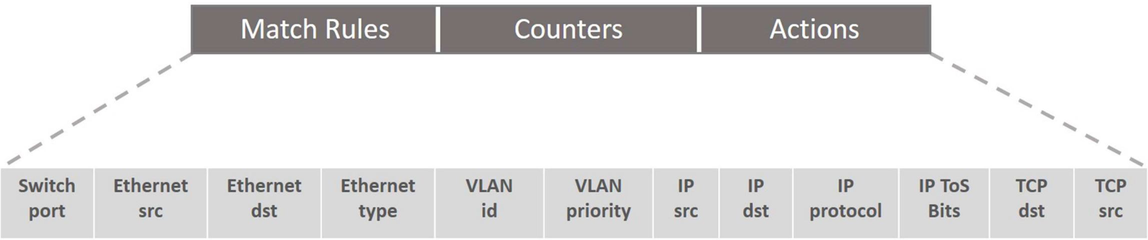

Fig. 2 *Structure of OpenFlow V1.0.0 matching rules*

as Procera [24], NetCore [25] and Frenetic [26], which provide high-level APIs that can be used to develop different SDN applications in more flexible and functional manner.

3 OpenFlow protocol

OF is considered the most commonly used southbound API in SDN, which is being continuously developed and standardised by open networking foundation [6]. OF provides an abstraction layer that enables the SDN controller to securely communicate with OF-enabled forwarding elements [6]. OF has become the de-facto standard for southbound APIs used in SDNs [6] and therefore, in this study, we mainly focus on OF-based SDNs. OF-based forwarding devices have been developed to coexist together with conventional Ethernet devices [16]. Hybrid switches, on the other hand, reveal new possibilities by including both OF and non-OF ports [6]. As we mentioned previously, a set of control messages can be sent by the controller to prepare and update a particular switch's flow tables. A typical OF-enabled switch handles new coming packets based on its flow table. Fig. 2 shows the fields of matching rules part in OF version 1.0.0. A table-miss occurs when a new packet does not match any of the flow table entries. In this case, the switch may either drop the packet or forward it to the corresponding controller using OF protocol [6].

It is worth mentioning that the identity-based access control of the Ethane project [7] became the first specification of OF switches. After the release of the first OF specification in Spring 2008, many vendors such as HP, NEC, Cisco and Juniper were able to build their first OF-enabled hardware switches, where NOX [27] was the only available controller at that period of time [28]. Recently, different versions of OF protocol have been introduced to add more flexible and reliable capabilities by including multiple flow tables, enhanced matching/action abilities, optical ports, group tables, meter tables and synchronised tables [4, 29]. More details concerning various versions of OF specification can be found in [30]. In addition, there are many available OF controllers, such as POX [31], Beacon [32], OpenDayLight [33], Floodlight [34] and Ryu [35].

4 Artificial intelligence

AI is a rapidly growing field that includes a wide range of sub-fields, including knowledge representation, reasoning, planning, decision making, optimisation, machine learning (ML) and meta-heuristic algorithms. Turing test [36] provides a fulfilling definition of intelligence, where a computer has to answer some questions written by a human interrogator. Accordingly, the computer passes the test if the interrogator cannot tell whether the answer was written by a human or a machine. In order to pass Turing test, the computer needs to include advanced capabilities such as natural language processing, knowledge representation, automated reasoning, ML and computer vision [36]. AI research started later in the mid-1950s, where a summer workshop organised by Martin Minsky and Claude Shannon at Dartmouth College resulted in the birth of the field of AI [37]. The first contribution, however, was made in 1943 by McCulloch and Pitts, when they proposed the first model for artificial neural networks, in which each neuron has a binary output (−1, +1) with a sign activation function [37]. Adoption of AI approaches increased, thanks to key contributions led to the emergence of new sub-fields such as expert systems, fuzzy logic (FL) and evolutionary computation (EC). Further efforts have fuelled the research in AI field by refining the existing methods and proposing brand new hybrid intelligent approaches. ML, meta-heuristic algorithms and fuzzy inference systems are widely used in SDN paradigm, therefore we provide an introduction regarding these approaches.

4.1 Machine learning

An intelligent machine learns from experience (i.e. learns from the data available in its environment) and uses it to improve the overall performance [36, 37]. In this context, learning methods fall into four groups.

4.1.1 Supervised learning: Supervised learning methods are provided with a pre-defined knowledge. For instance, a training dataset that consists of input–output pairs, where the system learns a function that maps a given input to an appropriate output [36]. This approach requires having a dataset that represents the system under consideration and can be used to estimate the performance of the selected method [38].

Artificial neural networks: Neural networks (NNs) are mainly inspired by biological learning systems such as biological neurons in human brain [37]. Artificial neural networks have many advantages. First, they can adjust themselves to the data without explicitly specifying a functional or distribution for representing the underlying model [39]. Second, NNs form a universal functional approximator, which can approximate any function [39]. Third, NNs are non-linear models, which gives them the flexibility to represent and model complex relationships [39]. The feed-forward multilayer networks or multilayer perceptrons (MLPs) are the most commonly used NN classifiers. MLPs are mainly trained with supervised training algorithms. NNs are subject to over-fitting when we use too many parameters in our model [36]. We also need to find the best network structure to achieve a good performance [36].

Support vector machines (SVMs): This algorithm finds linear separators that maximise the margin between two different classes in order to provide a better generalisation of the classifier. Kernel methods can be used to transform the input data into a high-dimensional space in order to deal with linearly non-separable cases [36]. SVMs can represent complex functions and show robustness against over-fitting [36].

Decision trees (DTs): DTs are very successful in classification problems. DT can describe a dataset by a tree-like structure [37]. The input and output data can be discrete or continuous. DTs can represent all Boolean functions. A DT performs a sequence of tests, where each internal node in the tree corresponds to a test of one of the input attributes [36]. Interpretability (i.e. the ability to understand the reason for the output of the learning algorithm) is one of the main advantages of DT, since this approach is very natural for humans [36]. DT, however, suffers from over-fitting where it may lead to a large tree when there is no pattern to be found in the input data [36].

Ensemble methods: The ensemble methods combine predictions of different approaches (by weighted or unweighted voting) and mainly used for improving the performance of learning algorithms [36]. Bagging represents the first effective method used for increasing the accuracy by creating an improved composite classifier that combines different outputs of learned classifiers into a single output. Each one of these classifiers is trained by instances generated by random sampling with replacement from the original dataset [36, 40]. In boosting, unlike bagging, each classifier is influenced by the performance of the previous classifier and tries to pay more attention to the errors made by the previous classifier [40]. For more details, we refer the reader to [40].

Supervised deep learning: Deep learning provides a general-purpose multi-level representation-learning approach [41]. In representation learning a machine can learn to automatically discover the representations required for classification or detection task based merely on raw data, whereas conventional machine-learning techniques cannot deal natural data in their raw form [41]. Multiple levels of representation allow transforming the representation from low level into a higher abstract one. An enough number of these transformations allows learning more complex functions [41]. Deep learning techniques [41, 42] have achieved better performance compared to the traditional algorithms used for many ML tasks such as speech recognition, intrusion detection, objection detection and natural language understanding. Deep learning models are categorised into three groups, namely: (i) generative, (ii) discriminative and (iii) hybrid models. Discriminative models mainly use supervised learning approaches, whereas generative models employ unsupervised learning approaches. Hybrid models, on the other hand, make use of both discriminative and generative models [42, 43]. In this paper, the term deep neural network (DNN) refers to deep feed-forward multilayer networks or MLPs. Other important two supervised models used in deep learning are recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

Recurrent neural networks: RNN is an extension of feed-forward neural networks (FFNNs) that addresses sequential (e.g. speech or text) or time-series problems [43, 44]. Unlike traditional FFNN, the output of an RNN depends on the previous computations [44]; this is the basic reason why it is called an recurrent neural network [44]. The back-propagation through time algorithm is mainly employed for the training stage of an RNN. However, the conventional RNN encounters vanishing/exploding gradient problems. Long short term memory (LSTM) networks and gated recurrent units (GRUs) were proposed to cope with this problem [44].

Convolutional neural networks: CNNs were proposed to process and deal with the data that comes in the form of multiple arrays [41] such as images. CNNs are used in feature learning for large-scale image classification. A typical CNN consists of three layers: (i) convolutional layer, (ii) sub-sampling layer (pooling layer) and (iii) fully-connected layer [45]. In the convolutional layer, a filtering operation performed by a feature map (i.e. discrete convolution) is used to attain the weight sharing, whereas the sub-sampling is employed in the pooling layer for dimensionality reduction [41, 45].

4.1.2 Unsupervised learning: Unsupervised learning methods are provided without a pre-defined knowledge (i.e. having an unlabelled data) [36]. Therefore, the system mainly focuses on finding specific patterns in the input. An example of the unsupervised learning approach is clustering, which is used for detecting useful clusters in the input data based on similar properties defined by a proper distance metric such as Euclidian, Jaccard and cosine distance metrics [36, 38].

K-means clustering: K-means [46] is one of the well-known clustering approaches. A prior knowledge of the parameter *k*, which indicates the number of the resulted clusters, is needed for this algorithm. Each data point will be assigned to the nearest centroid of each cluster. K-means minimises an objective function that represents the distance between the data points and their corresponding centroids [47]. The process of updating the centroids, based on their assigned data points, will be repeated until the centroids remain the same or no point changes. K-means depends mostly on the initial set of clusters. Therefore, an inappropriate choice of *k* may result in poor results [48]. Moreover, fuzzy-C-means [49] clustering, which is also known as soft K-means, allows each data point to belong to more than one clusters. In other words, a data point can belong to all clusters with different degree of membership.

Self-organising maps (SOMs): SOM [50, 51] is a well-known unsupervised learning approach in artificial neural networks, which maps high-dimensional distribution to a low-dimensional representation called a SOM map [52]. SOMs have proven to be successful in various pattern recognition tasks including very noisy signals [51]. The training process in SOM builds and reorganises the map using input data. Thereafter, it classifies a new input vector based on finding its winning neuron or node in the map [52].

Hidden Markov model (HMM): HMM [46] represents a statistical model, where the system under development is assumed to be a Markov process with unobserved (hidden) states [53]. The Markov process is the random process, which fits for the Markov assumption, where Markov assumption is that the probability of one state depends only on the previous state [53]. The HMM specifies five entities in the model which are: (i) the set of states, (ii) the output alphabet, (iii) the initial probability state, (iv) transition probabilities and (v) the observation probability. The parameters of HMM can be trained in supervised or unsupervised manner [53]. Baum-Welch algorithm [46] is considered to be the most commonly used unsupervised algorithm in HMM [53].

Restricted Boltzmann machine (RBM): An RBM represents a stochastic ANN that consists of two layers: input layer and hidden layer [43]. The restriction of RBMs compared with basic Boltzmann machine is that the connectivity of the neurons, where each neuron in the input layer is connected to all of the hidden neurons and vice versa, but there is no connection between any two neurons in the same layer [43]. Moreover, the bias unit is connected to all of the visible and hidden neurons. RBMs are an essential component in deep belief networks (DBNs) and can be used for feature extraction [43].

Unsupervised deep learning approaches: As we mentioned previously, deep architectures are classified into: (i) generative, (ii) discriminative and (iii) hybrid models. We mentioned also that the generative models employ unsupervised learning approaches to characterise the high-order correlation properties of the input data [42]. The generative models need an unsupervised pre-training stage to extract the structures in the input data. They also need an additional top layer to perform the discriminative task [42]. In this paper, we discuss two deep generative models namely stacked auto-encoder (SAE) and DBN, due to the fact that they were already employed in SDNs [54–57].

Stacked auto-encoder: Auto-encoder (AE) is suitable for feature extraction and dimensionality reduction [43]. A basic AE has two stages: (i) encoding and (ii) decoding stages [45]. The first stage receives the input data and transforms it to a new representation, called a code or latent variable, whereas the second stage receives the generated code at the first stage and reconstruct the original input data [43]. The training procedure aims at minimising the reconstruction error [43]. An SAE is trained by a two-stage approach. The pre-training stage includes training the initial parameters in a greedy layer-wise unsupervised style, whereas the next fine-tuning stage makes use of a supervised approach to fine-tune the parameters of the model with respect to the labelled instances by adding a softmax layer on the top layer [45].

Deep belief network: DBN is a type of generative ANNs that represents the first successful deep learning model in which several RBMs can be stacked into a deep learning model, called DBN [45]. DBNs extract hierarchical representation of the training data, as well as reconstruct their input data [43]. The efficiency of DBN as deep learning model comes from the fact that the training of a DBN is performed layer by layer, where each layer is treated as an RBM trained on top of the previous trained layer [45]. DNB is trained by a two-stage approach similar to the previously mentioned one for training SAEs. DBNs are suitable for hierarchical features discovery [43].

4.1.3 Reinforcement learning (RL):

In RL, the system learns based on a set of reinforcements from its environment. For instance, a reward or punishment determines whether the system performed well or not [36]. Each interaction with the environment returns an information, which the system makes the best use of this information to learn and update its knowledge [58]. A key concept in RL is the Markovian property (i.e. only the current state affects the next state) [58].

Q-learning: Q-learning, a form of model-free RL allows agents to act optimally in controlled Markovian domains without the need for building maps of these domains [59]. The task for the agent is determining an optimal policy that maximises the total discounted expected reward, called also Q, for executing a particular action at a particular state [59]. Q-learning is classified as incremental dynamic programming because it finds the optimal policy in step-by-step manner [59]. Watkins [59] has proved that Q-learning converges with probability one under reasonable conditions on the learning rates and the Markovian environment.

Deep RL: Deep learning gives RL the ability to scale-up to decision-making problems with high-dimensional state and action spaces [58]. Deep RL basically depends on DNNs for approximating the optimal policy [58]. Deep RL can leverage the representation learning to deal with the problem of the curse of dimensionality [58]. For example, it can use the CNNs to learn from high-dimensional raw data [58]. AlphaGo represents one of the promising success for deep RL, which defeated the world champion in Go. AlphaGo depended on NNs that were trained using supervised learning, RL and a traditional heuristic search algorithm [58].

4.1.4 Semi-supervised learning:

In semi-supervised learning, the system learns from both labelled and unlabelled data, where the lack of labels, as well as the labelled part may contain a random noise forms a situation between supervised and unsupervised learning [36]. For many real-world applications it is more realistic to rely on unlabelled data that does not require any additional cost or further expert-based labelling process [60]. As it includes some small labelled data, the performance of semi-supervise learning approaches is superior to unsupervised learning [60]. Table 1 shows a comparison between the main types of ML approaches.

4.2 Meta-heuristic algorithms

The heuristic algorithms use a problem-specific heuristic, whereas the meta-heuristic algorithms form an efficient general purpose approach that includes a wide range of application areas ranging from finance to engineering and networking [64]. Meta-heuristics have been increasingly employed to solve hard optimisation problems [65] that cannot be solved by any deterministic (exact) approach within a reasonable time [64]. Most of these algorithms are nature-inspired [65], from simulated annealing (SA) [66] to genetic algorithms (GAs) [67], and from ant colony optimisation

Table 1 Comparison of ML approaches [61–63]

ML-approach	Advantages	Disadvantages
supervised learning	learns from labelled data	requires a dataset that represents the system
	generalises well based on a sufficient dataset	the data is manually labelled by human experts, which is not appropriate for many real-world applications
unsupervised learning	finds hidden patterns without relying on labelled data	it may not provide a useful insight into the hidden patterns and what actually they mean
	performs better for unseen data compared with supervised approach	
semi-supervised learning	learns from both labelled and unlabelled data	it may lead to worse performance when we choose wrong assumptions
	certain assumptions about the underlying data distribution must be met	
reinforcement learning	dynamically adaptation and gradually refinement	there is a trade-off between exploration and exploitation. In addition, we need to specify a reward function, parameterised policy, strategy and initial policy
	an agent interacts with an uncertain environment, in which the goal is maximise the agent's reward. It can also be used for difficult problems that have no analytic formulation	

(ACO) [68] to whale optimisation algorithm (WOA) [69]. These algorithms are widely used to solve difficult problems such as combinatorial, highly non-linear and multi-modal optimisation problems [70]. Each one of these algorithms has different advantages, therefore many efforts were made to design hybrid approaches that combine their benefits and ultimately attain better results [70]. The success of these algorithms is determined by achieving a balanced performance between the exploration and the exploitation [64]. Exploitation is useful for determining the most promising high-quality solutions in the search space, whereas exploitation is needed to concentrate search in some areas based on previous search results [64]. The main disadvantage of meta-heuristics is that these methods find good solutions rather than a guaranteed optimal solution. Another drawback is that these methods include a large number of parameters that need to be set in order to produce a good solution [71].

4.2.1 Ant colony optimisation: ACO [68] is a swarm intelligence population-based meta-heuristic algorithm for finding the solution of combinatorial optimisation problems. ACO was inspired by the foraging behaviour of ants in nature. At first, the ants start randomly exploring the area surrounding their nest. Along the path they selected ants deposit a chemical pheromone trail, which guides the other ants to the food sources founded by the previous ants. After a period of time, the concentration of pheromone will increase along the shortest path of the food source. Pheromone evaporation helps in avoiding the problem of premature convergence [64].

4.2.2 Evolutionary algorithms (EAs): EA, also known as EC), is the main term for many optimisation algorithms that are developed based on Darwinian theory of nature's capability to revolve and survival of the fittest [64]. EA domain includes GAs, evolution strategies, evolutionary programming and genetic programming [64].

Genetic algorithm: The GA [67, 72] is one of the well-known and mostly used population-based technique. In GA, a solution of an optimisation problem is represented by a chromosome. A set of chromosomes forms the population. Two basic yet very important operations in GA are: crossover and mutation. The crossover operation combines previously selected individuals together by exchanging some of their parts. Mutation, on the other hand, brings some randomness into the search to avoid the problem of local optima. The important factors for implementing any GA are: the selection strategy and the type of crossover and mutation operators [64].

4.2.3 Particle swarm optimisation (PSO): PSO [73] is also another swarm intelligence, population-based, meta-heuristic algorithm that computationally mimics the flocking behaviour of birds to solve optimisation problems. In PSO, a swarm consists of N particles, which are stochastically generated in the search space. Each particle is represented by a velocity, a location, and has memory for remembering the best positions (solutions). PSO has succeeded at finding optimal regions of the search space. However, it has no feature that allows it to converge on optima.

