



A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges

| | |
|--------------|----------------------------|
| ▼ Status | Idea |
| 🕒 Date Added | @September 3, 2023 3:43 PM |
| 📅 Live Date | @March 21, 2023 |
| ☰ Sponsor | |
| ▼ Tags | Information |



GOAL | Khái niệm tổng quan

SCRIPT



Summary

Main contents

I. INTRODUCTION

II. RELATED WORK

III. BACKGROUND KNOWLEDGE OF SDN

A. Architecture of SDN

B. Workflow of SDN

IV. OVERVIEW OF MACHINE LEARNING ALGORITHMS

A. Supervised Learning

- 1) k-Nearest Neighbor (k-NN)
- 2) Decision Tree (DT)
- 3) Random Forest
- 4) Neural Network (NN)
- 5) Support Vector Machine (SVM)
- 6) Bayes' Theory
- 7) Hidden Markov Models (HMM)

B. Unsupervised Learning

- 1) k-Means
- 2) Self-Organizing Map (SOM)

C. Semi-supervised Learning

D. Reinforcement Learning

- 1) Reinforcement Learning (RL)
- 2) Deep Reinforcement Learning (DRL)
- 3) RL-based Game Theory

V. MACHINE LEARNING IN SDN

A. Traffic Classification

- 1) Elephant Flow-aware Traffic Classification
- 2) Application-aware Traffic Classification
- 3) QoS-aware Traffic Classification
- 4) Analysis

B. Routing Optimization

- 1) Supervised Learning-based Routing Optimization
- 2) RL-based Routing Optimization
- 3) Traffic Prediction
- 4) Others
- 5) Analysis

C. QoS/QoE Prediction

- 1) QoS Prediction
- 2) QoE Prediction
- 3) Analysis

D. Resource Management

- 1) Data Plane Resource Management
- 2) Control Plane Resource Management
- 3) Others
- 4) Analysis

E. Security

- 1) Coarse-grained Intrusion Detection
- 2) Fine-grained Intrusion Detection
- 3) DDoS Attack Detection
- 4) Others
- 5) Analysis

VI. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

- [A. High-quality Training Datasets](#)
- [B. Distributed Multi-controller Platform](#)
- [C. Improving Network Security](#)
- [D. Cross-layer Network Optimization](#)
- [E. Incrementally Deployed SDN](#)

VII. SOME BROADER PERSPECTIVES

- [A. Software Defined Network Function Virtualization](#)
- [B. Software Defined Edge Computing](#)
- [C. Software Defined Optical Networks](#)
- [D. Software Defined Internet of Things](#)
- [E. Software Defined Vehicular Networks](#)
- [F. Software Defined Mobile Networks](#)

Vocabulary

| Aa New words | Type | Meaning | Example |
|----------------------|---------------------|--|---|
| <u>realm</u> | noun | an area of interest or activity | In addition, we review how machine learning algorithms are applied in the realm of SDN, from the perspective of traffic classification, routing optimization, Quality of Service (QoS)/Quality of Experience (QoE) prediction, resource management and security |
| <u>heterogeneous</u> | adjective formal | consisting of parts or things that are very different from each other | In recent years, with the rapid development of current Internet and mobile communication technologies, the infrastructure, devices and resources in networking systems are becoming more complex and heterogeneous |
| <u>consortium</u> | noun | an organization of several businesses or banks joining together as a group for a shared purpose: | |
| <u>optical</u> | adjective | connected with the eyes or sight, or connected with or using light | |

I. INTRODUCTION

- Transmission coverage, power and working mechanisms:
 - **Pico-cells:** các thiết bị truyền tải tín hiệu di động có phạm vi phủ sóng rất nhỏ, thường chỉ trong khoảng vài chục mét. Chúng thường được sử dụng trong mô hình mạng di động để mở rộng phủ sóng và tăng dung lượng mạng trong các khu vực có mật độ người dùng cao hoặc có nhu cầu sử dụng dịch vụ di động tại các vị trí cụ thể.
 - **Femto-cells:** Femto-cells tương tự như pico-cells, nhưng phạm vi phủ sóng của chúng thường nhỏ hơn, thường chỉ trong vài mét đến vài chục mét. Femto-cells thường được triển khai trong các mô hình mạng di

động như "Home Base Stations" và được cài đặt bởi người dùng tại nhà hoặc trong doanh nghiệp để cung cấp dịch vụ di động tại nơi cần thiết.