4.2.4 Simulated annealing: SA [66] is a single-solution based meta-heuristic algorithm inspired by the annealing technique used to obtain a well-ordered solid state of minimal energy. The objective function of a problem in SA is then minimised by introducing the temperature parameter T, which represents the main parameter of the algorithm [64]. At each iteration, SA selects a random solution from the neighbourhood of the current solution. The new solution is accepted based on the value of the objective function and the value of the parameter T, which decreases during the search process [64].

4.2.5 Bee colony optimisation-based algorithms: Bee colony optimisation-based algorithms are new swarm-intelligence-based algorithms motivated mainly by the collective behaviour of honeybee colony [64]. Artificial bee colony (ABC) [74] is one of the well-known and commonly used foraging-inspired optimisation algorithm, which makes use of the bees' decentralised foraging behaviour. Interestingly, honey bees balance between exploiting known food sources and exploring potentially better food sources in the surrounding environment [64]. Bees in hive are divided into three sets: employed bees, onlooker bees and scouts. The number of employed bees is the same as the number of available food sources. When a food source is consumed an employed bee becomes a scout bee, which randomly searches for new food resources. Employed bees share the information about food resources with a certain probability using waggle dance [64]. ABC has a global search ability implemented through neighbourhood source production mechanism [75, 76]. In addition to ABC, many other new algorithms have been developed based on the cooperative behaviour of social honey bees [76–79]. For more details, we refer the reader to [78].

4.2.6 Whale optimisation algorithm: WOA [69] computationally mimics the social behaviour of humpback whales when hunting their prey. Humpback whales have their special hunting method, called bubble-net feeding, which is done by creating distinctive bubbles along a circle or 9-shaped path. They have two manoeuvres associated with bubble namely upward-spirals and double-loops. They dive around 12 m down and then they create bubble in a spiral shape around their prey and swim up towards the surface. Mirjalili and Lewis mathematically modelled this hunting behaviour in [69]. WOA has showed better results when compared with other meta-heuristic algorithms such as PSO [69].

4.2.7 Firefly optimisation (FFO): FFO [79] is a population-based, meta-heuristic algorithm inspired by the social (flashing) and communication behaviour of fireflies. Brighter firefly attracts other fireflies (i.e. the brightness of a firefly indicates the goodness a solution). The attractiveness exponentially decreases based on the distance between them [80, 81]. The main advantage of FFO is the fact that it uses real random numbers, and it depends on the global communication among the swarming fireflies [82]. FA, on the hand, performs a full pair-wise comparison, which might be considerably time consuming [81]. In addition to the fact that the light absorption coefficient is widely assumed to be equal to unity. However, as the distance between fireflies increases, fireflies might get trapped in their positions during the repeated evaluation of the algorithm [81].

4.2.8 Bat algorithm (BA): BA [83] is a new meta-heuristic method based on the fascinating capability of microbats to find their prey and distinguish between different types of insects even in complete darkness by decreasing the loudness and increasing the rate of emitted ultrasonic sound that bounces back from the surrounding objects [84, 85]. An artificial bat has three vectors: position vector, velocity vector and frequency vector. BA forms a balanced combination of PSO and intensive local search [84]. The balancing between these techniques is controlled by the parameters of loudness and pulse emission rate [84]. Yang [83] showed that the BA is able to outperform PSO and GA in terms of enhanced local optima avoidance and convergence speed. BA, however, could not be used to solve binary problems. Therefore, other researchers proposed a binary version of this algorithm called binary BA (BBA) [84].

4.2.9 Teaching-learning-based optimisation (TLBO): TLBO [86] is a new population-based algorithm basically proposed for constrained mechanical design optimisation problems. In TLBO, a set of learners is defined as population, and the optimal solution in the population is considered as the teacher [87]. TLBO needs merely the population size and the number of iterations as the control parameters [87]. Both the teacher and learner procedures are executed in each iteration of TLBO. When the teacher finds a better than the existing solution, it will be replaced with the new one [87].

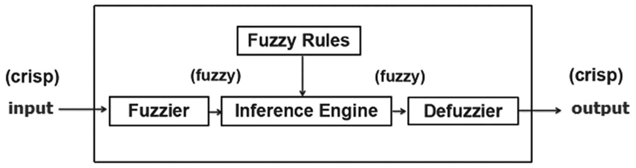

Fig. 3 *Structure of a typical fuzzy inference system*

4.2.10 Grey wolf optimisation (GWO): GWO is a population-based algorithm that computationally mimics the social hierarchy and hunting behaviour of grey wolves [88]. The first level of leaders, called alphas, responsible for making decisions about different actions that should be followed by the pack. The first level in the hierarchy of grey wolves is alphas followed by beta, delta and omega, respectively [88]. When grey wolves find a prey, they encircle it and then they start attacking the pray. Mirjalali *et al.* modelled this hunting behaviour in [88].

4.3 Fuzzy inference systems

Unlike classical binary logic where any fact can only be true or false, FL is a multi-valued logic, which deals with a degree of truth or degree of membership (i.e. any value between 0 and 1) [37]. Therefore, Boolean logic can be seen as a special case of FL. Fuzzy systems (FSs) are used for mapping a given input to an appropriate output based on the principles of fuzzy set theory, which was developed by Lotfi Zaheh [37, 89]. FS make use of fuzzy rules in the form

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } k \qquad (1)$$

where x, y and z are linguistic variables; A and B are linguistic values determined by fuzzy sets on the universe of discourses X, Y, respectively, and k is a constant that represents the consequent of the rule. As shown in Fig. 3, the system converts the crisp input to the appropriate fuzzy sets using the membership functions. Thereafter, it will be evaluated using the inference engine. Finally, the output is determined using an appropriate defuzzification method such as centre of gravity or weighted average [37].

Mamdani and Sugeno methods are two inference techniques that can be used in FSs. Unlike Mamdani's approach, which employs a fuzzy membership function of the rule consequent, Sugeno's approach uses a single point to represent the rule consequent. Therefore, Sugeno's approach is considered to be more efficient than Mamdani's approach which requires finding the centroid of a two-dimensional shape and this, in turn, increases the computational cost [37].

The main advantage of using FSs is the human-like knowledge representation and their explanation abilities. However, FSs cannot learn or adjust themselves to a new environment. In addition, building these models often requires tuning fuzzy sets and fuzzy rules to meet certain requirements. Hybird methods such as neuro-FSs combine the learning abilities of NNs with the representation and explanation abilities of FSs [37].

5 Artificial intelligence in SDN

AI and ML approaches have been widely used for solving various problems such as routing [90], traffic classification [91], flow clustering [92], intrusion detection [93], load balancing [94], fault detection [95], QoS and QoE optimisation [96], admission control and resource allocation [97]. However, in the era of SDN the role of AI was significantly increased due to the huge efforts made by industry and research community. Recent studies have shown a strong tendency of research community towards adoption of AI approaches in SDNs. It is worth mentioning that ML, meta-heuristics and FSs were the most common approaches for solving various networking related problems. We investigate the application areas of each approach in the SDN paradigm separately as will be shown in the reset of this study.

5.1 Machine learning in SDN

5.1.1 Supervised learning in SDN: In this context, NNs [97–106], SVM [105–113], DTs [114–128], ensemble methods [129–140] and supervised deep learning [44, 141–144] approaches were the most used supervised learning techniques in SDN.

NNs in SDN: NN approach was used mainly for intrusion detection and prevention [98, 102, 106], solving controller placement problem [100], load balancing [102], performance prediction [103], service level agreements (SLA) enforcement [104, 145], routing and optimal virtual machine (VM) placement [105].

Chen and Yu [98] proposed, CIPA, a collaborative intrusion prevention architecture, which represents a distributed intrusion prevention system based on NN approach. They used the following features: number of all packets monitored, proportion of icmp packets to all packets, the proportion of short packets, the proportion of long packets, the proportion of udp packets to all packets and the base-10 logarithm of the proportion of packets with syn flag set to packets with ack flag set. The experimental results showed that CIPA outperforms [146] in detecting DDoS flooding attacks. CIPA also achieved good results in detecting Witty, Slammer and Conficker worm outbreak. The system achieved low computational and communication overhead due to its parallel and simple computational capabilities.

He *et al.* [99] proposed a multi-label classification approach to predict global network allocations (i.e. weighted controller placement problem). Compared to DT and logistic regression (LR), NN approach showed better results and saved up to two-thirds of the algorithm runtime. Alvizu *et al.* [100] used a NN approach for off-line prediction of traffic demands in a mobile network operator, which resulted in reducing the optimality gap below 0.2% (virtual wavelength path-hourly-NN) and 0.45% (wavelength path-hourly-NN). In addition, NN approach was used for off-line prediction of the next configuration time point. Abubakar and Pranggono [101] proposed an intrusion detection system for SDN based on NN approach, which achieved a high accuracy of 97.3% using NSL-KDD dataset.

Chen-Xiao and Ya-Bin [102] proposed an NN approach for load balancing. Compared to [147] and static Round Robin (RR) strategy, the experimental study showed that this method can achieve better performance and resulted in 19.3% decreasing of network latency. Sabbeh *et al.* [103] proposed an NN approach based on Levenberg Marquardt algorithm to predict the performance of SDN according to round-trip time (RTT) and throughput which are the two parameters used for training. The experimental results showed that one hidden layer achieves low mean squared error (MSE).

Bendriss *et al.* [104, 145] proposed a new approach to enforce SLA in SDN and virtualised network functions (NFV). Their research focused on prediction of service level objectives breaches for streaming services running on NFV and SDN. The experimental results showed that LSTM is more robust than FFNNs.

Mestres *et al.* [105] presented a new paradigm that employs AI in SDN, termed as knowledge-defined networking. The paradigm included a proof of concept concerning routing in an overlay network based on NN approach where the MSE reached 1%. In addition to solving the problem of optimal VM placement in NFV paradigm, Mihai-Gabriel and Victor-Valeriu [106] proposed a method for mitigating DDoS attacks in SDNs by risk assessing based on NNs and biological danger theory. The risk of a DDoS attack is calculated on every host and then reported to a VM that is responsible for monitoring the network traffic. When the risk of the observed traffic exceeds a predefined value, instructions will be sent to the controller to install the appropriate controls that allow the SDN to enter in a proactive mode to reduce the burden on the controller caused by sending these flows to the controller for analysis.

SVMs in SDN: SVM approach was mainly used for deploying intrusion detection systems in the SDN paradigm [107–114]. Kokila *et al.* [107] proposed a method for the detection of DDoS attacks on the SDN controller. SVM showed higher accuracy and less false positive rate when compared to other classifiers. Phan *et* *al.* [108] proposed, OpenFlowSIA, a framework for detection of flooding attacks based on SVM and idle-timeout adjustment algorithm, which leads to an accurate system and reduction of CPU usage of Open vSwitches and their correspondence SDN controller.

On the other hand, an SVM approach for network intrusion is introduced in [109]. The model also employed the ID3 DT approach for feature selection. Based on the experimental results conducted on KDD CUP99 dataset, the system showed an accuracy of 97.60%. Boero *et al.* [110] used SVM for SDN-based malware detection where information gain (IG) metric was used for selecting the most relevant features. Their model achieved a detection rate of 80 and 95% for malware and normal traffic, respectively. In addition, it showed a false positive rate of 5.4 and 18.5% for malware and normal traffic, respectively.

Phan *et al.* [111] proposed a new approach that combines SVM with SOM in order to get a higher accuracy and better detection rate for DDoS attacks in SDN, as well as achieving lower false alarm rate. The input vector for the SOM module is a 4-tuple including four attributes: number of packet, number of byte, duration and protocol. The experimental results showed that this system was able to achieve an accuracy of 97.6% and a false positive rate of 3.85%. Shang *et al.* [112] proposed an SVM approach for implementing a traffic-based filtering classifier, which represents the second stage of a framework proposed to mitigate DoS attacks. Their model showed a high detection rate when attack rate is higher than 3000 packets per second.

Hu *et al.* [113] proposed, FADM, a real-time lightweight framework for detecting and mitigating DDoS attacks in SDN. The real-time attack detection module works as an application in the controller. The attack mitigation module consists of two main components: (i) mitigation server and (ii) mitigation agent. The mitigation server works as an application in the controller whereas the mitigation agent runs on a host in the SDN network. In the detection stage, the authors collected the following features: source IP address entropy, destination IP address entropy, source port entropy, destination port entropy and protocol type. The authors proposed to use controller-based method and sFlow-based method according to different network environments. The controller-based method (i.e. based on the incoming Packet-In messages) is efficient for low rate attacks. Whereas sFlow-based method (i.e. flow samples generated periodically from sFlow agents embedded in OF-enabled switches) is efficient for high rate attacks. FADM was implemented on the POX controller. TFN2K tool was used to launch multiple types of DDoS flooding attacks. The SVM algorithm was trained on a relatively small training dataset which consists of 552 attack samples and 662 benign samples. The attack mitigation mechanism was based on white-list and dynamic updating of forwarding rules. Compared with the controller-based method, the sFlow-based method showed higher detection rate reached 100% when the attack rate was >3000 packets per second (pps). In terms of the average detection, the difference between these methods is very small.

Latah and Toker [114] introduced a two-stage SDN-based approach for detecting DoS attacks. The detection loop consists of the following two stages: (i) calculation of packet rate on the controller side and (ii) SVM classification on the host side. When the packet rate exceeds a predefined probability, then the system will activate a host-based packet inspection unit, which collects packet-based statistics during a 4 s period and then uses the RBF-SVM algorithm in order to determine whether the attack is happened or not. The authors generated a dataset which consists of 321 instances of normal traffic and 639 instances of DoS flooding attack. Thereafter, the system was evaluated using 10-fold cross-validation and achieved 96.25% accuracy with 0.26% false alarm rate.

Rego *et al.* [115] presented an intelligence system to detect problems and correct errors in multimedia transmission in surveillance SDN-based IoT environments. Their proposed AI module consists of two different parts. The first one is the traffic classification part based on SVM algorithm, which detects the type of network traffic. The second part is an estimator that informs the SDN controller on which kind of action should be taken to guarantee the QoS and QoE. The experimental results showed that the jitter was reduced up to 70% on average and losses were reduced from 9.07% to nearly 1.16%. Furthermore, SVM was able to detect critical traffic with an accuracy of 77%.

Bouacida *et al.* [116] employed supervised approaches to detect long-term load on SDN controllers. They generated a dataset that contains nine features and 2344 instances (799 instances labelled as long-term load, 1545 instances labelled as short-term load) based on injecting Packet-In messages to the controller. The experimental results showed that the linear SVM approach achieved the best results among k-nearest neighbour (k-NN) and Naive Bayes approaches in terms of accuracy, precision, F1-measure and area under the curve (AUC). In addition, the real-time evaluation showed that both accuracy and precision values diminish proportionally when they use larger look-ahead intervals. Moreover, offloading delay increases linearly with the number of flow entries.

DTs in SDN: DTs were widely used for application identification [117, 118], packet and traffic classification [123, 126], intrusion detection [119, 129, 130], botnet detection [122, 130], solving flow table congestion [124], detection of elephant flows [127], prediction of QoS violations [128], as well as solving SDN-related security challenged [120, 125].

In [117, 118], the C4.5 DT approach was used for application identification in order to associate each application type with a corresponding QoS level. Le *et al.* [119], on the other hand, proposed an intrusion detection and prevention system based on C4.5 algorithm. Nagarathna and Shalinie [120] proposed, SLAMHHA, a supervised learning approach based on Iterative Dichotomiser 3 (ID3) DT algorithm for mitigating host location hijacking attacks on SDN controllers. This attack is a new attack vector for SDNs where it exploits the vulnerability of the SDN to launch a DoS attack on the controller. By exploiting the vulnerability of the host tracking service of the SDN controller, the host location can be estimated by examining the PACKET-IN messages. SLAMHHA consists of three main components: (i) DT construction, (ii) modified host tracking service module and (iii) classification module. When a new PACKET-IN message is received, SLAMHHA probes the message to investigate the host related information. A check is made to find whether the host related information is presented in the host profile. If no match is found, then the classification event is invoked by the SDN controller. SLAMHHA achieved less overhead in terms of CPU and memory consumption when compared to the authentication method.

Tariq and Baig [121] applied C4.5 for SDN-based botnet detection. In addition to the OF statistics they extracted four more features namely: average packet inter-arrival time, bytes per packets, bits per second and packets per second. The training phase included four botnets whereas the testing phase included five additional botnets to check the detection capability of the proposed model. The experimental results based on CTU-43 botnet dataset showed that the proposed approach achieved an accuracy of 80%.

Qazi *et al.* [122] proposed, Atlas, a framework that enables fine-grained and scalable application classification for mobile agents in SDN based on C5.0 algorithm. Atlas was prototyped in HP Lab wireless network and it collected the netstat logs using mobile agents running on employee devices. These logs are sent to the control plane, at which the classification algorithm works. The flow features were collected by an OF enabled wireless AP and sent to the control plane. The authors used 40 most popular applications in Google Play Store in order to collect over 100k flow samples from five devices during 3 weeks of the testing period. The experimental results showed that Atlas can achieve a 94% of accuracy on average.

Leng *et al.* [123] presented a method for solving flow table congestion problem based on C4.5 algorithm. C4.5 is employed to compress the flow entries with QoS guaranteed. DT is translated into a new flow table according to the path between each node and the root. The experimental results showed that the proposed approach achieved a higher compressing with larger number of flow entries and also it reduced the flow matching cost (average matching time dropped by 98% on average). Nanda *et al.* [124] proposed a method for predicting potential vulnerable hosts using historical network data and ML algorithms such as C4.5, Bayesian network (BN), decision table (DT), and naive-Bayes (NB). The best results were achieved by BN approach.

Stimpfling *et al.* [125] introduced new extensions for decision-tree algorithms, namely adaptive grouping factor and independent sub-rule leaf structure that allow achieving better packet classification for larger rules, as well as reducing the number of memory access by a factor of 3. Furthermore, it decreased the size of the data structure by about 45% over EffiCuts approach. The authors considered SDN rule-sets for this study.

Tang *et al.* [126] proposed a two-stage approach for detecting elephant flows, where the first stage consists of an efficient sampling used in order to attain a good balance between detection overhead and implementation. The next stage, on the other hand, employs an enhanced C4.5 for the previously correlated flows. The proposed approach depends mainly on finding the most effective sampling periods and classifying these samples based on enhanced C4.5. The new C4.5 is based on flow correlation and probability. It is worth mentioning that the proposed system requires only one packet to identify whether a particular flow is an elephant flow or not. The enhanced C4.5 improved the accuracy of C4.5 up to 12%.

Jain *et al.* [127] investigated the prediction of traffic congestion based on M5Rules approach, which combines DTs and linear regression in order to improve the management of QoS. They proposed a multi-dimensional analysis of key performance indicators followed by M5Rules DT to discover different types of correlations, which have been divided into three groups: expected correlations, discovered correlations and unexpected correlations. Van *et al.* [128] used a J48-tree classifier for intrusion detection on OF switches. They implemented an FPGA-based prototype where the experimental results based on KDD99 dataset showed that their proposed system can achieve an overall accuracy of 93.3% and a detection rate of 91.81% with low false alarm rates (0.55%). Wijesinghe *et al.* [129] studied the detection of botnets (IRC, HTTP and P2P botnets) using SDN paradigm based DTs approach, which showed better results for detecting peer-to-peer botnets, whereas SVM and BNs were more effective in detecting command and control (C&C) related botnets such as HTTP and IRC (internet relay chat) botnets.

Latah and Toker [130] presented a comparative analysis of the application of different supervised ML approaches for SDN-based intrusion detection task. They used principal component analysis (PCA) for feature reduction. The experimental results based on NSL-KDD dataset showed that DT approach achieved the best performance in terms of accuracy, precision, F1-measure, AUC and Mc Nemar's test. Whereas Bagging and boosting approaches achieved better results over traditional supervised ML approaches such as k-NN, NN and SVM. LogitBoost also showed the best results in terms of false alarm rate and recall.

Ensemble methods in SDN: The random forest (RF) approach was widely used for estimating service-level metrics [131], intrusion detection, security applications [133, 136–140, 148], QoE prediction [134], traffic classification [135] and prediction of traffic matrix and performance of optical path [149].

Stadler *et al.* [131] proposed a RF approach for estimating service-level metrics based on network-level statistics. Song *et al.* [132] presented an intrusion detection system where the RF approach has been used for both feature selection and classification. Miettinen *et al.* [133] introduced, IoT SENTINEL, an SDN-based system for automatic identification and security enforcement of potentially vulnerable IoT devices using device type-specific profiling approach. They used SDN for isolation and traffic filtering where the system can control the traffic flows of vulnerable devices to protect the other devices in the network from any potential threats. The device identification was achieved by fixed length fingerprints, where RF approach was used as a classification algorithm.

Abar *et al.* [134] proposed a ML-based QoE prediction approach in SDN paradigm. The authors utilised the following ML algorithms: RF, k-NN, NN and DTs where RF showed better performance. Amaral *et al.* [135] used several ML techniques such as RF, stochastic gradient boosting and extreme gradient boosting for traffic classification task. RF approach showed competitive results compared with the other two algorithms in terms of the classifier accuracy. Zago *et al.* [136] proposed an RF approach for cyber threat detection. RF approach showed the best performance among k-NN, naive Bayes and LR for cyber threat detection. Ajaeiya *et al.* [137] proposed a ML SDN-based approach for analysing, detecting and reacting against cyber threats. Experimental resulted based on ISOT botnet dataset showed that RF approach achieves the best results in terms of F1-score among k-NN, naive Bayes, bagged-trees and LR.

Anand *et al.* [138] introduced a method detecting compromised controllers in SDNs. The authors identified five threat models for representing the compromised controllers. They used nine OF-specific features in order to correctly and accurately build their ML model. The RF approach showed the highest accuracy (97%) compared to naive Bayes, SVM, NN, AdaBoost and DT. The proposed approach, however, was not able to exactly identify and locate the compromised controller when multiple physical controllers are included.

Hussein *et al.* [139] designed two architectures for building a general solution to defend and enhance the security of communication networks. The first architecture is distributed extraction, centralised processing and centralised management. The second one is distributed extraction, distributed processing and centralised management. Then the authors introduced a two-stage detection technique. The first stage includes detecting whether an attack happened or not, whereas the second stage includes identifying the type of attack. The experimental results conducted on the NSL-KDD dataset showed that the RF approach achieves better results when compared with the other techniques including SVM, kNN, DT, NNs and deep learning.

Su *et al.* [140] presented an intelligent approach to detect P2P botnets. The detection model consisted of two sub-modules namely: primary classification module and secondary classification module. The primary classification module included one binary classifier for each application or botnet. Whereas the secondary classification module employed a multi-class classification, which is used when more than one binary classifier matches in the primary classification module. The authors tested the performance of k-NN and SVM for the primary classification module. Whereas RF was used for the secondary classification module. Both of these stages were evaluated by 10-fold cross-validation. In addition, they collected network traffic samples generated by different P2P botnets and normal P2P applications. They used traffic trace files of Storm and Zeus botnet as malicious training samples of P2P botnets. Whereas network traffic trace files of eMule, uTorrent and Skype, were used as benign training samples of P2P applications. The following features were also selected: packet count, packet size, flow size, inter-arrival times (min, max, mean and standard deviation), TCP Push flag count, duration, total bytes, TCP Urgent flag count. The experimental results showed that SVM achieved better results compared to kNN. However, the authors used kNN for the primary classification, because the training time of SVM is much longer compared to kNN. In addition, RF was able to achieve an average accuracy of 99.77% for the second stage.

Chen *et al.* [148] used XGBoost classifier for DDoS attack detection in SDN-based cloud. XGBoost is an enhanced version of the traditional gradient boosting DT (GBDT) and provides a more flexible approach for preventing fitting and increasing the generalisation ability of the model. KDD Cup 99 dataset was used for training the model based on nine important features, which have the maximum IG and chi-square statistic. They used Mininet to simulate real DDoS attacks by an attack tool called Hyenae. XGBOS showed an accuracy of 98.53%. Whereas RF, GBDT and SVM showed an accuracy of 96.33, 97.69 and 97.19%, respectively. In addition, the false positive rate of XGBoost was 0.008 whereas RF, GBDT and SVM showed a false positive rate of 0.018, 0.013 and 0.011%, respectively. In terms of training time, however, the best results achieved by RF and the worst results achieved by SVM.

Choudhury *et al.* [149] introduced two applications of ML approaches for managing IP and optical networks. The first one was prediction of network traffic matrix, which allows proactive network updates for providing more flexible services. The second one is prediction of optical path performance in multi-vendor network based on the latest optical performance data. For the first task, they employed Gaussian process regression. For the second task they employed ML methods that broadly fall into three categories: penalised linear regressions, non-linear regressions and ensembles of regression trees. Among ensemble models, they applied gradient boosted regression trees and RFs. RF achieved the best overall error rates. Research efforts made to apply conventional ML approaches in SDN are briefly summarised in Table 2.

Supervised deep learning in SDN: Deep learning techniques have shown promising results in SDN compared to traditional ML approaches. Tang *et al.* [141] proposed an intrusion detection system based on a simple DNN, where deep learning approach achieved the best results compared to other supervised ML algorithms such as naive bayes, SVM and DT, with an accuracy of 75.75%. In [44], the same authors improved their results by using GRU-RNN instead of the simple deep learning model. Their new proposed model outperformed their previous simple DNN-based model [141], SVM and NB Tree with an accuracy of 89%. Lazaris and Prasanna [142] proposed, DeepFlow, an intelligent traffic measurement framework for SDNs that uses the available TCAM memory to install measurement rules for important flows, and employs LSTM-RNN to predict the size of rest of the flows when flow counters cannot be placed at a switch due to its limited resources, on the basis of historical data from previous measurement periods. The experimental results showed an average mean absolute percentage error (MAPE) of 12% for approximating real flow sizes on CAIDA traces and MAPE of 3.9% on simulating the network topology of Google's B4.

Azzouni *et al.* [150] employed a, LSTM-RNN, approach for predicting traffic matrix. GÉANT traffic matrices and related network states were the input of the LSTM network. The authors implemented a GÉANT network topology and generated 10,000 samples using Mininet. Experimental studies showed that this approach outperformed an efficient dynamic routing heuristic by finding the near-optimal path in shorter time. In the same context, Azzouni and Pujolle [143] used LSTM-RNN for the same purpose. However, in this work they validate their model based on real-world data from GÉANT backbone networks. The experimental results showed that RRN-LSTM can outperform linear forecasting models (ARMA, ARAR, HW) and FFNN.

Huang *et al.* [144] studied adversarial attacks on SDN-based deep learning port scan detection system. They applied three different adversarial attack mechanisms namely: fast gradient sign method (FGSM), Jacobian-based saliency map attack (JSMA), JSMA reverse (JSMA-RE) to three different deep learning algorithms: MLP (which is previously referred as DNN in this paper), CNN, LSTM. The deep learning models detected the port scan attacks based on Packet-In messages and STATs reports. They generated samples with three different adversarial test sets and one normal test set. JSMA showed a significant effect on the deep learning models ranges from 14 to 42%. JSMA-RE did not reduce the accuracy of CNN and LSTM, however it reduced the accuracy of MLP to 35%. FGSM caused a significant reduction in the accuracy of LSTM (more than 50%).

5.2 Unsupervised learning in SDN

K-means clustering [151–155], SOM [52, 156–160], HMM [53, 161], RBMs [162] and unsupervised deep learning approaches [54–57] were the most used unsupervised learning techniques in SDN paradigm.

K-means clustering in SDN: Ivannikova *et al.* [151] proposed a method for detecting application layer DDoS attacks in SDN-based cloud environments based on k-means and probabilistic transition. They extracted the following features at every time interval: (i) duration of the conversation, (ii) number of packets sent in 1 s, (iii) number of bytes sent in 1 s, (iv) average packet size and (v) presence of packets with different TCP flags. First, clustering is used to divide the features into different groups that represent specific classes of network traffic. Then conversations with the same source IP, destination IP and destination port at a certain time interval are grouped together. Each session in every time window is represented by a sequence of cluster labels obtained from the first step. Finally, they estimate conditional and marginal probabilities from the previously obtained sequences. Then they compare it against a corresponding threshold value, if it is lower then it is marked as anomalous. The experimental results showed that the combination of k-means and the probabilistic transition approach achieves the best results when compared with both clustering using representatives with the probabilistic transition, and n-gram with k-means, respectively.

Nguyen *et al.* [152] introduced a k-means approach for achieving Wi-Fi direct clustering in campus networks. Their simulation results showed that the proposed approach achieves better performances in terms of download time and packet error rate compared to the Wi-Fi infrastructure mode. Sahoo *et al.* [153] investigated the problem of optimal controller placement based on k-means approach. They considered two clustering algorithms namely: k-medoids and k-centre. The experimental results showed that k-centre algorithm achieves better result than K-medoids. Bakhshi and Ghita [154] used k-means for user traffic profiling in campus SDN based on their application trend. They derived six unique user traffic profiles obtained from OF statistics, which were generated by realistic campus switch during two weeks. Their proposed system showed minimum computational cost and low OF control overhead. Barki *et al.* [155] used Naive Bayes, k-NN, k-means and k-medoids for detecting DDoS attacks in SDN, where the highest detection rate achieved by naive Bayes approach. K-means, on the other hand, achieved better results over kNN and naive Bayes in terms of processing time.

SOM in SDN: SOM approach was used widely in the SDN paradigm for intrusion detection [52, 156–160]. Jankowski and Amanowicz [156] presented an intrusion detection based on SOM approach. Kohonen algorithm was used for training the SOM where the neuron whose weights are most similar to the input vector and its neighbours will update their weights. The classification step includes determining which neuron is activated under that particular input. The main disadvantage in this approach is high-grained traffic matching in flow tables. Wang and Chen [157] proposed, SGuard, a lightweight DoS attack detecting and mitigating framework based on a combination of access control and SOM classification. They collected six traffic features: percentage of flows with a small number of packets, percentage of flows with small average bytes, percentage of flows with short time duration, percentage of reversible flows, growth rate of irreversible flows and growth rate of ports. Then, they proposed two novel feature ranking and feature selecting algorithms, which allowed the SOM to achieved high detection rate with less number of features. In addition, their proposed framework can deal with IP/MAC spoofing attacks.

Phan *et al.* [52] proposed, DSOM, a distributed SOM approach for tackling the performance bottleneck and overload problems for large-sized SDNs under flooding attacks. They developed an application, called DSOMController, that manages the operation of DSOM. At each switch, DSOM was trained using a dataset obtained from the DSOMController. The DSOMController collects the results, which are SOM maps, from OF switches to construct a final merged SOM map. DSOM used the following six traffic features: number of flows, number of packet per flow, number of bytes per flow, duration, growth of client ports, protocol type. They used k-means clustering algorithm to make the final decision in DSOM. The lowest rate of accuracy was around 96.5% when they used 400 neurons, where it was increased above 97% when they used 900 and 1600 neurons. The false alarm rate also decreases with the increased number of neurons. In addition, the experimental results showed that the DSOM outperforms the single SOM in terms of the system overhead.

Braga *et al.* [158] introduced a lightweight DDoS flooding attack detection method based on SOM approach. They used six features namely: average of packets per flow, average of bytes per flow, average of duration per flow, percentage of pair-flows, growth of single-flows, growth of different ports. Their method showed high detection rate with low false positive rate. Jankowski and Amanowicz [159], on the other hand, compared different ML

Table 2 Conventional supervised ML approaches in SDN paradigm

References	Supervised ML approach	Task	Findings
Chen and Yu [98]	NN	collaborative intrusion prevention	outperformed [146]. Also it achieved a low overhead due to its parallel and simple computational capabilities
He *et al.* [99]	NN	solving weighted controller placement problem	outperformed DT and LR
Alvizu *et al.* [100]	NN	off-line prediction of traffic demands in a mobile network operator	reduced the optimality gap below 0.2% (virtual wavelength path-hourly) and 0.45% (wavelength path-hourly)
Abubakar *et al.* [101]	NN	intrusion detection	an accuracy of 97.3% using NSL-KDD dataset
Chen-Xiao and Ya-Bin [102]	NN	load balancing	compared to [147] and static RR strategy, NN achieved better performance and 19.3% decreasing of network latency
Sabbeh *et al.* [103]	NN	predicting the performance of SDN	achieved low MSE
Bendriss *et al.* [104, 145]	NN	SLA enforcement in SDN and NFV	showed less robust compared with LSTM
Mestres *et al.* [105]	NN	routing in an overlay network	MSE reached 1%
Mihai-Gabriel and Victor-Valeriu [106]	NN + biological danger theory	mitigating DDoS attacks in SDNs	proposal without simulated proof of applicability
Kokila *et al.* [107]	RBF-SVM	DDoS attack detection	an accuracy of 95.11% and false positive rate of 0.01%
Phan *et al.* [108]	multiple linear SVM	DDoS attack detection	reduction of the consumption of SDN's resources
Wang *et al.* [109]	RBF-SVM	DDoS attack detection	an accuracy of 97.60%
Boero *et al.* [110]	RBF-SVM	Malware detection	a detection rate of 80% for malware 95% for normal traffic
			false positive rate of 5.4% for malware 18.5% for normal traffic
Phan *et al.* [111]	multiple linear SVM + SOM	DDoS attack detection	an accuracy of 97.6% and false positive rate of 3.85%
FloodDefender, Shang *et al.* [112]	SVM	DoS attack detection	attack detection rate of 96% with <5% of false-positive rate
FADM Hu *et al.* [113]	SVM	DDoS attack detection	high detection rate when attack rate is higher than 3000 packets per second
Latah and Toker [114]	RBF-SVM	DoS attack detection	an accuracy of 96.25% with false positive rate of 0.26%
Rego *et al.* [115]	SVM	traffic classification	SVM was able to detect critical traffic with an accuracy of 77%
Bouacida *et al.* [116]	linear-SVM	detecting long-term load on SDNs	SVM outperformed k-NN and naive Bayes
Li *et al.* [117]	C4.5	application identification	an average accuracy of 99%
Pasca *et al.* [118]	C4.5	application identification	an accuracy of 98%. outperformed naive Bayes, naive Bayes kernel estimation, BN and SVM
Le *et al.* [119]	C4.5	intrusion detection and prevention	high precision, recall with low false positive rate
Nagarathna and Shalinie [120]	ID3	mitigating host location hijacking attacks on SDN controllers	less overhead in terms of CPI and memory consumption compared to authentication method
Tariq and Baig [121]	C4.5	botnet detection	an accuracy of 80%
Qazi *et al.* [122]	C5.0	fine-grained and scalable application classification	an average accuracy of 94%
Leng *et al.* [123]	C4.5	solving the problem of flow table congestion	high compression with large number of flow entries and reduced the flow matching cost
Nanda *et al.* [124]	C4.5	prediction of potential vulnerable hosts	outperformed NB and DT. The best results, however, achieved by BN
Stimpfling *et al.* [125]	extensions for DTs	new extensions for DTs for better packet classification and lower memory access	better packet classification for larger rules, reducing the number of memory access by a factor of 3, and decreasing the size of data structure 45% over EffiCuts
Tang *et al.* [126]	enhanced C4.5	detection of elephant flows	improve the accuracy of C4.5 up to 12%, recall rate 88.3%, false positive rate <2.13%
Jain *et al.* [127]	M5Rules	prediction of QoS violations	discover different types of correlations
Van *et al.* [128]	J48-tree	intrusion detection on OF switches	an overall accuracy of 93.3% and detection rate of 91.81% with low false alarm rates 0.55%
Wijesinghe *et al.* [129]	DT	botnet detection	DT showed better results for detecting P2P botnets whereas SVM and BNs showed effectiveness in detecting C&C botnets
Latah and Toker [130]	comparing different supervised ML algorithms	SDN-based intrusion detection	DT achieved the best level of accuracy over other supervised ML approaches. However, ensemble methods achieved the best false positive rate

Stadler *et al.* [131]	RF	estimating service-level metrics	outperformed regression tree (RT) in terms of estimation accuracy. However, RF is 3× longer than RT in terms computation time
Song *et al.* [132]	RF	intrusion detection	an accuracy of 0.99% on KDD99
Miettinen *et al.* [133]	RF	automatic identification and security enforcement for IoT devices	an accuracy of 0.815% and low execution time (<1 ms)
Abar *et al.* [134]	RF	QoE prediction	outperformed k-NN, NN and DT
Amaral *et al.* [135]	RF	traffic classification	RF achieved competitive results with the stochastic gradient boosting and extreme
Zago *et al.* [136]	RF	cyber threat detection	outperformed k-NN, naive Bayes and LR
Ajaeiya *et al.* [137]	RF	cyber threat detection	outperformed k-NN, naive Bayes, bagged-trees and LR in terms of F1-score
Anand *et al.* [138]	RF	detecting compromised controller	outperformed naive Bayes, SVM, MLP and AdaBoost
Hussein *et al.* [139]	RF	intrusion detection	outperformed SVM, k-NN, DT, NN and DNN
Su *et al.* [140]	RF	botnet detection	an average accuracy of 99.77%
Chen *et al.* [148]	XGBOS	DDoS attack detection	outperformed RF, GBDT and SVM in terms of accuracy and false positive rate
Choudhury *et al.* [149]	RF and gradient boosted regression trees	prediction of traffic matrix and performance of optical path	outperformed rigde regression, LASSO regression, LASSO with quadratic features, MLP, Guassian process regression, gradiant boosted regression trees

approaches for intrusion detection task in SDN paradigm. The experimental study showed that hierarchical learning vector quantisation (Hierarchical-LVQ1) is more efficient than other approaches such as SOM and LVQ1.