- **Relays:** Trong mạng di động, relays là các thiết bị hoặc nút trung gian được sử dụng để cải thiện phạm vi phủ sóng và cải thiện hiệu suất truyền dẫn. Chúng có khả năng nhận và chuyển tiếp tín hiệu từ nguồn đến đích, giúp tăng cường độ phủ sóng và giảm thiểu hiện tượng mất tín hiệu.
- **RRHs (Remote Radio Heads):** RRHs là các thiết bị được sử dụng trong mạng di động để tách biệt phần điện tử truyền tải (radio) và phần xử lý (baseband). Điều này cho phép mạng có các RRHs phân tán, giúp tối ưu hóa hiệu suất mạng và quản lý tài nguyên mạng hiệu quả.
- Communication technologies:
 - **ZigBee:** ZigBee là một giao thức mạng không dây tiêu chuẩn thấp tiêu thụ năng lượng, thường được sử dụng trong các ứng dụng Internet of Things (IoT). Nó cho phép các thiết bị IoT truyền tải dữ liệu một cách hiệu quả trong khoảng cách ngắn và sử dụng ít năng lượng.
 - **WiMAX (Worldwide Interoperability for Microwave Access):** WiMAX là một công nghệ truyền thông không dây tiên tiến, cho phép truyền dữ liệu thông qua mạng không dây với tốc độ cao. Nó thường được sử dụng trong các mạng di động không dây và mạng cố định không dây.
 - **IEEE 802.11 ac/ad:** Đây là các tiêu chuẩn giao thức mạng Wi-Fi. IEEE 802.11ac là một tiêu chuẩn Wi-Fi dựa trên sóng vô tuyến 5GHz với tốc độ truyền dữ liệu cao hơn, trong khi IEEE 802.11ad là một tiêu chuẩn Wi-Fi dựa trên sóng millimeter wave (mmWave) với tốc độ truyền dữ liệu cực nhanh, thường được sử dụng cho ứng dụng đòi hỏi băng thông lớn.
 - **LTE (Long-Term Evolution):** LTE là một công nghệ mạng di động tiên tiến được sử dụng trong mạng di động thế hệ thứ tư (4G). Nó cung cấp tốc độ truyền dữ liệu cao và hiệu suất tốt hơn cho các dịch vụ di động và truyền dẫn dữ liệu.
- Recent advances in computing technologies
 - **Graphics Processing Unit (GPU):** GPU (Graphics Processing Unit) là một loại bộ xử lý dùng để xử lý đồ họa và tính toán song song. GPU có hàng trăm hoặc thậm chí hàng nghìn lõi xử lý nhỏ, cho phép chúng thực hiện nhiều tác vụ đồng thời. Điều này làm cho GPU trở thành một công cụ mạnh mẽ cho các ứng dụng yêu cầu tính toán song song, chẳng hạn như huấn luyện mạng nơ-ron trong deep learning.
 - **Tensor Processing Unit (TPU):** một loại vi xử lý đặc biệt được thiết kế bởi Google để tối ưu hóa các phép tính dựa trên tensor, đặc biệt là trong các ứng dụng máy học sâu và trí tuệ nhân tạo (AI). TPU được tạo ra để cải thiện hiệu suất của các nhiệm vụ liên quan đến mạng nơ-ron và tính toán tensor.

II. RELATED WORK

- **Cognitive Radio Networks (CRNs):** một loại mạng không dây đặc biệt được thiết kế để cải thiện hiệu suất sử dụng tần số radio thông qua công nghệ trí tuệ nhân tạo (AI) và học máy (machine learning). Mục tiêu của CRNs là làm cho việc sử dụng tần số radio trở nên thông minh hơn và hiệu quả hơn. CRNs cho phép các thiết bị không dây có khả năng phát hiện tần số radio không sử dụng và tự động chuyển đổi sang các tần số không sử dụng để tránh gây nhiễu và tăng cường hiệu suất truyền tải. Các kỹ thuật AI và học máy được sử dụng để quản lý và tối ưu hóa việc sử dụng tần số radio trong mạng CRNs.

III. BACKGROUND KNOWLEDGE OF SDN

- Background knowledge of SDN: Perspective of architecture & workflow of SDN.

A. Architecture of SDN

- Decouple the control plane and data plane
- Network intelligence and state: logically centralized
- High-level SDN architecture composed of three main planes:
 - Data plane (lowest layer): comprised of forwarding devices including physical & virtual switches
 - Virtual: software-based, run on common operating systems ⇒ support complete features of SDN protocols
 - Physical: hardware-based, two types (implemented on open network hardware/ networking hardware vendors' merchant) ⇒ lack the flexibility and feature completeness but higher flow forwarding rate.
 - ⇒ **Purpose:** Forwarding, dropping and modifying packets based on policies received from the Control Plane (CP) through Southbound Interfaces (SBIs)
 - SBIs đề cập đến các giao diện hoặc giao thức dùng để kết nối và tương tác giữa lớp điều khiển (control plane) và lớp chuyển tiếp (forwarding plane) trong mô hình SDN.
 - **Loại SBIs:** OpenFlow, NETCONF, RESTful API, BGP-LS.
 - Control plane: program network resources, update forwarding rules dynamically, and make network administration flexible and agile.
 - Logically centralized controller: control the communication between forwarding devices and applications.
 - controller exposes and abstracts network state information of the data plane to the application plane
 - controller translates the requirements from applications into custom policies and distributes them to forwarding devices.
 - SBIs (Control-Data-Plane Interfaces (CDPIs)): allow forwarding to exchange state info & control policies, provide functions (programmatic control of all device-capability advertisements, forwarding operations, event notifications and statistics reports).
 - **OpenFlow:** truyền thông tin điều khiển từ bộ điều khiển SDN tới các thiết bị chuyển tiếp mạng (switches) thông qua giao thức này. OpenFlow cho phép kiểm soát trung tâm đối với cấu hình và quyết định về lưu trữ và chuyển tiếp dữ liệu trong mạng.
 - **OVSDB (Open vSwitch Database):** quản lý và cấu hình các thành phần của một công cụ chuyển tiếp mạng mã nguồn mở gọi là Open vSwitch (OVS). OVSDB cung cấp các dịch vụ quản lý trạng thái và cấu hình cho OVS, cho phép điều khiển linh hoạt và tùy chỉnh của OVS.
 - **ForCES (Forwarding and Control Element Separation):** tách biệt các chức năng kiểm soát và chuyển tiếp trong mạng. Nó cho phép quản lý và điều khiển các thiết bị mạng trực tiếp thông qua giao diện kiểm soát và thay đổi luồng dữ liệu thông qua giao diện chuyển tiếp.
 - **POF (Protocol-Oblivious Forwarding):** "điều khiển không quan tâm đến giao thức." Nó cho phép kiểm soát mạng tùy chỉnh quyết định cách chuyển tiếp dữ liệu dựa trên các quy tắc và chính sách cụ thể mà không bị ràng buộc bởi giao thức cụ thể.

- **NETCONF**: giao thức truyền thông dựa trên XML được sử dụng trong SBIs để cấu hình và quản lý các thiết bị mạng. Nó cung cấp các cơ chế quản lý trạng thái và cấu hình cho các thiết bị mạng.
 - **LISP (Locator/ID Separation Protocol)**: giao thức giúp tách biệt địa chỉ và vị trí trong mạng, giúp tăng cường tính linh hoạt và quản lý trong việc định tuyến và chuyển tiếp dữ liệu.
 - **OpFlex**: giao diện hướng xuống được thiết kế để làm việc với các hệ thống điều khiển mạng trong mô hình SDN, giúp điều khiển các chính sách mạng và phản ánh chúng lên các thiết bị mạng.
 - **OpenState**: giao diện hướng xuống thí nghiệm được thiết kế để hỗ trợ kiểm soát trạng thái chuyển tiếp trong mạng SDN, cho phép điều khiển linh hoạt và định tuyến dựa trên trạng thái.
- **Northbound Interfaces (NBIs)**: defined between the control plane and the application plane.
 - applications can exploit the abstract network views provided by the CP to express network behaviors and requirements, and facilitate automation, innovation and management of SDN networks.
1. **Eastbound Interfaces** (Giao diện Hướng Đông): các giao diện được sử dụng để **kết nối bộ điều khiển (controller) với các thành phần và thiết bị mạng nằm ở phía đông của bộ điều khiển**. Các thành phần ở phía đông có thể là các switch, router, hoặc các thiết bị chuyển tiếp mạng khác. Eastbound interfaces cho phép bộ điều khiển gửi các lệnh, chính sách, và quyết định đến các thiết bị mạng để điều khiển chúng.
 2. **Westbound Interfaces** (Giao diện Hướng Tây): các giao diện được sử dụng để kết nối bộ điều khiển với các ứng dụng và dịch vụ ở phía tây của mạng. Các ứng dụng SDN, các hệ thống phân tích, và các dịch vụ khác ở phía tây có thể sử dụng westbound interfaces để **truy xuất thông tin trạng thái mạng, chính sách điều khiển và thông tin liên quan từ bộ điều khiển**. Westbound interfaces cho phép các ứng dụng và dịch vụ tương tác với mạng và thực hiện các chức năng quản lý và điều khiển.
- ⇒ Sự tách biệt giữa eastbound và westbound interfaces thông thường được thực hiện với mục đích cải thiện bảo mật, kiểm soát truy cập, và đảm bảo tính tin cậy trong mạng SDN.
- **Bảo mật**: Bằng cách tách biệt các giao diện này, mạng SDN có thể duy trì mức độ bảo mật cao hơn. Eastbound interfaces cho phép bộ điều khiển tương tác với các thiết bị mạng và điều khiển chúng. Westbound interfaces cho phép ứng dụng và dịch vụ ở phía tây tương tác với bộ điều khiển. Nếu chúng có thể giao tiếp trực tiếp, có thể xuất hiện các lỗ hổng bảo mật có thể bị tấn công từ phía tây.
 - **Kiểm soát truy cập**: Tách biệt eastbound và westbound interfaces giúp kiểm soát truy cập vào hệ thống SDN. Các thiết bị mạng và bộ điều khiển có thể có quyền truy cập và quyền thực thi chính sách riêng biệt trên mỗi giao diện.
 - **Độ tin cậy**: Cách tách biệt này cũng đảm bảo tính tin cậy và ổn định của mạng SDN. Nếu các giao diện này có thể giao tiếp với nhau, có thể xảy ra các xung đột và vấn đề kỹ thuật khi các thông tin và lệnh điều khiển được truyền giữa chúng.
 - **Phân tách trách nhiệm**: Tách biệt eastbound và westbound interfaces giúp phân tách trách nhiệm giữa việc điều khiển mạng (eastbound) và việc cung cấp dịch vụ và ứng dụng (westbound). Điều này giúp tổ chức quản lý và duy trì mạng SDN một cách hiệu quả hơn.
- **Application plane (highest layer)**: composed of business applications.