Nam *et al.* [160] used SOM for detecting DDoS attacks in SDNs. They used the following five features for the classification task: entropy of source IP address, entropy of source port, entropy of destination port, entropy of packet protocol, total number of packets. First, SOM was used for dimensionality reduction. Then, they proposed two classification techniques. The first one is based on an algorithm that combines SOM and k-NN. The second one is based on SOM with centre-distributed classification, where in this case SOM is trained only on normal traffic samples. Then, the algorithm calculates the distance between each input sample to a universal reference point where the trained data is within a hyper-sphere, centred around that reference point with a predefined threshold. If the distance is less than that particular threshold, then it will be flagged as normal otherwise it will be flagged as anomalous. The traditional k-NN achieved the highest accuracy but it showed high processing time. The first algorithm that combines SOM and k-NN outperformed the second one (SOM distributed-centre) in terms of detection rate and false positive rate. However, it showed a higher processing time compared with the second one. SOM distributed-centre showed a good detection rate with the lowest processing time, however it has very high false positive rate.

HMM in SDN: Fan *et al.* [53] investigated security situation assessment in SDNs based on an advanced HMM called multiple observations HMM. They used 12 observed features and each has two different values, where SVM is used to classify each feature value at different times into 1 or −1 values. According to the risk vector, the situation values change to show the security risk of SDN. They used Baum-Welch algorithm for training and Viterbi algorithm for predicting the network state. The experimental results showed that status of Switch compromised attack has the highest average value followed by ARP attack, OF flooding attack, scanning attack and normal situation, respectively. The prediction accuracy was 88.25%. Shan-Shan and Ya-Bin [161], on the other hand, introduced a model for detecting advanced persistent threat (APT) in SDNs based on HMM. APT is a multi-stage complex attack. Therefore, the authors proposed HMM to identify the stage of APT. Again, in this study, Baum-Welch algorithm was used for training and Viterbi algorithm for predicting the stage of APT. The experimental study showed that HMM was able to detect the stage of APT with low overhead.

RBM in SDN: MohanaPriya and Shalinie [162] studied detection of DDoS attacks in SDN based on RBM. RBM was trained using constrastive divergence algorithm. They used the following features as the input vector of RBM: source IP address, destination IP address, source port, destination port and protocol type. Their proposed model achieved a detection rate of 92% with a false positive rate of 8% based on the dataset generated by hping3 tool.

Unsupervised deep learning approaches in SDN: Mao *et al.* [54] proposed an approach for construction of routing table based on DBNs. The experimental results showed that the proposed approach outperforms traditional open shortest path first (OSPF) protocol in terms of throughout and average delay per hope. In addition, the experimental results showed that the proposed routing method can run more than 100 times faster on a GPU than on a CPU.

Zhang *et al.* [55] introduced a hybrid DNN for SDN-based network application classification, which consists of the SAE and softmax regression layer. SAE was used as unsupervised-learning-based feature extractor, whereas softmax regression was used as a supervised classifier. The proposed model achieved higher classification accuracy over SVM in terms of accuracy, precision, recall and F1-measure.

Niyaz *et al.* [56] proposed a DDoS detection system based on SAE as feature extractor in SDN. A softmax layer is also used for classification. Their proposed system can identify DDoS attacks with an accuracy of 95.65% for 8-class classification. It also achieved an accuracy of 99.82% for binary classification to show whether an attack happened or not with very low false-positive (0.3%). It also outperformed soft-max and NN classifiers. Liu *et al.* [57] SAE was used for extracting spatio-temporal features of content popularity. Then they also used softmax classifier for the classification task. Their model achieved 2.1–15 and 5.2–40% accuracy improvements over NNs and autoregressive, respectively.

Other unsupervised ML approaches in SDN: Other unsupervised ML methods were used in the SDN paradigm for intrusion detection task. In this context, He *et al.* [163] introduced an anomaly detection and mitigation method based on a two-stage unsupervised learning approach for feature selection and clustering. For the first stage, they used maximal information coefficient to calculate the relation information between two continuous features and they calculate the relevancy, a symmetric uncertainty estimator, for discrete features. In the second stage, they applied a density peak clustering algorithm on the sampled data points. In the case of a hierarchy of SDN controllers is employed, then this approach can be used to reduce the volume of traffic shuffled across the network by locally analysing the traffic data in each controller.

Ahmed *et al.* [164] proposed a method for mitigating domain name system (DNS) query-based DDoS attacks based on Dirichlet process mixture model for clustering traffic flows. The proposed system achieved better results over mean shift clustering approach.

5.2.1 RL in SDN: In this context, it is worth noting that RL [165–177] was widely applied in SDN paradigm for routing and adaptive video streaming.

Al-Jawad *et al.* [165] proposed, LearnQoS, a RL-based framework that utilises Q-learning for policy-based network management to optimise QoS requirements in multimedia-based SDNs. The RL was modelled by defining three elements: state, action and reward. The state was represented by the traffic matrix. Four different actions were considered for the agent: (i) do nothing, (ii) reduce data rate, (iii) increase data rate and (iv) reroute. The rewards, on the other hand, were based on the SLA requirements. In spite of the network overhead introduced by LearnQoS, the experimental results showed that the performance of the QoS was considerably improved when compared with the default multimedia-based SDN.

Sendra *et al.* [166] presented a routing method based on RL approach. Given a set of possible paths and a set of network measurements (delay, loss rate and bandwidth), the agent tries to maximise the reward by choosing the path with less cost. Their method showed less loss rate and better jitter values, when compared to traditional OSPF routing protocol. Lin *et al.* [167] proposed QoS-aware adaptive routing for distributed hierarchical control planes where Markov decision processes (MDPs) with QoS-aware reward function used for modelling the system. Instead of using the conventional Q-learning method, softmax action selection policy and state-action-reward-state-action approach were used for the quality update.

Kim *et al.* [168] investigated congestion prevention based on Q-learning approach. Compared with Dijkstra's algorithm and extended Dijkstra's algorithm, the proposed approach showed better results when the size of transmitted data increases. Uzakgider *et al.* [169] proposed an adaptive video streaming method based on Q-learning approach, in which the MDP was used for modelling the system. The experimental results showed that their proposed system achieves better results when compared with the shortest path routing and greedy-based approaches.

Bentaleb *et al.* [170] proposed an end-to-end SDN-based intelligent architecture for large-scale HTTP adaptive streaming (HAS) systems, where partially observable Markov decision process was used for modelling the system. Then Q-learning based approach that employs a per-cluster decision algorithm was used to maximise the QoE. The experimental study showed that their system was able to increase the video stability and achieve better QoE fairness and network resource utilisation. Jiang *et al.* [171] proposed, Q-FDBA, an on-line Q-learning-based dynamic bandwidth allocation algorithm for better QoE fairness. Q-FDBA showed better results when compared to bandwidth-aware streaming and QoE fairness framework.

Geng [172] presented MIND, which employs online version of Relative Entropy Policy Search using Reproducing Kernel Hilbert Space (REPS-RKHS) approach to learn the probability distribution of choosing the top-k best path. REPS-RKHS is a combination of Relative Entropy Policy Search (REPS) and Reproducing Kernel Hilbert Space (RKHS) embeddings in order to provide a more stable, effective learning progress for non-parametric RL. Online REPS-RKHS, on the other hand, is more appropriate for real-world problems such as routing, which require a real-time performance. MIND has a policy generation module that chooses an optimal routing policy by learning based on traffic data and network state.

Chavula *et al.* [173] used RL approach for SDN-based traffic engineering in UbuntuNet Alliance, which is the regional internetwork for National Research and Education Networks in southern and eastern Africa. The performance rewards were calculated based on distance (packet delay), available capacity on the link and the resultant load (number of flows) at the next hop. The implementation of Q-learning approach consisted of two tables: (i) a local Q-values table at each network node and (ii) a global aggregation table managed by the network controller. Each switch conducts QoS measurements on its next neighbours and passes them to the controller. The controller makes use of both active measurement data and interface-level statistics to update the Q-values based on calculating the reward values. The optimal path is selected by probabilistically choosing the forwarding link based on Q-values at each switch. They implemented the Q-learning approach using Mininet emulator to distribute traffic through multiple forwarding links in a way that maximises the throughput and reduces the latency. For multipath configurations, the best values of latency and jitter were achieved when the rewards were based on both the available link capacity and latency. On the other hand, the best throughput was achieved when the rewards were based on the links' available bandwidth. The lowest values of latency were obtained with single path forwarding, where the rewards are based on the link delays. Also, single path forwarding achieves the lowest jitter.

Francois and Gelenbe [174] proposed cognitive routing engine (CRE) for SDNs. CRE consists of three main modules. The first module is cognitive routing algorithm module (CRAM), which employs random NN with RL for finding network paths that maximise a customisable objective function to meet QoS requirements of host applications. Selection of the port that will be used as the next hop can be done in two different modes: (i) the exploratory mode and (ii) the exploitation mode. In the exploratory mode, it chooses the output port randomly, but when CPN is in the exploitation mode, it will choose the neuron which has the highest probability. Accordingly, when the reward of the new path is higher than the threshold, then the path is considered valid and the corresponding weights are updated. Otherwise, RNNs have made the wrong decision, and therefore, the weights are updated to allow other paths to be selected. The second module is network monitoring module, which obtains the network state information and notifies the CRAM. The last module is path-to-OF translator module, which converts the paths found by CRAM into the appropriate OF messages. The experiment study, which has been conducted based on Mininet emulator and GÉANT network topology showed that CRE reaches near optimal paths with 9.5 times less monitoring data than conventional SDN. However, the solutions of CRE were on average 1.65% worse than the optimal RTT.

Stampa *et al.* [175] investigated a deep RL approach for SDN-based routing optimisation based on deep deterministic policy gradients approach [176]. The state of the deep RL agent is determined by the traffic matrix, the action by a tuple of link weights and the reward is based on the mean network delay. The proposed deep RL agent outperformed their primary benchmark. Streiffer *et al.* [177], on the other hand, introduced DeepConf, a novel RL-based SDN architecture for developing and training deep ML models for automating data centre network topologies management. DeepConf consists of three components: (i) the network simulator for training Deep RL agents, (ii) an abstraction layer to facilitate communication between the DeepRL agents and the network and (iii) the DeepRL agents, which encapsulate data centre functionality. The authors used asynchronous advantage actor critic (A3C) approach which employs a deep network to approximate the policy and the value function. The agent's state space is the network topology whereas the action space for the model is represented by a vector which corresponds to different possible link combinations. The goal here is to maximise link utilisation and minimise the average flow-completion time. The learning model makes use of a CNN to compute policy decisions, one CNN-block for each state space. The authors evaluated two clos-style data centre topologies namely: Fat-tree and VL2 where the experiments showed that DeepConf was able to find near-optimal solutions across a range of topologies. In this context, the research efforts made to apply deep learning techniques in SDN are briefly summarised in Table 3.

5.2.2 Semi-supervised learning in SDN: Semi-supervised learning [178–181] was also used in SDN, but much less common compared with other learning approaches. The research was focused on traffic classification [178, 180], routing [179] and intrusion detection [180]. Wang *et al.* [178] proposed a new

Table 3 Summary for applying deep learning (DL) techniques to the SDN paradigm

References	DL approach	Task	Findings
Tang *et al.* [44]	GRU-RNN	intrusion detection	outperformed DNN [141], SVM and NB Tree with an accuracy of 89%
Mao *et al.* [54]	DBN	routing	outperformed OSPF protocol in terms of throughout and average delay per hope
Zhang *et al.* [55]	SAE	feature extraction for network application classification	outperformed SVM in terms of accuracy, precision, recall and F1-measure
Niyaz *et al.* [56]	SAE	feature extraction for DDoS detection	an accuracy of 95.65 and 99.82% for 8-class and 2-class classification, respectively. Outperformed soft-max and NN classifiers
Liu *et al.* [57]	SAE	feature extraction for content popularity prediction in information centric network	accuracy improvements over NNs and auto-regressive, 2.1–15 and 5.2–40%, respectively
Tang *et al.* [141]	DNN	intrusion detection	outperformed naive Bayes, SVM and DT with an accuracy of 75.75%
Lazaris and Prasanna [142]	LSTM-RNN	time series traffic prediction	an average MAPE of 12% for approximating real flow sizes on CAIDA traces and MAPE of 3.9% on simulating the network topology of Google's B4
Azzouni and Pujolle [150]	LSTM-RNN	traffic matrix prediction	outperformed an efficient dynamic routing heuristic by finding the near optimal path in shorter time using generated data
Azzouni and Pujolle [143]	LSTM-RNN	traffic matrix prediction	outperformed linear forecasting models (ARMA, ARAR, HW) and FFNN using real data
Huang *et al.* [144]	deep MLP (DNN), CNN and LSTM	adversarial attacks on SDN-based deep IDS	JSMA attack showed a significant impact on the deep ML models ranges from 14 to 42%, JSMA-RE reduced the accuracy of MLP to 35%. FGSM caused a significant reduction in the accuracy of LSTM (more than 50%)
Stampa *et al.* [175]	deep deterministic policy gradients	routing	outperformed their primary benchmark
Streiffer *et al.* [177]	A3C	automating data centre network topologies management	finds near optimal solutions across a range of topologies

framework for QoS-aware traffic classification based on semi-supervised learning. This approach can classify the network traffic according to the QoS requirements. The system allows achieving deep packet inspection (DPI) and semi-supervised learning based on Laplacian SVM. The Laplacian SVM approach outperformed a previous semi-supervised approach based on k-means classifier.

Chen and Zheng [179] introduced an efficient routing pre-design solution based on semi-supervised approach. The study suggested using an appropriate clustering algorithm such as Gaussian mixture model and k-means clustering for feature extraction. Thereafter, a supervised classification approach such as extreme learning machine can be used for flow demand forecasting. The authors also suggested using an adaptive multipath routing approach based on analytic hierarchy process for handling to elephant flows according to different constraint factor weights.

Li *et al.* [180] proposed a new method for fine-grained traffic classification based on semi-supervised approach called nearest application based cluster classifier. Unlike traditional methods, which use one feature vectors, this algorithm constructs a matrix with several cluster centroids based on k-means clustering to represent the application. The algorithm uses a small number of labelled flows to build a supervised dataset. Then collects the unlabelled flows to be merged with previously collected dataset, based on investigating the correlated flows, which is used to map an application to different clusters. The experimental results showed a good identification accuracy reaching 90%.

Wang *et al.* [181] introduced an intrusion detection method based on semi-supervised approach for wireless SDN-based e-health monitoring systems. The proposed system employed the concept of off-line training and on-line testing to allow running localised intrusion detection on wireless massive machine-type communications (mMTC) devices. Their system adopts semi-supervised learning on the basis of modified contrastive pessimistic likelihood estimation (CPLE), in which they replace the maximisation calculation by a relaxation function. CPLE [182] performs semi-supervised parameter estimation for likelihood-based classifiers. The proposed modified CPLE outperformed naive Bayes, SVM, DNN, self-training (semi-supervised approach) and the original CPLE based on the experiments conducted on NSL-KDD dataset.

5.3 Meta-heuristic algorithms used in SDN

A large variety of meta-heuristic algorithms such as ACO [183–191], EAs [192–194], GAs [195–203], PSO [204–209], SA [210–212], bee colony optimisation-based [213, 214], whale optimisation [215, 216], FFO [217], BA [85], TLBO [87] and GWO [207] were used in SDN.

5.3.1 ACO in SDN: ACO has been widely used for solving various networking problems such as routing [183–185, 188], load balancing [186, 187], network security [189, 190] and maximising network utilisation [191]. Dobrijevic *et al.* [183] presented an ACO approach for flow routing. The proposed model also employs QoE estimation models and attempts to maximise the user QoE for multimedia services. The experimental results showed promising QoE improvements in compared with shortest path routing.

Wang *et al.* [184] introduced two ACO-based algorithms for routing and spectrum assignment. The first one is ACO algorithm of minimum consecutiveness loss (ACO-MCL). The second one is ACO algorithm of maximum spectrum consecutiveness (ACO-MSC). The experimental results show that the proposed algorithms can reduce the blocking rate by at least 5% and perform better in spectrum efficiency.

Gao *et al.* [185] presented, CACO-RSP, a traffic engineering ACO-based approach to solve the routing rule space occupation problem for multiple unicast sessions. CACO-RSP algorithm considers both the local and global pheromone trail to guide the searching. The simulation results show that the proposed algorithm was successful in reducing the routing rule consumption.

Di Stefano *et al.* [186] proposed, A4SND, a load-balancing algorithm with low complexity and high scalability based on extending AAA algorithm, which is a modified version of ACO algorithm. A4SDN is different from the ACO-based routing algorithms in the following two points. First, the opposite interpretation of the pheromone trails. Second, the sub-path pheromone evaluation. The decision making policy in A4SND is based on pheromone laid on sub-paths, which makes it appropriate for on-line decision problems such as the routing. A4SND was emulated with the Mininet tool with two well-known variants of the Dijkstra's shortest path algorithm. The experimental results showed that A4SDN was able to outperform other two variants of Dijkstra's shortest path algorithm in terms of network throughput, end-to-end latency and packet loss.

Wang *et al.* [187] presented, LLBACO, a link load balancing algorithm based on ACO. LLBACO considers link load as main factor, whereas delay and pack-loss are the secondary factors. The experimental results showed that LLBACO has a better performance in balancing the network load and improving the QoS.