- Provide new services, perform business management & optimization \Rightarrow require network state info through controllers' NBIs.
- Implement the control logic to change network behaviors

B. Workflow of SDN

- OpenFlow switch has a flow table & uses OpenFlow protocol to communicate with the SDN controller.
- Protocol: standardize the messages transmitted between switches and controller.
- Flow table: comprise of flow entries to determine the processing actions of different packets on the data plane.
 - **Data Plane** (Mặt Dữ liệu): phần của hệ thống mạng chịu trách nhiệm xử lý và chuyển tiếp các gói tin dữ liệu từ nguồn đến đích. Nó thực hiện các nhiệm vụ liên quan đến việc chuyển gói tin qua các thiết bị mạng như switch, router, hoặc các thiết bị chuyển tiếp khác. Mặt dữ liệu là nơi xảy ra việc chuyển tiếp dữ liệu và tuân theo các quy tắc, chính sách, và luồng dữ liệu đã được định nghĩa.
- Process: extract header fields & match against flow entries.
 - found matching entry: switch process packet locally according to the actions in matched flow entry.
 - not found: switch forward an OpenFlow PacketIn message to the controller
 - controller send OpenFlow FlowMod messages to [manage the switch's flow table](#) (chuyển tiếp gói tin đến bộ điều khiển) by adding flow entries \Rightarrow [process subsequent packets](#) of the flow. (thêm các mục luồng mới vào bảng luồng của switch, định rõ cách xử lý cho các gói tin của luồng này trong tương lai.)

IV. OVERVIEW OF MACHINE LEARNING ALGORITHMS

- Consist 2 main phrases: training & decision making
 - Training: learn system model using training dataset
 - Decision making: obtain estimated output
- Algorithms: 4 categories - supervised, unsupervised, semi-supervised and reinforcement learning

A. Supervised Learning

- Labelling learning technique
- Given a labeled training dataset \Rightarrow build system model representing the learned relation between input & output.
- Trained model used to get expected output

1) k-Nearest Neighbor (k-NN)

- Classification of a data sample determined based on the k nearest neighbors of that unclassified sample.

2) Decision Tree (DT)

- Classification through a learning tree.
- Each node represents a feature (attribute) of a data
- Each leaf node: class label

- Unlabeled sample classified by comparing its feature values
- Advantages: intuitive knowledge expression, simple implementation & high classification accuracy.
- Algorithms (splitting criteria): ID3 (Information gain), C4.5 (Gain ratio), CART (Gini impurity)
 - **ID3 (Information Gain):** đo lường sự giảm thiểu entropy (độ không chắc chắn) khi một nút cây được chia thành các nút con. Nó chọn thuộc tính (biến) tạo ra sự giảm thiểu lớn nhất về entropy sau khi chia - chọn thuộc tính có Information Gain cao nhất.
 - **C4.5 (Gain Ratio):** Gain Ratio cũng đo lường sự giảm thiểu entropy sau khi một nút được chia, nhưng nó còn điều chỉnh bằng cách chia cho thông tin của thuộc tính ban đầu. Điều này giúp ngăn chặn thuộc tính có nhiều giá trị (điểm dữ liệu) khỏi ảnh hưởng quá nhiều đến việc chọn thuộc tính.
 - **CART (Classification and Regression Trees) (Gini Impurity):** Gini Impurity đo lường xác suất sai phân lớp nếu một điểm dữ liệu được ngẫu nhiên chọn từ nút. Nó chọn thuộc tính dẫn đến Gini Impurity thấp nhất sau khi chia. Điều này có nghĩa rằng nó tập trung vào việc tạo ra các nút con có xác suất phân lớp đồng đều hoặc gần đồng đều nhất.

3) Random Forest

- Classification and regression tasks
- Randomly choose only a subset of feature space to construct each decision tree.
- Steps to classify new data sample: put the data sample to each tree, each tree gives classification result in tree's "vote", data sample classified into class has most votes.

4) Neural Network (NN)

- Large number of processing units: learn experiential knowledge from historical data.
- Nodes: equivalent to neurons in human brain
- Activation function: perform nonlinear computations.
- Nodes connect to each other by variable link weights.
- Input layers - hidden layers - output layers
- Complex models: change number of hidden layers and the number of nodes
- Two training types: supervised or unsupervised

a) Random NN

- Interconnected network of neurons
- Exchange excitatory and inhibitory spiking signals probabilistically (trao đổi tín hiệu kích thích và tín hiệu ức chế một cách ngẫu nhiên dựa trên xác suất)
- internal excitatory state represented by an integer ("potential")
- Potential value rises when receive excitatory spiking signal, drop when receive inhibitory spiking signal.
 - Positive: send out excitatory or inhibitory spiking signals to other according to specific neuron dependent spiking rates
 - neuron sends out a spiking signal, its potential value drops one.