Parsaei *et al.* [188] investigated a method for providing QoS for remote telesurgery applications. The proposed approach periodically collects statistics of network state and uses ACO for computing the best path between surgeon and patient. The proposed approach improved the average end-to-end delay, packet loss ratio and peak signal-to-noise ratio (PSNR) with values of 56.3, 50.5 and 1.33%, respectively. Chen *et al.* [189] proposed an ACO-based method for detecting low-rate distributed denial of service. The proposed method consists of three stages, an information heuristic stage, a multi-agent ACO-based algorithm, and backward and forward search stages. The experimental results showed that the detection rate was more than 89% and the accuracy is greater than 83%.

Liu *et al.* [190] introduced a defence mechanism of random routing mutation in SDNs. The authors constructed an entropy matrix of network traffic characteristics and detected network anomalies. Routing mutation is triggered according to the anomaly detection results. The generation of a random routing path is formalised as a 0-1 knapsack problem, which is calculated by means of an enhanced version of ACO algorithm. The experimental results showed that the proposed approach was successful in defending against reconnaissance, eavesdropping, and DoS attacks. The main drawback of this approach was the increased burden on the controller. Yi *et al.* [191] used WiseAnt colony optimisation (WACO) algorithm to solve the bounded forwarding-rules maximum flow problem (i.e. limited resource problem in TCAM-based SDN switches). The experimental results showed that WACO can improve the network utilisation and throughput by up to 16% on the premise of a certain level of QoS.

5.3.2 EAs in SDN: The EAs were employed for solving controller placement problem in distributed SDNs [192], routing [193] and moving target defence (MTD) [194].

Zhang *et al.* [192] investigated the problem of controller placement by taking into consideration controller-to-switch and controller-to-controller delays for wide area networks (WANs), where they proposed two EAs. The first one, denoted as Evo-Place, finds a set of Pareto controller placements. The second one, denoted as Best-Reactivity, finds the final placement that minimises the average reaction time perceived at the switches. The experimental study showed that a better Pareto frontier obtained by Evo-Place compared to Rnd-Place (a basic randomised algorithm), given the same number of considered placements and the same number of iterations.

Fernandez-Fernandez *et al.* [193] presented an SDN-based routing strategy that considers both QoS requirements and energy awareness. They employed a multi-objective EA based on the strength pareto EA 2. They employed two objective functions. The first one is based on the performance requirements for the communications between the control and data planes. The second objective, related with energy awareness, aims at minimising the number of links that need to be activated when a connection request arrives. The proposed routing algorithm significantly outperformed a modified shortest path routing in terms of accepted demands.

Makanju *et al.* [194] suggested using EC techniques for MTD in SDN. MTD systems have three main challenges: (i) how to choose another configuration, (ii) adapting to next configuration and (iii) when to start the adoption. They mentioned that a multi-objective GA can be suitable for MTDs requirements.

GAs in SDN: GA were mainly used for routing [195], virtual network planning [196], optimising the cost of deploying DPI functions in SDN [197], load balancing [198–200], designing optimal observation matrices [201], resource reallocation for SDN-based data centres [202] and congestion avoidance [203].

Yu and Ke [195] introduced, GA-SDN, a GA-based routing algorithm for enhancing the video delivery quality over SDNs. The experimental results showed that GA-SDN outperformed Bellman-Ford routing algorithm in terms of packet drop rate, throughput and average PSNR. Wang *et al.* [196] studied the problem of virtual network planning in SDNs based on GA approach. The virtual network refers to customise a network topology and place the controllers that should meet given QoS requirements. Their proposed approach outperformed a greedy solution. In addition, compared to k-medoids, the proposed algorithm showed that it can reduce the number of required SDN controllers. However, k-medoids showed that it can outperform the GA-based solution in terms of the average latency.

Bouet *et al.* [197] presented a cost-based method for optimal deployment of DPI engines in NFV-SDNs. The experimental results showed that the method was able to reach a trade-off between the number of DPI engines and network load. In addition, the global cost can be reduced up to 58% when relaxing the constraint on the used link capacity. Chou *et al.* [197] introduced a policy-based load balancing system, where the authors proposed a genetic-based load balancing approach. Their system outperformed load-based, round-robin and random choice-based approaches in terms of arithmetic average for coefficient of variation.

Kang and Kwon [199] also introduced an SDN-based load balancer where the authors introduced a mapping solution as a tree structure. The authors suggested that a master (super) controller, which should be responsible for the load balancing operation, can be represented as a root node, and the other controllers can be indicated by an internal node and finally a switch will be represented by a leaf node. Mahlab *et al.* [200] proposed a fragmentation-aware load-balancing strategy for optical networks. The goal here is to optimise the fibre-load across the network. The authors employed an entropy-based metric for measuring the load imbalance and then used it for designing a joint entropy/hits utility function for the optimisation, which was solved by a GA.

Malboubi *et al.* [201, 202] presented, SNIPER, a framework for designing the optimal observation matrix that can lead to the best estimation accuracy using matrix completion techniques. The authors used GA and PSO to deal with large-scale optimal observation matrices. The experimental results showed that the proposed approach can be applied to many network monitoring applications in large-scale networks under hard resource constraints. For instance, the authors showed that by measuring only 8.8% of all per-flow path delays in Harvard network [1], congested paths can be detected with probability of 0.94.

Tajiki *et al.* [203] introduced, CECT, a congestion avoidance scheme for SDN-based cloud data centres. They used a routing architecture to reconfigure the network resources, which is triggered when a predefined time interval or when a congestion occurs. They proposed a meta-heuristic approach based on GAs to find the optimal solution for the optimisation problem. CECT can enhance the total network throughput up to 3× while it decreases the packet loss up to 2× when compared with equal cost multiple path.

5.3.3 PSO in SDN: It is used for routing [204, 205], resource management [206], multi-tenant virtual network customisation [218], multi-class routing [208] and detection of DDoS attacks [209].

Awad *et al.* [204] proposed, PSOPR, a PSO-based power-efficient routing approach for solving the problem of flow table overflow. The experimental results showed that PSOPR can achieve more than 90% of the optimal network power consumption while it requires only 0.0045–0.9% of the optimal computation time in real-network topologies obtained from SNDlib (a library of test instances for Survivable fixed telecommunication Network Design). Moreover, PSOPR resulted in shorter routes than the optimal routes obtained by the optimisation package CPLEX.

Xiong *et al.* [205] proposed, PALM, power-aware light-path management algorithm for traffic prediction in SDN-based elastic optical networks. The goal here is to avoid tearing down a light-path that becomes idle and takes into consideration reducing the switching power, which is required for re-establishment of a light-path again in short time later. They also introduced a PSO-BP neutral network model to assist the PALM algorithm in accurately predicting future traffic demands and reducing the power consumption. More precisely, a PSO-BP NN is used to predict the traffic load for every light-path within a particular time interval. PLAM achieved the power saving of 36 and 16% when compared with energy-efficient many-cast algorithm and dynamic scheduling and distance-adaptive transmission (DS + DAT) algorithm, respectively.

Chang *et al.* [206] used GA and PSO for network allocation in 5G under NFV/SDN architecture. In terms of energy consumption, GA-PSO approach achieved better results over both greedy strategy and OSFP approaches. In heavy load network environments, the proposed system can save energy nearly about 32% less than the OSPF. Their experimental results showed also a larger average link resource utilisation compared with OSPF and greedy strategy. Li *et al.* [218] introduced a PSO-based virtual SDN customisation for multi-tenant cloud services. Their experimental results showed that the PSO algorithm can significantly improve utilisation rate of the underlying network bandwidth.

Abdulqadder *et al.* [208] presented SecSDN-cloud, an integrated, secure, SDN and cloud-based architecture for providing enhanced QoS and more secure network. In SecSDN, the authors introduced an enhanced PSO multi-class routing protocol, which takes three parameters into consideration namely: node congestion, link congestion and delay. The proposed PSO-based multi-class routing protocol included a data traffic classifier per-node to segregate traffic flows into QoS classes. Then it moves to the route selection process in which a node is used to check the QoS of other nodes. Another node is selected as the one-hop neighbour for transmission, if it has a higher QoS value than the current node. SecSDN showed better results in terms of throughput when compared with OpenSec and AuthFlow. SecSDN showed better results in terms of throughput when compared with OpenSec and AuthFlow. In addition, it achieved better end-to-end delays when compared with OMC-RPL and a lower packet loss rate when compared with RPL-based SDN and FlowDefender.

Dayal and Srivastava [209] proposed a model for DDoS detection based on RBF-based NN with PSO optimisation. The author used the following features: average packets per flow, average bytes per flow, number of flows per second, average duration per flow, entropy of destination IP addresses per second, entropy of source IP address per second and entropy of IP protocol per second. The experimental results showed RBF-PSO achieved better accuracy and less training time when compared with NN-BP and NN-PSO.

5.3.4 SA in SDN: SA is used for solving controller placement problem [210, 211] and dynamic controller provisioning [212]. Hu *et al.* [210] proposed a method for solving the problem of placing controllers in SDNs in order to maximise the reliability of control networks. They proposed a novel metric that indicates the reliability of the SDN control network, called expected percentage of control. The aim of the optimisation is to minimise the expected percentage of control path loss. The simulation results showed that SA achieved the best results when compared to other methods. However, the results showed that adding a large number of controllers has an adverse effect. The reason for that is adding more additional controllers will reassemble a full mesh network with too many control paths leading to a low reliability [210, 211].

Bari *et al.* [212] used SA for deploying multiple controllers within an WAN. In terms of flow set-up time, the experimental results showed that SA can outperform both greedy knapsack (GK) and, a single controller for the entire network called 1-CRTL. However, SA required much longer time to run than GK. On the other hand, the overhead of SA is smaller than GK due to the fact that it employed fewer number of controllers and which is very close to 1-CTRL.

5.3.5 Bee colony optimisation-based algorithms in SDN: Mohammadi and Javidan [213] used ABC algorithm for traffic engineering in SDN-based video surveillance systems. The main goal of this work was to increase the quality of the received video data at monitoring server, which were streamed by cameras. The performance of the proposed method has been compared to Dijkstra routing algorithm in different scenarios. The experimental results showed that the proposed method has achieved better performance compared to Dijkstra algorithm in terms of average end-to-end delay, packet delivery ratio and PSNR.

Kang and Choo [214] proposed a system for load balancing in SDN-based cloud systems based on centralised dynamic load balancing approach, which uses the history from each cloud to reach its decision. The algorithm includes two steps: job selection and job dispatching. It adopts Best-Fit approach, which means that it will dispatch a job to the shortest average response time (ART) among available clouds. The authors compared the performance of their proposed approach to honeybee foraging algorithm and RR approaches. The experiments showed that the ART of the proposed system has a higher saturation point than the other two approaches.

5.3.6 WOA in SDN: Farshin and Sharifian [215] proposed, MAP-SDN, a framework for assignment and provisioning for SDN-based cloud data centres with different classes of service. The authors compared the performance of WOA with PSOGSA algorithm. PSOGSA combines the exploitation of PSO with exploration of gravitational search algorithm. The experimental results showed that WOA was more accurate and it can reach better results with lesser computation time.

WOA also has been used for achieving clustering-based routing for heterogeneous, randomly distributed and dense IoT networks [216]. The proposed approach consisted of two stages: set-up stage and transmission stage. At the set-up stage, the SDN controller divides the sensing area into virtual zones for balancing the number of cluster heads (CH). Then at the transmission stage the controller uses WOA for each virtual zone to determine the optimal set of CHs in that particular virtual zone. The simulation results showed that the proposed approach can minimise the power consumption compared to conventional routing protocols. In addition, the total number of packets received by the sink throughout the simulation time has improved by ∼55% compared to conventional clustering protocols, and 20% compared to the optimisation based (PSO) clustering protocol.

5.3.7 FFO in SDN: Sahoo *et al.* [217] used PSO and FireFly algorithm (FFA) for solving the controller placement problem in SDN-based WAN. They considered three metrics namely: (i) controller to switch latency, (ii) inter-controller latency and (iii) multi-path connectivity between the switch and controller. In addition, they presented average delay rise metric to measure the increased delay due to the failure of the primary path. The experimental results showed that FFA produced efficient and accurate results.

5.3.8 BA in SDN: Sathya and Thangarajan [85] employed BBA for feature selection in an SDN-based intrusion detection system. Based on the experiments conducted on NSL-KDD dataset, the selected features achieved a detection rate of 90.9, 91.1, 80.2 and 98.1% for DoS, Probe, R2L and U2R attack types, respectively, with low false positive rates when used with J48 DT.

5.3.9 TLBO in SDN: Mohammadi and Javidan [87] used TLBO for providing SDN-based QoS for remote telesurgery applications.

The proposed model calculates the optimal path between the surgeon and operating room based on network resources status. TLBO achieved the best PSNR, end-to-end delay, packet loss ratio and SSIM when compared to the shortest based approach.

5.3.10 GWO in SDN: Farshin and Sharifian [207] proposed PSO-based chaotic grey wolf optimiser algorithm for dynamic controller allocation in a SDN cloud-based 5G cellular networks. They used chaotic maps to avoid local optimum and improve the convergence speed. Due to the fact that the proposed framework consisted of several classes and each class included its own switching topology and controllers, we should use two nested chaotic GWO. The experimental results showed that chaotic GWO can achieve results that are more accurate (1% better) during fewer iterations (68% less) compared to the traditional PSO.

5.4 Fuzzy inference systems in SDN

Fuzzy inference systems were also widely used in SDN paradigm [219–224]. The main research was focused on introducing new protocols [219], intrusion detection [220–222], selection of the optimal network deployment schemes [223] and traffic engineering [224].

Abdolmaleki *et al.* [219] proposed a topology discovery protocol for software-defined wireless sensor networks. Their proposed protocol showed better performance when compared with SDN solution for wireless sensor networks [225]. Dang-Van and Truong-Thu [220] presented an approach for preventing DDoS attacks based on Sugeno's fuzzy inference method. They analysed a real network traffic to conclude DDoS indicators and thresholds. The input of the fuzzy module was the following two parameters: rate of packets having inter-arrival times in range (0–0.2 ms] and rate of flows having only one packet per flow. The experimental results showed the ability to detect and filter 97% of attack flows. In addition, it achieved a reasonably low false positive rate (∼5%) and maintained a reduction of flow entries of 50% during attack time.

Rezaei *et al.* [221] proposed a cooperative defence approach against DoS flood attacks based on fuzzy modelling. The model has two parameters, which are trust in service provider and service sensitivity. The proposed system achieved lower computational load and response time compared to SynCookie and service provider's firewall. Dotcenko *et al.* [222] proposed an information security management system based on Mamdani's fuzzy inference approach. Threat degree was determined by the fuzzy inference module, which mainly depends on threshold random walk with credit-based rate limiting (TRW-CB) and rate limiting algorithms. The proposed system was able to detect 95% of the attacks with 1.2% false positive rate.

Abdallah *et al.* [223] studied the selection of network deployment schemes based on Mamdani's approach. Their model consists of 21 rules and has the following 5 input parameters: network size, the network utilisation, network updates, initial technology deployment and ease of management. The output parameter is the network technology choice. This model can help the operators, who provide different options of networks to deploy. Mohammadi and Javidan [224] proposed a traffic engineering method for SDN-based video surveillance systems using adaptive, type-2 fuzzy approach, which showed better results when compared with type-1 fuzzy approach, available bandwidth based traffic engineering approach and, OSPF routing protocol in terms of end-to-end delay, PSNR and packet loss ratio.

5.5 Other AI-based approaches in SDN

Other researchers proposed many hybrid models that make use of two or more intelligent approach to enhance the overall performance and accuracy of these algorithms.

Petrangeli *et al.* [226] investigated preventing video freezes in HAS-based on combination of random under-sampling boosting (RUSBoost) algorithm and FL approach. The experimental results showed that the proposed system was able to reduce video freezes with 65% and freeze time with 45%, when compared with fair in-network enhanced adaptive streaming and Microsoft ISS smooth streaming.

De Assis *et al.* [227] proposed a method for mitigating DoS/DDoS attacks based on a game theatrical approach that employs Holt-Winters and GA with FL. Holt-Winters for digital signature (HWDS) approach was used for detecting and identifying anomalies and game theory decision-making (GTDM) approach was used for choosing the optimal defence strategy. HWDS was compared with fuzzy GA digital signature (Fuzzy-GADS) approach where HWDS fared better in performance tests. Fuzzy-GADS, on the other hand, found to be more efficient on detecting the occurrence of stealthier anomalies.

Li *et al.* [228] introduced an intelligent hybrid intrusion detection system for SDN-based 5G network where RF approach was used for feature selection and k-means + + with Adaboost approach was used for flow classification. The algorithm was able to achieve a good precision (94.48%), recall (92.62%) and F1-score (91.02%) with a low false positive rate (0.54%) based on 23 selected features from KDD99 dataset. Da Silva *et al.* [229] proposed an anomaly detection and classification method based on a two-phase approach: (i) lightweight phase based on entropy analysis and (ii) heavyweight approach based on a hybrid ML approach, which employs k-means and SVM. Similar flows were clustered together where each cluster represents a particular traffic profile then the SVM algorithm is used to classify the flows in each cluster. The SVM showed an accuracy of 88.7% and a precision of 82.3%.

Sabih *et al.* [230] studied optimising the performance of SDN based on hybrid intelligence approach, which makes use of a NN model. GA and PSO algorithms were used separately to select the optimal set of inputs that maximise the network efficiency. The experimental results showed that PSO achieves a better performance and a faster convergence compared with GA. Li *et al.* [231] presented an intelligent SDN-based video management system, which uses face detection and recognition techniques based on EigenFace algorithm PCA, FisherFace algorithm based on linear discriminant analysis and local binary patterns.

Yuan *et al.* [232] investigated revenue maximisation in data centres based on workload-aware SDN controller, which determines the optimal combination of a VM and routing path for each application. The proposed system uses a hybrid approach named hybrid chaotic simulated-annealing PSO that combines chaotic PSO and SA algorithm. Compared with OSPF and RR approaches, the proposed approach was successful in reducing the RTT and increasing the revenue. Huang *et al.* [233] proposed a method for application identification based on DNS responses inspection and ML. A classification system, which adopts a voting strategy, was implemented by Weak Java API, from which three algorithms were used namely: RF, rotation forest and random committee with random tree. The experimental results showed that the combination of DNS responses inspection and ML techniques was able to achieve higher accuracy compared to standalone ML techniques. Pillutla and Arjunan [234] introduced, FSOMDM, a fuzzy self-organising maps-based DDoS mitigation approach for cloud-based SDNs. FSOMDM enhances the NN approach by replacing the neurons of the traditional Kohonen NN model by updating fuzzy rules. The experimental results showed that FSOMDM achieved a true positive rate of 94%.