b) Deep NN

- Neural networks with multiple hidden layers: process high-dimensional data and to learn increasingly complex models
- Train a feature representation based on the previous layer's output (feature hierarchy)
 - **Feature Representation (Biểu Đồ Đặc Trưng):** Đây là cách mà một mạng nơ-ron biểu diễn thông tin về dữ liệu đầu vào. Trong mạng nơ-ron, dữ liệu được truyền qua các tầng (layers) và được biểu diễn dưới dạng các đặc trưng hoặc biểu đồ đặc trưng. Mỗi tầng của mạng có khả năng trích xuất các đặc trưng cụ thể từ dữ liệu.
 - **Previous Layer's Output (Đầu Ra của Tầng Trước):** Mỗi tầng trong mạng nơ-ron nhận dữ liệu từ tầng trước đó, xử lý nó và tạo ra đầu ra. Đầu ra của một tầng sẽ là đầu vào của tầng tiếp theo trong mạng.
 - **Feature Hierarchy (Hệ Thống Đặc Trưng Cấp Độ):** Trong quá trình xây dựng mạng nơ-ron sâu, thông tin thường được biểu diễn dưới dạng một hệ thống đặc trưng cấp độ, nghĩa là các đặc trưng trừu tượng và trừu tượng hơn được tạo ra từ các đặc trưng cơ bản. Điều này tạo ra một cấu trúc phân cấp của các đặc trưng.

c) Convolutional NN

- Feed-forward neural network
- Successive layers, weight sharing and pooling
 - **Local Sparse Connections (Kết Nối Rải Rác Cục Bộ):** Mạng nơ-ron tích chập thường xây dựng các kết nối giữa các tầng (layers) một cách rải rác và cục bộ. Điều này có nghĩa là mỗi nơ-ron ở tầng trước sẽ chỉ kết nối với một số nơ-ron cụ thể ở tầng tiếp theo thay vì kết nối với tất cả nơ-ron. Các kết nối này giúp mạng tập trung vào các phần quan trọng của dữ liệu và giảm độ phức tạp của mạng.
 - **Weight Sharing (Chia Sẻ Trọng Số):** Trong mạng tích chập, các trọng số của các kết nối cục bộ được chia sẻ giữa các vị trí khác nhau trong dữ liệu đầu vào. Điều này có nghĩa là cùng một bộ trọng số được sử dụng để xử lý các phần khác nhau của dữ liệu. Chia sẻ trọng số giúp giảm lượng tham số của mạng và tạo ra tính toán hiệu quả hơn.
 - **Pooling (Lọc Động):** Pooling là một phần quan trọng trong CNN và thường được sử dụng sau khi các tầng tích chập. Pooling giúp giảm kích thước của biểu đồ đặc trưng (feature map) và tạo ra tính bất biến đối với dịch chuyển của đối tượng trong dữ liệu. Pooling thường là các phép tổng hợp (ví dụ: pooling max) hoặc trung bình (ví dụ: pooling trung bình) được thực hiện trên một khu vực nhỏ của biểu đồ đặc trưng. Điều này giúp tạo ra một biểu đồ đặc trưng với độ phân giải thấp hơn nhưng vẫn giữ lại các đặc điểm quan trọng.

d) Recurrent NN

- Stateful network, use internal state (memory) to handle sequential data
- **Long Short-Term Memory (LSTM):** LSTM là một biến thể phổ biến của RNN được thiết kế để giải quyết vấn đề biến mất gradient (vanishing gradient problem) và có khả năng nắm bắt các phụ thuộc dài hạn trong chuỗi dữ liệu. LSTM sử dụng ba cổng (input gate, output gate, forget gate) để tính toán trạng thái ẩn, giúp nó biểu diễn thông tin chuỗi một cách hiệu quả hơn.
 1. **Giải quyết Vấn đề Biến mất Gradient:** Trong việc huấn luyện mạng nơ-ron hồi quy truyền thống, có một vấn đề gọi là "biến mất gradient," nghĩa là các gradient của hàm mất mát truyền ngược (backpropagation) có thể trở nên rất nhỏ và làm cho quá trình học trở nên khó khăn khi chuỗi dữ liệu dài. LSTM sử dụng một cơ chế đặc biệt để duy trì gradient và tránh tình trạng biến mất gradient, cho phép nó học được các phụ thuộc dài hạn trong chuỗi dữ liệu.
 2. **Ba Cổng Quan trọng:**

- **Input Gate (Cổng Đầu Vào):** Cổng này quyết định thông tin nào từ đầu vào nên được lưu vào trạng thái ẩn. Nó điều chỉnh quá trình cập nhật trạng thái ẩn.
 - **Output Gate (Cổng Đầu Ra):** Cổng này quyết định trạng thái ẩn cần phải đưa ra dự đoán tại thời điểm hiện tại. Nó điều chỉnh cách chúng ta sử dụng thông tin trong trạng thái ẩn để tạo ra đầu ra.
 - **Forget Gate (Cổng Quên):** Cổng này quyết định thông tin nào trong trạng thái ẩn cần phải "quên" hoặc bỏ qua. Nó giúp mô hình xử lý các chuỗi dữ liệu dài mà không bị quá tải thông tin không cần thiết.
- X_t : Đây là đầu vào tại thời điểm t . Mỗi thời điểm tương ứng với một phần của chuỗi dữ liệu đầu vào.
 - h_t : Đây là trạng thái ẩn (hidden state) tại thời điểm t . Trạng thái ẩn chứa thông tin về những gì đã xảy ra trong tất cả các thời điểm trước đó, được gọi là "bộ nhớ" (memory).
 - Y_t : Đây là đầu ra tại thời điểm t . Nó thể hiện kết quả hoặc dự đoán tại thời điểm đó.
 - U, V và W : Đây là các tham số trong RNN, chúng điều chỉnh cách RNN thực hiện phép biến đổi từ đầu vào tới trạng thái ẩn và từ trạng thái ẩn tới đầu ra.

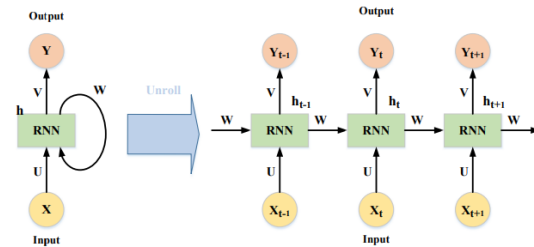


Fig. 8. A typical recurrent NN and its unrolled form. X_t is the input at time step t . h_t is the hidden state at time step t . Y_t is the output at time step t . U, V and W are parameters in the recurrent NN.

5) Support Vector Machine (SVM)

- Used in classification and pattern recognition: find a separating hyperplane in feature space to maximize the margin between different classes.
- Map the input vectors into a high-dimensional feature space.
 - **Không gian đặc trưng (Feature Space):** Mục tiêu của SVM là tạo ra một đường ranh giới (decision boundary) hoặc siêu mặt phẳng (hyperplane) để phân tách các điểm dữ liệu của các lớp khác nhau. Tuy nhiên, trong không gian đặc trưng ban đầu, việc này có thể trở nên phức tạp hoặc không thực hiện được. Do đó, SVM sử dụng một chiến lược để ánh xạ các điểm dữ liệu vào không gian đặc trưng mới, nơi việc tạo ra đường ranh giới trở nên dễ dàng hơn.
 - **Tích Vô Hướng trong Không Gian Đặc Trưng:** Sau khi ánh xạ dữ liệu vào không gian đặc trưng, SVM tính toán tích vô hướng giữa các điểm dữ liệu. Tích vô hướng này được sử dụng để xác định đường ranh giới tối ưu giữa các lớp dữ liệu. SVM tìm kiếm siêu mặt phẳng hoặc siêu phẳng tạo ra bằng cách tối ưu hóa khoảng cách giữa các điểm dữ liệu gần nhất của các lớp khác nhau.
- Applying different kernel functions, such as linear, polynomial and Radial Based Function (RBF).
 - **Hàm Kernel:** Hàm kernel là một phần quan trọng của SVM và được sử dụng để tính toán sự tương đồng hoặc tích vô hướng giữa các điểm dữ liệu trong không gian đặc trưng. Các hàm kernel phổ biến bao gồm:
 - **Linear Kernel:** Ánh xạ dữ liệu vào không gian tuyến tính. Nó thích hợp khi đường ranh giới là một siêu mặt phẳng tuyến tính.
 - **Polynomial Kernel:** Ánh xạ dữ liệu vào không gian đa thức, cho phép SVM tạo ra các đường ranh giới không tuyến tính.

- **Radial Based Function (RBF) Kernel:** Sử dụng hàm RBF để ánh xạ dữ liệu vào không gian vô hướng, thường dùng khi đường ranh giới phức tạp và không tuyến tính.

- Selection of kernel function depends on the training dataset.

6) Bayes' Theory

- Conditional probability: calculate the probability of an event occurring given prior knowledge of conditions that might be related to the event.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- E is a new evidence, H is a hypothesis
- E is a data sample, hypothesis H is the class to assign for the data sample. \Rightarrow requires a relatively small number of training dataset

7) Hidden Markov Models (HMM)

Used in randomly dynamic environments which obey the memoryless property.

1. **Mô hình Markov:** Mô hình Markov là một lớp các mô hình xác suất dựa trên việc mô tả các trạng thái và các xác suất chuyển đổi giữa các trạng thái này. Điểm chính của các mô hình Markov là tính chất không có bộ nhớ, nghĩa là xác suất chuyển đổi giữa các trạng thái chỉ phụ thuộc vào trạng thái hiện tại.
2. **Sự khác biệt giữa HMM và các mô hình khác:** Sự khác biệt chính giữa HMM và các mô hình Markov khác là HMM thường được áp dụng trong các môi trường trong đó trạng thái của hệ thống chỉ có thể quan sát một phần hoặc không quan sát được hoàn toàn. Trong HMM, chúng ta có hai loại biến:
 - **Trạng thái ẩn (Hidden State):** Đây là trạng thái thực sự của hệ thống, nhưng nó không thể quan sát trực tiếp. Ví dụ, trong ứng dụng nhận dạng tiếng nói, trạng thái ẩn có thể là ngữ cảnh (ví dụ: nói, im lặng).
 - **Trạng thái quan sát (Observation State):** Đây là trạng thái mà chúng ta có thể quan sát được. Trạng thái quan sát có thể phụ thuộc vào trạng thái ẩn, nhưng chúng ta chỉ biết về trạng thái quan sát.