Aibin *et al.* [235] employed Monte Carlo Tree Search (MCTS) for advanced traffic predication for bandwidth-on-demand services and provisioning of dynamic lightpaths based on Cross-Stratum Optimisation architecture. MCTS builds a sparse search tree and chooses actions using Monte Carlo sampling approach. They compared the performance of the different approaches for selecting a data centre that can be used for providing services to a request. These approaches are: the nearest, the cheapest, the least utilised data centre, hybrid assignment (combining the three simple approaches) and traffic predication approaches based on MCTS. In the experimental studies they considered data centre models and pricing structure provided by Amazon Web Services. The hybrid assignment and traffic prediction were the two best approaches among the others. The traffic prediction approach resulted in low request blocking percentage and more efficient utilisation network resources. Hybrid approach was slightly lower than the cost of using the MCTS approach. However, it showed a higher request blocking percentage in case of high traffic loads.

Gao *et al.* [236] presented a hybrid approach that combines supervised and unsupervised approaches for defending against Packet-In based flooding attacks. Their proposed approach consisted of two stages. In the first stage, BN, which is a supervised learning algorithm, is used for classifying the potential compromised switches. In the next stage, they make use of fuzzy c-means approach to determine whether the Packet-In messages flooding attack happened or not. They conducted their experimental study based on Mininet where they used Distributed Internet Traffic Generator to generate the legitimate traffic and DARPA intrusion detection datasets were used as the malicious traffic. Based on the receiver operating characteristic curves, we can observe that the system can achieve a good accuracy, however it is combined with relatively high false alarm rate. For instance, when the detection rate is 95.68%, the false positive rate reaches 10%. The system however achieves a low overhead, as it only needs to monitor the vulnerable switches rather than all the available switches.

6 Conclusion

In this paper, we provided a state-of-the-art overview of research efforts made for applying AI techniques in SDN paradigm. Compared to our recent paper [13], this study showed an increased adoption of various AI techniques to solve a wide range of networking problems and address new challenges introduced by the SDN paradigm. This study showed that ML, meta-heuristics and FSs were the most commonly used AI approaches in SDN. In addition, it has shown that the recent advances in ML techniques such as deep learning and hybrid AI approaches can provide better results compared with traditional ML approaches. Overall, AI approaches have been proved to be very useful tools in SDNs. However, more efforts towards studying the robustness of AI approaches under adversarial settings need to be taken into consideration, as well.

7 Acknowledgment

The authors thank the anonymous reviewers for their insightful comments and constructive suggestions, which significantly helped them to improve the quality of this work.

8 References

[1] Raza, S., Huang, G., Chuah, C.N., *et al.*: 'Measurouting: a framework for routing assisted traffic monitoring', *IEEE/ACM Trans. Netw.*, 2012, **20**, (1), pp. 45–56

[2] Heorhiadi, V., Reiter, M.K., Sekar, V.: 'Simplifying software-defined network optimization using SOL'. Proc. of the 13th Usenix Conf. on Networked Systems Design and Implementation, California, USA, March 2016, pp. 223–237

[3] Martini, B., Adami, D., Sgambelluri, A., *et al.*: 'An SDN orchestrator for resources chaining in cloud data centers'. Proc. of 2014 European Conf. on Networks and Communications (EuCNC), Bologna, Italy, June 2014, pp. 1–5

[4] Akyildiz, I.F., Lee, A., Wang, P., *et al.*: 'A roadmap for traffic engineering in SDN-OpenFlow networks', *Comput. Netw.*, 2014, **71**, pp. 1–30

[5] Feamster, N., Rexford, J., Zegura, E.: 'The road to SDN: an intellectual history of programmable networks'. *ACM SIGCOMM Comput. Commun. Rev.*, 2014, **44**, (2), pp. 87–98

[6] Nunes, B.A.A., Mendonca, M., Nguyen, X.N., *et al.*: 'A survey of software-defined networking: past, present, and future of programmable networks', *IEEE Commun. Surv. Tutorials*, 2014, **16**, (3), pp. 1617–1634

[7] Casado, M., Freedman, M.J., Pettit, J., *et al.*: 'Ethane: taking control of the enterprise', *ACM SIGCOMM Comput. Commun. Rev.*, 2007, **37**, (4), pp. 1–12

[8] Jain, R., Paul, S.: 'Network virtualization and software defined networking for cloud computing: a survey', *IEEE Commun. Mag.*, 2013, **51**, (11), pp. 24–31

[9] Bera, S., Misra, S., Vasilakos, A.V.: 'Software-defined networking for internet of things: a survey', *IEEE Internet Things J.*, 2017, **4**, (6), pp. 1994–2008

[10] Lin, P., Bi, J., Wolff, S., *et al.*: 'A west-east bridge based SDN inter-domain testbed', *IEEE Commun. Mag.*, 2015, **53**, (2), pp. 190–197

[11] Jain, S., Kumar, A., Mandal, S., *et al.*: 'B4: experience with a globally-deployed software defined WAN', *ACM SIGCOMM Comput. Commun. Rev.*, 2013, **43**, (4), pp. 3–14

[12] Hong, C., Kandula, S., Mahajan, R., *et al.*: 'Achieving high utilization with software-driven WAN'. Proc. of ACM SIGCOMM Conf., Hong Kong, China, August 2013, pp. 15–26

[13] Latah, M., Toker, L.: 'Application of artificial intelligence to software defined networking: a survey', *Indian J. Sci. Technol.*, 2016, **9**, (44), pp. 1–7

[14] Sheinbein, D., Weber, R.P.: 'Stored program controlled network: 800 service using spc network capability', *Bell Labs Tech. J.*, 1982, **61**, (7), pp. 1737–1744

[15] Casado, M., Garfinkel, T., Akella, A., *et al.*: 'SANE: A protection architecture for enterprise networks'. Proc. 15th Conf. USENIX Security Symp, Vancouver, Canada, July 2006, pp. 1–15

[16] Kreutz, D., Ramos, F.M., Verissimo, P.E., *et al.*: 'Software-defined networking: a comprehensive survey', *Proc. IEEE*, 2015, **103**, (1), pp. 14–76

[17] Internet Engineering Task Force (IETF): 'Forwarding and control element separation (ForCES) protocol specification [internet]'. 2010. Available at http://www.ietf.org/rfc/rfc5810.txt, accessed 5 December 2017

[18] IETF: 'The open vswitch database management protocol [internet]'. 2013. Available at https://tools.ietf.org/html/rfc7047, accessed 5 December 2017

[19] Song, H.: 'Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane'. Proc. 2nd ACM SIGCOMM Workshop Hot Topics Software Defined Networking, Hong Kong, China, August 2013, pp. 127–132

[20] Bianchi, G., Bonola, M., Capone, A., *et al.*: 'Openstate: programming platform-independent stateful openflow applications inside the switch', *ACM SIGCOMM Comput. Commun. Rev.*, 2014, **44**, (2), pp. 44–51

[21] McKeown, N., Anderson, T., Balakrishnan, H., *et al.*: 'Openflow: enabling innovation in campus networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2008, **38**, (2), pp. 69–74

[22] IETF: 'Opflex control protocol'. [Internet]. 2014. Available at http://tools.ietf.org/html/draft-smith-opflex-00, accessed 16 October 2017

[23] Ferguson, A.D., Guha, A., Liang, C., *et al.*: 'Participatory networking: an API for application control of SDNs', *ACM SIGCOMM Comput. Commun. Rev.*, 2013, **43**, (4), pp. 327–338

[24] Voellmy, A., Kim, H., Feamster, N.: 'Procera: a language for high-level reactive network control'. Proc. of the first workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, August 2012, pp. 43–48

[25] Monsanto, C., Foster, N., Harrison, R., *et al.*: 'A compiler and run-time system for network programming languages', *ACM SIGPLAN Not.*, 2012, **47**, pp. 217–230

[26] Foster, N., Harrison, R., Freedman, M.J., *et al.*: 'Frenetic: a network programming language', *ACM SIGPLAN Not.*, 2011, **46**, (9), pp. 279–291

[27] NOX - An OpenFlow Controller [Internet]. 2008. Available at: http://www.noxrepo.org, accessed: 10 December 2017

[28] Kobayashi, M., Seetharaman, S., Parulkar, G., *et al.*: 'Maturing of OpenFlow and software-defined networking through deployments', *Comput. Netw.*, 2014, **61**, pp. 151–175

[29] Dargahi, T., Caponi, A., Ambrosin, M., *et al.*: 'A survey on the security of stateful SDN data planes. *IEEE Commun. Surv. Tutorials*, 2017, **19**, (3), pp. 1701–1725

[30] ONF. OpenFlow Specification [Internet]. 2009. Available at: https://www.opennetworking.org/software-defined-standards/specifications/ accessed 12 December 2017

[31] POX Wiki [Internet]. 2011. Available at: https://openflow.stanford.edu/display/ONL/POX+Wiki accessed 20 December 2017

[32] Beacon [Internet]. 2010. Available at: https://openflow.stanford.edu/display/Beacon/Home, accessed 20 December 2017

[33] OpenDayLight [Internet]. 2014. Available at http://www.opendaylight.org/, accessed 20 December 2017

[34] Floodlight [Internet]. 2012. Available at http://www.projectfloodlight.org/floodlight/, accessed 12 December 2017

[35] Ryu [Internet]. 2014. Available at http://osrg.github.com/ryu/, accessed 12 December 2017

[36] Russell, S., Norvig, P.: '*Artificial intelligence (a modern approach)*' (Prentice Hall, New Jersey, 1995, 3rd edn.), 1152p

[37] Negnevitsky, M.: '*Artificial intelligence - a guide to intelligent systems*' (Addison-Wesley, Essex, 2005, 2nd edn.), 415p

[38] Nguyen, T.T., Armitage, G.: 'A survey of techniques for internet traffic classification using machine learning'. *IEEE Commun. Surv. Tutor.*, 2008, **10**, (4), pp. 56–76

[39] Zhang, G.P.: 'Neural networks for classification: a survey', *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 2000, **30**, (4), pp. 451–462

[40] Rokach, L.: 'Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography', *Comput. Stat. Data Anal.*, 2008, **53**, (12), pp. 4046–4072

[41] LeCun, Y., Bengio, Y., Hinton, G.: 'Deep learning', *Nature*, 2016, **521**, (7553), pp. 436–444

[42] Deng, L.: 'A tutorial survey of architectures, algorithms, and applications for deep learning', *APSIPA Trans. Signal Inf. Process.*, 2014, **3**, (e2), pp. 1–19

[43] Mohammadi, M., Al-Fuqaha, A., Sorour, S., *et al.*: 'Deep learning for IoT big data and streaming analytics: a survey', *IEEE Commun. Surv. Tutorials*, 2018, **20**, (4), pp. 2923–2960

[44] Tang, T., Zaidi, S.A.R., McLernon, D., *et al.*: 'Deep recurrent neural network for intrusion detection in SDN-based networks'. 2018 IEEE Int. Conf. on Network Softwarization (NetSoft 2018), Montreal, Canada, June 2018

[45] Zhang, Q., Yang, L.T., Chen, Z., *et al.*: 'A survey on deep learning for big data', *Inf. Fusion*, 2018, **42**, pp. 146–157

[46] MacQueen, J.B.: 'Some methods for classification and analysis of multivariate observations'. Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability, Berkeley, USA, 1967, pp. 281–297

[47] Khan, S.S., Ahmad, A.: 'Cluster center initialization algorithm for K-means clustering', *Pattern Recognit. Lett.*, 2004, **25**, (11), pp. 1293–1302

[48] Zhang, C., Xia, S.: 'K-means clustering algorithm with improved initial center'. IEEE Second Int. Workshop on Knowledge Discovery and Data Mining, Moscow, Russia, January 2009, pp. 790–792

[49] Pal, N.R, Bezdek, J.C: 'On cluster validity for the fuzzy c-means model', *IEEE Trans. Fuzzy Syst.*, 1995, **3**, (3), pp. 370–379

[50] Kohonen, T.: 'The self-organizing map', *Neurocomputing*, 1988, **21**, (1-3), pp. 1–6

[51] Kohonen, T.: 'The self-organizing map', *Proc. IEEE*, 1990, **78**, (9), pp. 1464–1480

[52] Phan, T.V., Bao, N.K., Park, M.: 'Distributed-SOM: a novel performance bottleneck handler for large-sized software-defined networks under flooding attacks', *J. Netw. Comput. Appl.*, 2017, **91**, pp. 14–25

[53] Fan, Z., Xiao, Y., Nayak, A., *et al.*: 'An improved network security situation assessment approach in software defined networks', *Peer-to-Peer Netw. Appl.*, 2017, DOI: 10.1007/s12083-017-0604-2

[54] Mao, B., Fadlullah, Z.M., Tang, F., *et al.*: 'Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning', *IEEE Trans. Comput.*, 2017, **66**, (11), pp. 1946–1960

[55] Zhang, C., Wang, X., Li, F., *et al.*: 'Deep learning-based network application classification for SDN', *Trans. Emerg. Telecommun. Technol.*, 2018, **29**, (5), pp. 1–18

[56] Niyaz, Q., Sun, W., Javaid, A.Y.: 'A deep learning based DDoS detection system in software-defined networking (SDN)', *EAI Endorsed Trans. Sec. Safety*, 2017, **4**, (12), pp. 1–12

[57] Liu, W. X., Zhang, J., Liang, Z. W., *et al.*: 'Content popularity prediction and caching for ICN: a deep learning approach with SDN', *IEEE Access*, 2018, **6**, pp. 5075–5089

[58] Arulkumaran, K., Deisenroth, M.P., Brundage, M., *et al.*: 'Deep reinforcement learning: A brief survey', *IEEE Signal Process. Mag.*, 2017, **34**, (6), pp. 26–38

[59] Watkins, C.J., Dayan, P.: 'Q-learning', *Mach. Learn.*, 1992, **8**, (3-4), pp. 279–292

[60] Fan, X., Guo, Z.: 'A semi-supervised text classification method based on incremental EM algorithm'. WASE Int. Conf. on Information Engineering, Beidaihe, China, August, 2010, pp. 211–214

[61] Sutton, R.S., Barto, A.G.: '*Reinforcement learning: an introduction*' (MIT Press, Cambridge, MA, 2018, 2nd edn.), 548 p

[62] Kormushev, P., Calinon, S., Caldwell, D.G.: 'Reinforcement learning in robotics: applications and real-world challenges', *Robotics*, 2013, **2**, (3), pp. 122–148

[63] Bibault, J.E., Giraud, P., Burgun, A.: 'Big data and machine learning in radiation oncology: state of the art and future prospects', *Cancer Lett.*, 2016, **382**, (1), pp. 110–117

[64] Boussaid, I., Lepagnot, J., Siarry, P.: 'A survey on optimization metaheuristics', *Inf. Sci.*, 2013, **237**, pp. 82–117

[65] Yang, X.S.: 'Metaheuristic optimization: algorithm analysis and open problems', *Lect. Notes Comput. Sci.*, 2011, **6630**, pp. 21–32

[66] Kirkpatrick, S., Gellat, C.D., Vecchi, M.P.: 'Optimization by simulated annealing', *Science*, 1983, **220**, (4598), pp. 670–680

[67] Holland, J.H.: '*Adaption in natural and artificial Systems*' (University of Michigan Press, Ann Arbor, 1975), 183p

[68] Dorigo, M., Birattari, M., Stutzle, T.: 'Ant colony optimization', *IEEE Comput. Intell. Mag.*, 2006, **1**, (4), pp. 28–39

[69] Mirjalili, S., Lewis, A.: 'The whale optimization algorithm', *Adv. Eng. Softw.*, 2016, **95**, pp. 51–67

[70] Fong, S., Wang, X., Xu, Q., *et al.*: 'Recent advances in metaheuristic algorithms: does the Makara dragon exist?', *J. Supercomput.*, 2016, **72**, (10), pp. 3764–3786

[71] Jones, D.F., Mirrazavi, S.K., Tamiz, M.: 'Multi-objective meta-heuristics: an overview of the current state-of-the-art', *Eur. J. Oper. Res.*, 2002, **137**, (1), pp. 1–9

[72] Srinivas, M., Patnaik, L.M.: 'Genetic algorithms: a survey', *Computer (Long. Beach. Calif)*, 1994, **27**, (6), pp. 17–26

[73] Kennedy, J., Eberhart, R.: 'Particle swarm optimization'. Proc. of the 1995 IEEE int. Conf. on neural networks (1995), Perth, Australia, November–December 1995, pp. 1942–1948

[74] Karaboga, D.: 'An idea based on honey bee swarm for numerical optimization'.Technical ReportTR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

[75] Rao, R.S., Narasimham, S.V.L., Ramalingaraju, M.: 'Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm', *Int. J. Electr. Power Energy Syst. Eng.*, 2008, **1**, pp. 116–122

[76] Krishna, P.V.: 'Honey bee behavior inspired load balancing of tasks in cloud computing environments', *Appl. Soft Comput.*, 2013, **13**, (5), pp. 2292–2303

[77] Sadik, S., Ali, A., Ahmad, H.F., *et al.*: 'Using honey bee teamwork strategy in software agents'. CSCWD'06 10th Int. Conf. on Computer Supported Cooperative Work in Design, Nanjing, China, May 2006, pp. 1–6

[78] Karaboga, D., Akay, B.: 'A survey: algorithms simulating bee swarm intelligence', *Artif. Intell. Rev.*, 2009, **31**, pp. 61–85

[79] Yang, X. S.: '*Nature-inspired metaheuristic algorithms*' (Luniver Press, Frome, UK, 2008)

[80] Yang, X.S.: 'Firefly algorithm, Levy flights and global optimization', in Bramer, M., Ellis, R., Petridis, M. (Eds.): '*Research and development in intelligent systems XXVT*' (Springer, London, 2010), pp. 209–218

[81] Baykasoglu, A., Ozsoydan, F.B.: 'Adaptive firefly algorithm with chaos for mechanical design optimization problems', *Appl. Soft Comput.*, 2015, **36**, pp. 152–164

[82] Apostolopoulos, T., Vlachos, A.: 'Application of the firefly algorithm for solving the economic emissions load dispatch problem', *Int. J. Comb.*, 2011, **523806**, pp. 1–23