B. Unsupervised Learning

- Set of inputs without labels \Rightarrow find patterns, structures, or knowledge in unlabeled data by clustering sample data into different groups
 - **Phân cụm (Clustering):** Trong phân cụm, mục tiêu là nhóm các dữ liệu có tính chất tương tự lại với nhau. Kết quả của phân cụm thường là các nhóm (clusters) của dữ liệu, và các điểm dữ liệu trong cùng một nhóm được coi là tương tự.
- Used in clustering and data aggregation

1) k-Means

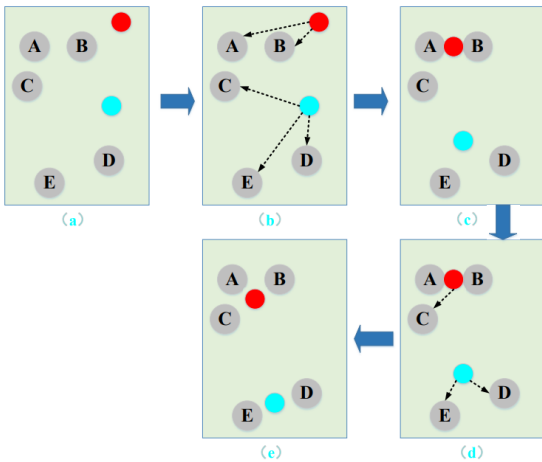


Fig. 10. Example of k-means algorithm, for $k = 2$. (a) Randomly choosing two data points as two centroids; (b) label each node with the closest centroid, resulting that node A and B are a class, node C, D and E are another class; (c) assign new centroids; (d) label each node with the closest centroid again, resulting that node A, B and C are a class, node D and E are another class; (e) the algorithm is converged.

- Implement: need two parameters (i.e., the initial dataset and the desired number of clusters)
 - Mục tiêu: tạo ra k cụm (clusters) sao cho mỗi dữ liệu thuộc vào cụm có trung tâm (centroid) gần nhất
- Desired number of clusters is k , resolve node clustering problem:
 - a. initialize k cluster centroids by randomly choosing k nodes
 - **Khởi tạo centroids:** Bước đầu tiên là chọn ngẫu nhiên k điểm dữ liệu ban đầu làm centroids ban đầu.
 - b. use a distance function to label each node with the closest centroid
 - **Gán dữ liệu vào các cụm:** Mỗi điểm dữ liệu trong tập dữ liệu được gán vào cụm có trung tâm gần nhất dựa trên một hàm khoảng cách (thường là khoảng cách Euclidean) giữa điểm và các centroids.
 - c. assign new centroids according to the current node memberships
 - **Tính toán trung tâm mới cho từng cụm:** Sau khi gán các điểm vào cụm, tính toán lại vị trí của các centroids bằng cách lấy trung bình của tất cả các điểm thuộc cùng một cụm.
 - d. stop the algorithm if the convergence condition is valid, otherwise go back to step (b).
 - **Kiểm tra điều kiện dừng:** Kiểm tra xem liệu các centroids đã hội tụ hay chưa. Nếu đã hội tụ, tức là không có thay đổi đáng kể trong vị trí của centroids, thuật toán dừng lại. Ngược lại, quay lại bước gán dữ liệu vào các cụm với các centroids mới.
 - **Kết thúc thuật toán:** Khi thuật toán dừng lại, bạn có k cụm được hình thành và mỗi điểm dữ liệu được gán vào một

2) Self-Organizing Map (SOM)

- Perform dimensionality reduction and data clustering
- Has two layers: input layer & map layer

- Number of neurons = desired number of clusters
- Each neuron has weight vector.
- Resolve data clustering:
 - a. initialize the weight vector of each neuron in the map layer
 - b. choose a data sample from the training dataset;
 - c. use a distance function to calculate the similarity between the input data sample and all weight vectors.
 - d. neuron whose weight vector has the highest similarity
 - e. neighborhood of the BMU is calculated
 - f. weight vectors of the neurons in the BMU's neighborhood (including the BMU itself) are adjusted towards the input data sample.
 - g. Stop the algorithm if the convergence condition valid, otherwise go back to step (b).