[83] Yang, X.S.: 'A new metaheuristic bat-inspired algorithm', in González, J.R., Pelta, D.A., Cruz, C., *et al.* (Eds.): '*Nature inspired cooperative strategies for optimization (NICSO 2010)*', vol. 284 (Springer, Berlin, 2010), pp. 65–74

[84] Mirjalili, S., Mirjalili, S.M., Yang, X.S.: 'Binary bat algorithm', *Neural Comput. Appl.*, 2014, **25**, (3-4), pp. 663–681

[85] Sathya, R., Thangarajan, R.: 'Efficient anomaly detection and mitigation in software defined networking environment'. Proc. of 2nd Int. Conf. on Electronics and Communication Systems (ICECS), Coimbatore, India, February 2015, pp. 479–484

[86] Rao, R.V., Savsani, V.J., Vakharia, D.P.: 'Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems', *Comput.-Aided Des.*, 2011, **43**, (3), pp. 303–315

[87] Mohammadi, R., Javidan, R.: 'On the feasibility of telesurgery over software defined networks', *Int. J. Intell. Robot. Appl.*, 2018, **2**, (3), pp. 1–12

[88] Mirjalili, S., Mirjalili, S.M., Lewis, A.: 'Grey wolf optimizer', *Adv. Eng. Softw.*, 2014, **69**, pp. 46–61

[89] Zadeh, L.A.: 'Fuzzy sets', *Inf. Control*, 1965, **8**, (3), pp. 338–353

[90] Gui, T., Ma, C., Wang, F., *et al.*: 'Survey on swarm intelligence based routing protocols for wireless sensor networks: An extensive study'. Proc. of IEEE Int. Conf. on Industrial Technology (ICIT), Taipei, Taiwan, March 2016, pp. 1944–1949

[91] Soysal, M., Schmidt, E.G.: 'Machine learning algorithms for accurate flow-based network traffic classification: evaluation and comparison', *Perform. Eval.*, 2010, **67**, (6), pp. 451–467

[92] McGregor, A., Hall, M., Lorier, P., *et al.*: 'Flow clustering using machine learning techniques'. Proc. of Int. Workshop on Passive and Active Network Measurement (PAM), Antibes Juan-les-Pins, France, April 2014, pp. 205–214

[93] Xu, X., Wang, X.: 'An adaptive network intrusion detection method based on PCA and support vector machines'. Proc. of Advanced Data Mining and Applications, Wuhan, China, July 2005, pp. 696–703

[94] Kim, H.Y., Kim, J.M.: 'A load balancing scheme based on deep-learning in IoT', *Cluster Comput.*, 2017, **20**, (1), pp. 873–878

[95] Moustapha, A.I., Selmic, R.R.: 'Wireless sensor network modeling using modified recurrent neural networks: application to fault detection', *IEEE Trans. Instrum. Meas.*, 2008, **57**, (5), pp. 981–988

[96] Mushtaq, M.S., Augustin, B., Mellouk, A.: 'Empirical study based on machine learning approach to assess the QoS/QoE correlation'. Proc. of 17th European Conf. on Networks and Optical Communications (NOC), Vilanova i la Geltru, Spain, June 2012, pp. 1–7

[97] Testolin, A., Zanforlin, M., De Grazia, M.D.F., *et al.*: 'A machine learning approach to QoE-based video admission control and resource allocation in wireless systems'. Proc. of 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), Piran, Slovenia, June 2014, pp. 31–38

[98] Chen, X.F., Yu, S.Z.: 'CIPA: A collaborative intrusion prevention architecture for programmable network and SDN', *Comput. Secur.*, 2016, **58**, pp. 1–19

[99] He, M., Kalmbach, P., Blenk, A., *et al.*: 'Algorithm-data driven optimization of adaptive communication networks'. Proc. of IEEE 25th Int. Conf. on Network Protocols (ICNP), Toronto, Canada, October 2017, pp. 1–6

[100] Alvizu, R., Troia, S., Maier, G., *et al.*: 'Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks', *IEEE/OSA J. Opt. Commun. Netw.*, 2017, **9**, (9), pp. D19–D30

[101] Abubakar, A., Pranggono, B.: 'Machine learning based intrusion detection system for software defined networks'. Proc. of Seventh Int. Conf. on Emerging Security Technologies (EST), Canterbury, UK, September 2017, pp. 138–143

[102] Chen-Xiao, C., Ya-Bin, X.: 'Research on load balance method in SDN', *Int. J. Grid Distrib. Comput.*, 2016, **9**, (1), pp. 25–36

[103] Sabbeh, A., Al-Dunainawi, Y., Al-Raweshidy, H.S., *et al.*: 'Performance prediction of software defined network using an artificial neural network'. Proc. of SAI Computing Conf. (SAI), London, UK, July 2016, pp. 80–84

[104] Bendriss, J., Yahia, I.G.B., Chemouil, P., *et al.*: 'AI for SLA management in programmable networks'. Proc. of 13th Int. Conf. on Design of Reliable Communication Networks (DRCN), Munich, Germany, March 2017, pp. 1–8

[105] Mestres, A., Rodriguez-Natal, A., Carner, J., *et al.*: 'Knowledge-defined networking', *ACM SIGCOMM Comput. Commun. Rev.*, 2017, **47**, (3), pp. 2–10

[106] Mihai-Gabriel, I., Victor-Valeriu, P.: 'Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory'. Proc. of IEEE 15th Int. Symp. on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, November 2014, pp. 319–324

[107] Kokila, R.T., Selvi, S.T., Govindarajan, K.: 'DDos detection and analysis in SDN-based environment using support vector machine classifier'. Proc. of IEEE Sixth Int. Conf. on Advanced Computing (ICoAC), Chennai, India, December 2014, pp. 205–210

[108] Phan, T.V., Van Toan, T., Van Tuyen, D., *et al.*: 'OpenFlowSIA: an optimized protection scheme for software-defined networks from flooding attacks'. Proc. of IEEE Sixth Int. Conf. on Communications and Electronics (ICCE), Ha Long, Vietnam, July 2016, pp. 13–18

[109] Wang, P., Chao, K.M., Lin, H.C., *et al.*: 'An efficient flow control approach for SDN-based network threat detection and migration using support vector machine'. Proc. of IEEE 13th Int. Conf. on e-Business Engineering (ICEBE), Macau, China, November 2016, pp. 56–63

[110] Boero, L., Marchese, M., Zappatore, S.: 'Support vector machine meets software defined networking in IDS domain'. Proc. of IEEE 29th Int. Teletraffic Congress (ITC 29), Genoa, Italy, September 2017, pp. 25–30

[111] Phan, T.V., Bao, N.K., Park, M.: 'A novel hybrid flow-based handler with DDoS attacks in software-defined networking'. Proc. of 2016 Intl IEEE Conf. on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, Toulouse, France, July 2016, pp. 350–357

[112] Shang, G., Zhe, P., Bin, X., *et al.*: 'Flooddefender: protecting data and control plane resources under SDN-aimed DoS attacks'. Proc. of IEEE Conf. on

Computer Communications (INFOCOM 2017), Atlanta, USA, May 2017, pp. 1–9
[113] Hu, D., Hong, P., Chen, Y.: 'FADM: DDoS flooding attack detection and mitigation system in software-defined networking'. GLOBECOM 2017 IEEE Global Communications Conf., Singapore, Singapore, December 2017, pp. 1–7
[114] Latah, M., Toker, M.: 'A novel intelligent approach for detecting DoS flooding attacks in software-defined networks', *Int. J. Adv. Intell. Inf.*, 2018, **4**, (1), pp. 11–20
[115] Rego, A., Canovas, A., Jimenez, J.M., *et al.*: 'An intelligent system for video surveillance in IoT environments', *IEEE Access*, 2018, **6**, pp. 31580–31598
[116] Bouacida, N., Alghadhban, A.M.J., Alalmaei, S.M.A., *et al.*: 'Failure mitigation in software defined networking employing load type prediction'. 2017 IEEE Int. Conf. on Communications (ICC), Paris, France, July, 2017, pp. 1–7
[117] Li, F., Cao, J., Wang, X., *et al.*: 'A QoS guaranteed technique for cloud applications based on software defined networking', *IEEE Access*, 2017, **5**, pp. 21229–21241
[118] Pasca, S.T.V., Kodali, S.S.P., Kataoka, K.: 'AMPS: application aware multipath flow routing using machine learning in SDN'. Proc. of IEEE Twenty-third National Conf. on Communications (NCC), Chennai, India, March 2017, pp. 1–6
[119] Le, A., Dinh, P., Le, H., *et al.*: 'Flexible network-based intrusion detection and prevention system on software-defined networks'. Proc. of IEEE Int. Conf. on Advanced Computing and Applications (ACOMP), Ho Chi Minh, Vietnam, November 2015, pp. 106–111
[120] Nagarathna, R., Shalinie, S.M.: 'SLAMHHA: A supervised learning approach to mitigate host location hijacking attack on SDN controllers'. IEEE Fourth Int. Conf. on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 16–18 March, pp. 1–7
[121] Tariq, F., Baig, S.: 'Botnet classification using centralized collection of network flow counters in software defined networks'. *Int. J. Comput. Sci. Inf. Secur.*, 2016, **14**, (8), pp. 1075–1080
[122] Qazi, Z.A., Lee, J., Jin, T., *et al.*: 'Application-awareness in SDN', *ACM SIGCOMM Comput. Commun. Rev.*, 2013, **43**, (4), pp. 487–488
[123] Leng, B., Huang, L., Qiao, C., *et al.*: 'A decision-tree-based on-line flow table compressing method in software defined networks'. Proc. of IEEE/ACM 24th Int. Symp. Quality of Service (IWQoS), Beijing, China, June 2016, pp. 1–2
[124] Nanda, S., Zafari, F., DeCusatis, C., *et al.*: 'Predicting network attack patterns in SDN using machine learning approach'. Proc. of IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, USA, November 2016, pp. 167–172
[125] Stimpfling, T., Belanger, N., Cherkaoui, O., *et al.*: 'Extensions to decision-tree based packet classification algorithms to address new classification paradigms', *Comput. Netw.*, 2017, **122**, pp. 83–95
[126] Tang, F., Li, L., Barolli, L., *et al.*: 'An efficient sampling and classification approach for flow detection in SDN-based big data centers'. Proc. of IEEE 31st Int. Conf. on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, March 2017, pp. 1106–1115
[127] Jain, S., Khandelwal, M., Katkar, A., *et al.*: 'Applying big data technologies to manage QoS in an SDN'. Proc. of 12th Int. Conf. on Network and Service Management (CNSM), Montreal, Canada, October 2016, pp. 302–306
[128] Van, N.T., Bao, H., Thinh, T.N.: 'An anomaly-based intrusion detection architecture integrated on OpenFlow switch'. Proc. of the 6th Int. Conf. on Communication and Network Security, Singapore, Singapore, November 2016, pp. 99–103
[129] Wijesinghe, U., Tupakula, U., Varadharajan, V.: 'Botnet detection using software defined networking'. Proc. of IEEE 22nd Int. Conf. on Telecommunications (ICT), Sydney, Australia, April 2015, pp. 219–224
[130] Latah, M., Toker, L.: 'Towards an efficient anomaly-based intrusion detection for software-defined networks', *IET Netw.*, 2018, **7**, (6), pp. 453–459
[131] Stadler, R., Pasquini, R., Fodor, V.: 'Learning from network device statistics', *J. Netw. Syst. Manage.*, 2017, **25**, (4), pp. 672–698
[132] Song, C., Park, Y., Golani, K., *et al.*: 'Machine-learning based threat-aware system in software defined networks'. Proc. of 26th Int. Conf. on Computer Communication and Networks (ICCCN), Vancouver, Canada, July–August 2017, pp. 1–9
[133] Miettinen, M., Marchal, S., Hafeez, I., *et al.*: 'Iot Sentinel: automated device-type identification for security enforcement in IoT'. Proc. of IEEE 37th Int. Conf. on Distributed Computing Systems (ICDCS), Atlanta, USA, June 2017, pp. 2177–2184
[134] Abar, T., Letaifa, A.B., El Asmi, S.: 'Machine learning based QoE prediction in SDN networks'. Proc. of 13th Int. Wireless Communications and Mobile Computing Conf. (IWCMC), Valencia, Spain, June 2017, pp. 1395–1400
[135] Amaral, P., Dinis, J., Pinto, P., *et al.*: 'Machine learning in software defined networks: data collection and traffic classification'. Proc. of IEEE 24th Int. Conf. on Network Protocols (ICNP), Singapore, Singapore, November 2016, pp. 1–5
[136] Zago, M., Ruiz Sanchez, V.M., Perez, M.G., *et al.*: 'Tackling cyber threats with automatic decisions and reactions based on machine-learning techniques'. Proc. of the 2nd Conf. on Network Management, Quality of Service and Security for 5G Networks, Oulu, Finland, June 2016, pp. 1–4
[137] Ajaeiya, G.A., Adalian, N., Elhajj, I.H., *et al.*: 'Flow-based intrusion detection system for SDN'. Proc. of 2017 IEEE Symp. on Computers and Communications (ISCC), Heraklion, Greece, July 2017, pp. 787–793
[138] Anand, N., Babu, S., Manoj, B.S.: 'On detecting compromised controller in software defined networks', *Comput. Netw.*, 2018, **137**, pp. 107–118
[139] Hussein, A., Chehab, A., Kayssi, A., *et al.*: 'Machine learning for network resilience: The start of a journey'. Proc. of 2018 IEEE Fifth Int. Conf. on Software Defined Systems (SDS), Barcelona, Spain, April 2018, pp. 59–66
[140] Su, S.C., Chen, Y.R., Tsai, S.C., *et al.*: 'Detecting P2P botnet in software defined networks', *Secur. Commun. Netw.*, 2018, **2018**, pp. 1–13, 4723862
[141] Tang, T.A., Mhamdi, L., McLernon, D., *et al.*: 'Deep learning approach for network intrusion detection in software defined networking'. Proc. of IEEE Int. Conf. on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, October 2016, pp. 258–263
[142] Lazaris, A., Prasanna, V. K.: 'Deepflow: a deep learning framework for software-defined measurement'. Proc. of the 2nd Workshop on Cloud-Assisted Networking, Incheon, Republic of Korea, December 2017, pp. 43–48
[143] Azzouni, A., Pujolle, G.: 'NeuTM: A neural network-based framework for traffic matrix prediction in SDN'. IEEE/IFIP Network Operations and Management Symp. NOMS 2018–2018, Taipei, Taiwan, April 2018
[144] Huang, C.H., Lee, T.H., Chang, L., *et al.*: 'Adversarial attacks on SDN-based deep learning IDS system'. Proc. of Int. Conf. on Mobile and Wireless Technology (ICMWT 2018), Hong Kong, China, June 2018, pp. 181–191
[145] Bendriss, J., Yahia, I.G.B., Zeghlache, D.: 'Forecasting and anticipating SLO breaches in programmable networks'. Proc. of 20th Conf. on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, March 2017, pp. 127–134
[146] Gamer, T.: 'Collaborative anomaly-based detection of large-scale internet attacks', *Comput. Netw.*, 2012, **56**, (1), pp. 169–185
[147] Li, Y., Pan, D.: 'Openflow based load balancing for Fat-tree networks with multipath support'. Proc. 12th IEEE Int. Conf. on Communications (ICC'13), Budapest, Hungary, June 2013, pp. 1–5
[148] Chen, Z., Jiang, F., Cheng, Y., *et al.*: 'XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud'. Proc. of IEEE Int. Conf. on Big Data and Smart Computing (BigComp), Shanghai, China, January 2018, pp. 251–256
[149] Choudhury, G., Lynch, D., Thakur, G., *et al.*: 'Two use cases of machine learning for SDN-enabled IP/optical networks: traffic matrix prediction and optical path performance prediction', *J. Opt. Commun. Netw.*, 2018, **10**, pp. D52–D62
[150] Azzouni, A., Boutaba, R., Pujolle, G.: 'Neuroute: predictive dynamic routing for software-defined networks'. 2017 13th Int. Conf. on Network and Service Management (CNSM), Tokyo, Japan, November 2017, pp. 1–6
[151] Ivannikova, E., Zolotukhin, M., Hamalainen, T.: 'Probabilistic transition-based approach for detecting application-layer DDoS attacks in encrypted software-defined networks'. Int. Conf. on Network and System Security, Helsinki, 21–23 August 2017, pp. 531–543
[152] Nguyen, T.M.T., Hamidouche, L., Mathieu, F., *et al.*: 'SDN-based Wi-Fi direct clustering for cloud access in campus networks', *Ann. Telecommun.*, 2018, **73**, (3-4), pp. 239–249
[153] Sahoo, K.S., Sahoo, S., Mishra, S.K., *et al.*: 'Analyzing controller placement in software defined networks'. IJCA Proc. on National Conf. on Next Generation Computing and its Applications in Computer Science and Technology NGCAST, Sarang, India, September 2016, pp. 1–5
[154] Bakhshi, T., Ghita, B.: 'OpenFlow-enabled user traffic profiling in campus software defined networks'. Proc. of IEEE 12th Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, USA, October 2016, pp. 1–8
[155] Barki, L., Shidling, A., Meti, N., *et al.*: 'Detection of distributed denial of service attacks in software defined networks'. IEEE Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, September, pp. 2576–2581
[156] Jankowski, D., Amanowicz, M.: 'Intrusion detection in software defined networks with self-organized maps'. *J. Telecommun. Inf. Technol.*, 2015, **4**, pp. 3–9
[157] Wang, T., Chen, H.: 'SGuard: a lightweight SDN safe-guard architecture for DoS attacks', *China Commun.*, 2017, **14**, (6), pp. 113–125
[158] Braga, R., Mota, E., Passito, A.: 'Lightweight DDoS flooding attack detection using NOX/OpenFlow'. Proc. of IEEE 35th Conf. on Local Computer Networks (LCN), Denver, USA, October 2010, pp. 408–415
[159] Jankowski, D., Amanowicz, M.: 'On efficiency of selected machine learning algorithms for intrusion detection in software defined networks', *Int. J. Electron. Telecommun.*, 2016, **62**, (3), pp. 247–252
[160] Nam, T.M., Phong, P.H., Khoa, T.D., *et al.*: 'Self-organizing map-based approaches in DDoS flooding detection using SDN'. IEEE 2018 Int. Conf. on Information Networking (ICOIN), Chiang Mai, Thailand, January 2018, pp. 249–254
[161] Shan-Shan, J., Ya-Bin, X.: 'The APT detection method in SDN'. IEEE 2017 3rd Int. Conf. on Computer and Communications (ICCC), Chengdu, China, December 2017, pp. 1240–1245
[162] MohanaPriya, P., Shalinie, S.M.: 'Restricted Boltzmann machine based detection system for DDoS attack in software defined networks'. Proc. of IEEE Fourth Int. Conf. on Signal Processing,Communication and Networking (ICSCN), Chennai, India, March 2017, pp. 1–6
[163] He, D., Chan, S., Ni, X., *et al.*: 'Software-defined-networking-enabled traffic anomaly detection and mitigation', *IEEE Internet Things J.*, 2017, **4**, (6), pp. 1890–1898
[164] Ahmed, M.E., Kim, H., Park, M.: 'Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking'. Proc. of The 36th IEEE Military Communications Conf., Baltimore, USA, October 2017, pp. 1–6
[165] Al-Jawad, A., Shah, P., Gemikonakli, O., *et al.*: 'Learnqos: a learning approach for optimizing QoS over multimedia-based SDNs'. Proc. of IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting, Valencia, Spain, June 2018
[166] Sendra, S., Rego, A., Lloret, J., *et al.*: 'Including artificial intelligence in a routing protocol using software defined networks'. Proc. of IEEE Int. Conf. on Communications Workshops (ICC Workshops), Paris, France, May 2017, pp. 670–674
[167] Lin, S.C., Akyildiz, I.F., Wang, P., *et al.*: 'QoS-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning

approach'. Proc. of IEEE Int. Conf. on Services Computing (SCC), San Francisco, USA, June 2016, pp. 25–33

[168] Kim, S., Son, J., Talukder, A., *et al.*: 'Congestion prevention mechanism based on Q-leaning for efficient routing in SDN'. Proc. of Int. Conf. on Information Networking (ICOIN), Kota Kinabalu, Malaysia, January 2016, pp. 124–128

[169] Uzakgider, T., Cetinkaya, C., Sayit, M.: 'Learning-based approach for layered adaptive video streaming over SDN', *Comput. Netw.*, 2015, **92**, (P2), pp. 357–368, doi: 10.1016/j.comnet.2015.09.027

[170] Bentaleb, A., Begen, A.C., Zimmermann, R., *et al.*: 'SDNHAS: an SDN-enabled architecture to optimize QoE in http adaptive streaming', *IEEE Trans. Multimed.*, 2017, **19**, (10), pp. 2136–2151

[171] Jiang, J., Hu, L., Hao, P., *et al.*: 'Q-FDBA: improving QoE fairness for video streaming', *Multimedia Tools Appl.*, 2018, **77**, (9), pp. 10787–10806

[172] Geng, Y.: 'MIND: A machine learning-based sdn control paradigm'. Proc. Open Networking Summit, Santa Clara, USA, March, 2016, pp. 1–16

[173] Chavula, J., Densmore, M., Suleman, H.: 'Using SDN and reinforcement learning for traffic engineering in UbuntuNet alliance'. IEEE Int. Conf. on Advances in Computing and Communication Engineering (ICACCE), Durban, South Africa, November 2016, pp. 349–355

[174] Francois, F., Gelenbe, E.: 'Towards a cognitive routing engine for software defined networks'. IEEE Int. Conf. on Communications (ICC), Kuala Lumpur, Malaysia, May 2016, pp. 1–6

[175] Stampa, G., Arias, M., Sanchez-Charles, D., *et al.*: 'A deep-reinforcement learning approach for software-defined networking routing optimization'. The 13th Int. Conf. on emerging Networking EXperiments and Technologies - Student workshop CoNEXT 2017, Incheon, South Korea, December 2017

[176] Silver, D., Lever, G., Heess, N., *et al.*: 'Deterministic policy gradient algorithms'. Proc. of the 31st Int. Conf. on Machine Learning (ICML), Beijing, China, June 2014, pp. 387–395

[177] Streiffer, C., Chen, H., Benson, T., *et al.*: 'Deepconf: automating data center network topologies management with machine learning'. Proc. of the 2018 Workshop on Network Meets AI & ML, Budapest, Hungary, August, 2018, pp. 8–14

[178] Wang, P., Lin, S.C., Luo, M.: 'A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs'. Proc. of IEEE Int. Conf. on Services Computing (SCC), San Francisco, USA, June–July 2016, pp. 760–765

[179] Chen, F., Zheng, X.: 'Machine-learning based routing pre-plan for SDN'. Int. Workshop on Multi-disciplinary Trends in Artificial Intelligence, Fuzhou, China, November 2015, pp. 149–159

[180] Li, W., Li, G., Yu, X.: 'A fast traffic classification method based on SDN network'. Proc. of the 4th Int. Conf. on Electronics, Communications and Networks (CECNET IV), Beijing, China, December 2014, pp. 223–229

[181] Wang, B., Sun, Y., Yuan, C., *et al.*: 'LESLA: A smart solution for SDN-enabled mMTC E-health monitoring system'. Proc. of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop, Los Angeles, USA, June 2018

[182] Loog, M.: 'Contrastive pessimistic likelihood estimation for semi-supervised classification', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, **38**, (3), pp. 462–475

[183] Dobrijevic, O., Santl, M., Matijasevic, M.: 'Ant colony optimization for QoE-centric flow routing in software-defined networks'. Proc. of 11th Int. Conf. on Network and Service Management (CNSM), Barcelona, Spain, November 2015, pp. 274–278

[184] Wang, F., Liu, B., Zhang, L., *et al.*: 'Dynamic routing and spectrum assignment based on multilayer virtual topology and ant colony optimization in elastic software-defined optical networks', *Opt. Eng.*, 2017, **56**, (7), p. 076111

[185] Gao, C., Wang, H., Zhai, L., *et al.*: 'Optimizing routing rules space through traffic engineering based on Ant colony algorithm in software defined network'. Proc. of IEEE 28th Int. Conf. on Tools with Artificial Intelligence (ICTAI), San Jose, USA, November 2016, pp. 106–112

[186] Di Stefano, A., Cammarata, G., Morana, G., *et al.*: 'A4sdn-adaptive alienated ant algorithm for software-defined networking'. Proc. of 10th Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, November 2015, pp. 344–350

[187] Wang, C., Zhang, G., Xu, H., *et al.*: 'An ACO-based link load-balancing algorithm in SDN'. Proc. of 7th Int. Conf. on Cloud Computing and Big Data (CCBD), Macau, China, November 2016, pp. 214–218

[188] Parsaei, M.R., Mohammadi, R., Javidan, R.: 'A new adaptive traffic engineering method for telesurgery using ACO algorithm over software defined networks', *Eur. Res. Telemed.*, 2017, **6**, (3-4), pp. 173–180

[189] Chen, H.H., Huang, S.K.: 'LDDos attack detection by using ant colony optimization algorithms', *J. Inf. Sci. Eng.*, 2016, **32**, (4), pp. 995–1020

[190] Liu, J., Zhang, H., Guo, Z.: 'A defense mechanism of random routing mutation in SDN'. *IEICE Trans. Inf. Syst.*, 2017, **100**, (5), pp. 1046–1054

[191] Yi, S., Wang, H., Yao, X., *et al.*: 'Maximizing network utilization for SDN based on WiseAnt colony optimization'. Proc. of IEEE 18th Int. Conf. on High Performance Computing and Communications; IEEE 14th Int. Conf. on Smart City; IEEE 2nd Int. Conf. on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, Australia, December 2016, pp. 959–966

[192] Zhang, T., Giaccone, P., Bianco, A., *et al.*: 'The role of the inter-controller consensus in the placement of distributed SDN controllers', *Comput. Commun.*, 2017, **113**, pp. 1–13

[193] Fernandez-Fernandez, A., Cervello-Pastor, C., Ochoa-Aday, L.: 'A multi-objective routing strategy for QoS and energy awareness in software-defined networks', *IEEE Commun. Lett.*, 2017, **21**, (11), pp. 2416–2419

[194] Makanju, A., Zincir-Heywood, A.N., Kiyomoto, S.: 'On evolutionary computation for moving target defense in software defined networks'. Proc. of the Genetic and Evolutionary Computation Conf. Companion, Berlin, Germany, July 2017, pp. 287–288

[195] Yu, Y.S., Ke, C.H.: 'Genetic algorithm-based routing method for enhanced video delivery over software defined networks', *Int. J. Commun. Syst.*, 2017, **31**, (1), p. e3391

[196] Wang, J., De Laat, C., Zhao, Z.: 'QoS-aware virtual SDN network planning'. IFIP/IEEE Symp. on Integrated Network and Service Management (IM), Lisbon, Portugal, May 2017, pp. 644–647

[197] Bouet, M., Leguay, J., Conan, V.: 'Cost-based placement of virtualized deep packet inspection functions in SDN'. Proc. of IEEE Military Communications Conf., San Diego, USA, November 2013, pp. 992–997

[198] Chou, L.D., Yang, Y.T., Hong, Y.M., *et al.*: 'A genetic-based load balancing algorithm in openflow network', in Huang, Y.M., Chao, H.C., Deng, D.J., Park, J. (Eds.): '*Advanced technologies, embedded and multimedia for human-centric computing*' (Springer, Dordrecht, 2014), pp. 411–417

[199] Kang, S.B., Kwon, G.I.: 'Load balancing strategy of SDN controller based on genetic algorithm', *Adv. Sci. Technol. Lett.*, 2016, **129**, pp. 219–222

[200] Mahlab, U., Omiyi, P.E., Hundert, H., *et al.*: 'Entropy-based load-balancing for software-defined elastic optical networks'. Proc. of 19th Int. Conf. on Transparent Optical Networks (ICTON), Girona, Spain, July 2017, pp. 1–4

[201] Malboubi, M., Gong, Y., Yang, Z., *et al.*: 'Software defined network inference with evolutionary optimal observation matrices', *Comput. Netw.*, 2017, **129**, pp. 93–104

[202] Malboubi, M., Gong, Y., Xiong, W., *et al.*: 'Software defined network inference with passive/active evolutionary-optimal probing (sniper)'. Proc. of 24th Int. Conf. on Computer Communication and Networks (ICCCN), Las Vegas, USA, August 2015, pp. 1–8

[203] Tajiki, M.M., Akbari, B., Shojafar, M., *et al.*: 'CECT: computationally efficient congestion-avoidance and traffic engineering in software-defined cloud data centers', *Cluster Comput.*, 2018, **21**, (4), pp. 1881–1897

[204] Awad, M.K., El-Shafei, M., Dimitriou, T., *et al.*: 'Power-efficient routing for SDN with discrete link rates and size-limited flow tables: a tree-based particle swarm optimization approach', *Int. J. Netw. Manage.*, 2017, **27**, (5), p. e1972

[205] Xiong, Y., Shi, J., Lv, Y., *et al.*: 'Power-aware lightpath management for SDN-based elastic optical networks'. Proc. of 26th Int. Conf. on Computer Communication and Networks (ICCCN), Vancouver, Canada, July 2017, pp. 1–9

[206] Chang, Z.X., Heng, Z., Yang, W.J.: 'Optimization of resource management for 5G', Proc. of International Conference on Computer, Electronics and Communication Engineering (CECE 2017, Sanya, China, June 2017, pp. 538–543

[207] Farshin, A., Sharifian, S.: 'A chaotic grey wolf controller allocator for software defined mobile network (SDMN) for 5th generation of cloud-based cellular systems (5G)', *Comput. Commun.*, 2017, **108**, pp. 94–109

[208] Abdulqadder, I.H., Zou, D., Aziz, I.T., *et al.*: 'SecSDN-cloud: defeating vulnerable attacks through secure software-defined networks', *IEEE Access*, 2018, **6**, pp. 8292–8301

[209] Dayal, N., Srivastava, S.: 'An RBF-PSO based approach for early detection of DDoS attacks in SDN'. Proc. of IEEE 2018 10th Int. Conf. on Communication Systems & Networks (COMSNETS), Bengaluru, India, January 2018, pp. 17–24

[210] Hu, Y., Wendong, W., Gong, X., *et al.*: 'Reliability-aware controller placement for software-defined networks'. IFIP/IEEE Int. Symp. on Integrated Network Management (IM 2013), Ghent, Belgium, May 2013, pp. 672–675

[211] Selvi, H., Guner, S., Gur, G., *et al.*: 'The controller placement problem in software defined mobile networks (SDMN)', in Liyanage, M., Gurtov, A., Ylianttila, M. (Eds.): '*Software defined mobile networks (SDMN): beyond LTE network architecture*' (Wiley, Chichester, 2015), pp. 129–147

[212] Bari, M.F., Roy, A.R., Chowdhury, S.R., *et al.*: 'Dynamic controller provisioning in software defined networks'. Proc. of 9th Int. Conf. on Network and Service Management (CNSM), Zurich, Switzerland, October 2013, pp. 18–25

[213] Mohammadi, R., Javidan, R.: 'An intelligent traffic engineering method over software defined networks for video surveillance systems based on artificial bee colony', *Int. J. Intell. Inf. Technol. (IJIIT)*, 2016, **12**, (4), pp. 45–62

[214] Kang, B., Choo, H.: 'An SDN-enhanced load-balancing technique in the cloud system', *J. Supercomput.*, 2018, **74**, (11), pp. 5706–5729

[215] Farshin, A., Sharifian, S.: 'MAP-SDN: a metaheuristic assignment and provisioning SDN framework for cloud datacenters', *J. Supercomput.*, 2017, **73**, (9), pp. 4112–4136

[216] Al-Janabi, T.A., Al-Raweshidy, H.S.: 'Efficient whale optimisation algorithm-based SDN clustering for IoT focused on node density'. Proc. of 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Budva, Montenegro, June 2017, pp. 1–6

[217] Sahoo, K.S., Puthal, D., Obaidat, M.S., *et al.*: 'On the placement of controllers in software-defined-WAN using meta-heuristic approach', *J. Syst. Softw.*, 2018, **145**, pp. 180–194

[218] Li, K., Bao, J., Lu, Z., *et al.*: 'A PSO-based virtual SDN customization for multi-tenant cloud services'. Proc. of the 11th Int. Conf. on Ubiquitous Information Management and Communication, Beppu, Japan, January 2017, pp. 1–6

[219] Abdolmaleki, N., Ahmadi, M., Malazi, H.T., *et al.*: 'Fuzzy topology discovery protocol for SDN-based wireless sensor networks', *Simul. Modelling Pract. Theory*, 2017, **79**, pp. 54–68

[220] Dang-Van, T., Truong-Thu, H.: 'A multi-criteria based software defined networking system architecture for DDoS-attack mitigation', *REV J. Electron. Commun.*, 2016, **6**, (3-4), pp. 50–60

[221] Rezaei, A.A., Mohammadifar, R., Lotfi, E., *et al.*: 'A heterogeneous defense method using fuzzy decision making'. Proc. of IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE), Naples, Italy, July 2017, pp. 1–8

[222] Dotcenko, S., Vladyko, A., Letenko, I.: 'A fuzzy logic-based information security management for software-defined networks'. Proc. of 16th Int. Conf.

on Advanced Communication Technology (ICACT), Pyeongchang, South Korea, February 2014, pp. 167–171

[223] Abdallah, S., Elhajj, I.H., Chehab, A., *et al.*: 'Fuzzy decision system for technology choice in hybrid networks'. Proc. of Fourth Int. Conf. on Software Defined Systems (SDS), Valencia, Spain, May 2017, pp. 106–111

[224] Mohammadi, R., Javidan, R.: 'An adaptive type-2 fuzzy traffic engineering method for video surveillance systems over software defined networks', *Multimedia Tools Appl.*, 2016, **76**, (22), pp. 23627–23642

[225] Galluccio, L., Milardo, S., Morabito, G., *et al.*: 'SDN-wise: design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks'. Proc. of 2015 IEEE Conf. on Computer Communications (INFOCOM), Kowloon, Hong Kong, April-May 2015, pp. 513–521

[226] Petrangeli, S., Wu, T., Wauters, T., *et al.*: 'A machine learning-based framework for preventing video freezes in HTTP adaptive streaming', *J. Netw. Comput. Appl.*, 2017, **94**, pp. 78–92

[227] De Assis, M.V., Hamamoto, A.H., Abrao, T., *et al.*: 'A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks', *IEEE Access*, 2017, **5**, pp. 9485–9496

[228] Li, J., Zhao, Z., Li, R.: 'A machine learning based intrusion detection system for software defined 5G network', *IET Netw.*, 2018, **7**, (2), pp. 53–60

[229] Da Silva, A.S., Wickboldt, J.A., Granville, L.Z., *et al.*: 'ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN'. Proc. of 2016 IEEE/IFIP Network Operations and Management Symp. (NOMS), Kowloon, China, April 2015, pp. 27–35

[230] Sabih, A., Al-Dunainawi, Y., Al-Raweshidy, H.S., *et al.*: 'Optimisation of software-defined networks performance using a hybrid intelligent system', *Adv. Sci. Technol. Eng. Syst.*, 2017, **2**, (3), pp. 617–622

[231] Li, S., Guo, Z., Liu, Y., *et al.*: 'The intelligent video management system: A use case of software defined class'. Proc. of 12th Int. Conf. on Computer Science and Education (ICCSE), Istanbul, Turkey, April 2016, pp. 149–154

[232] Yuan, H., Bi, J., Zhang, J., *et al.*: 'Workload-Aware revenue maximization in SDN-enabled data center'. IEEE 10th Int. Conf. on Cloud Computing (CLOUD), Honolulu, USA, June 2017, pp. 18–25

[233] Huang, N.F., Li, C.C., Li, C.H., *et al.*: 'Application identification system for SDN QoS based on machine learning and DNS responses'. Proc. of 19th Asia-Pacific Network Operations and Management Symp. (APNOMS), Seoul, South Korea, September 2017, pp. 407–410

[234] Pillutla, H., Arjunan, A.: 'Fuzzy self organizing maps-based DDoS mitigation mechanism for software defined networking in cloud computing', *J. Ambient. Intell. Humaniz. Comput.*, 2018, pp. 1–13, https://doi.org/10.1007/s12652-018-0754-y

[235] Aibin, M., Walkowiak, K., Haeri, S., *et al.*: 'Traffic prediction for inter-data center cross-stratum optimization problems'. 2018 IEEE Int. Conf. on Computing, Networking and Communications (ICNC), Maui, USA, March 2018, pp. 393–398

[236] Gao, D., Liu, Z., Liu, Y., *et al.*: 'Defending against packet-in messages flooding attack under SDN context', *Soft Comput.*, 2018, **22**, (20), pp. 1–13