C. Semi-supervised Learning

- Uses both labeled and unlabeled data
- Make the best use of unlabeled data: assumptions have to be hold in semisupervised learning
 1. **Giả định về sự mềm mại (Smoothness Assumption):** Giả định này ngụ ý rằng các điểm dữ liệu gần nhau trong không gian đặc trưng (feature space) thường thuộc vào cùng một lớp hoặc có cùng một nhãn. Điều này có nghĩa rằng các điểm dữ liệu gần nhau có khả năng có nhãn tương tự.
 2. **Giả định về phân cụm (Cluster Assumption):** Giả định này cho rằng dữ liệu thường tự nhóm lại thành các cụm riêng biệt, và các cụm này có thể tương ứng với các lớp hoặc nhãn khác nhau. Điều này ngụ ý rằng các điểm dữ liệu trong cùng một cụm có thể thuộc cùng một lớp.
 3. **Giả định về phân tách mật độ thấp (Low-Density Separation Assumption):** Giả định này cho rằng các lớp hoặc nhãn trong dữ liệu thường được phân tách bởi các khu vực có mật độ thấp (low-density regions). Điều này ngụ ý rằng các điểm dữ liệu trong các khu vực mật độ thấp có thể thuộc vào các lớp khác nhau.
 4. **Giả định về manifold (Manifold Assumption):** Giả định này nói rằng dữ liệu thường được biểu diễn trên các đường cong hoặc bề mặt không gian (manifold) có số chiều thấp hơn. Điều này ngụ ý rằng dữ liệu trong không gian cao chiều thường nằm trên một không gian con có số chiều thấp hơn.
- Pseudo Labeling:
 - use the labeled data to train a model
 - use the trained model to predict pseudo labels of the unlabeled data
 - combine the labeled data and the newly pseudo-labeled data to train the model again
- Semi-supervised learning methods: Expectation Maximization (EM), co-training, transductive, SVM and graph-based methods.
 1. **Expectation Maximization (EM):** Phương pháp EM dựa trên giả định về cụm (cluster assumption). Nó giả định rằng dữ liệu được sinh ra từ một số phân phối xác suất khác nhau, và nhiệm vụ của mô hình là phát hiện và gán nhãn các cụm này.
 2. **Co-training:** Co-training là một phương pháp dựa trên giả định về sự phân cụm và sự mềm mại (cluster and smoothness assumptions). Nó giả định rằng có hai tập dữ liệu độc lập, và mô hình được đào tạo trên

mỗi tập dữ liệu một cách riêng biệt. Sau đó, các dự đoán từ hai mô hình được sử dụng để cải thiện hiệu suất.

3. **Transductive SVM:** Phương pháp SVM dựa trên giả định về sự phân tách mật độ thấp (low-density separation assumption). Nó giả định rằng các lớp trong dữ liệu thường được phân chia bởi các **khu vực mật độ thấp**, và nó cố gắng tận dụng thông tin từ các điểm dữ liệu không được gán nhãn để cải thiện mô hình SVM.
4. **Graph-based methods:** Các phương pháp dựa trên đồ thị (graph-based methods) dựa trên giả định về manifold (manifold assumption). Chúng giả định rằng dữ liệu thường nằm trên các đường cong hoặc bề mặt không gian có số chiều thấp hơn và sử dụng đồ thị để biểu diễn các mối quan hệ giữa các điểm dữ liệu.

D. Reinforcement Learning

1) Reinforcement Learning (RL)

- RL involves an agent, a state space S and an action space A .
- Agent: learning entity which interacts with its environment to learn the best action to maximize its long-term reward (a cumulative discounted reward)
- SDN: controller monitors the network status and learns to make decisions to control data forwarding
 - learn the optimal behavior policy π : direct map from the state space S
 - $(\pi : S \rightarrow A)$: maximize the expected long-term reward.
 - determine the best corresponding action given a particular state
- Q-learning to learn a table storing all state-action pairs and their long-term rewards

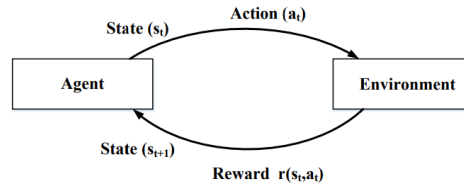


Fig. 11. A basic diagram of a RL system. The agent takes an action according to the current state and then receives a reward. $r(s_t, a_t)$ denotes the immediate reward that the agent receives after performing an action a_t at the state s_t .

1. **Mục tiêu của Q-learning:** Mục tiêu chính của Q-learning là tìm cách tối ưu hóa hàm giá trị Q (Q-function) cho một tác vụ cụ thể trong môi trường tương tác. Hàm Q đo lường giá trị dự kiến của một cặp trạng thái-hành động và được sử dụng để đưa ra quyết định tối ưu.
2. **Hàm Q và bảng Q :** Hàm Q được biểu diễn bằng một bảng Q , thường được gọi là bảng Q-learning. Bảng này lưu trữ giá trị $Q(s, a)$ cho từng trạng thái s và hành động a . Ban đầu, bảng Q được khởi tạo một cách ngẫu nhiên hoặc với các giá trị ban đầu.
3. **Cập nhật hàm Q :** Q-learning cập nhật giá trị của hàm Q dựa trên các trải nghiệm tương tác với môi trường. Cụ thể, thuật toán sử dụng công thức cập nhật sau khi thực hiện hành động a tại trạng thái s và quan sát trạng thái tiếp theo s' và phần thưởng r :

$$Q(s, a) = Q(s, a) + \alpha * [r + \gamma * \max_{a'}(Q(s', a')) - Q(s, a)]$$

- $Q(s, a)$ là giá trị Q trước cập nhật.
- α là tốc độ học tập (learning rate) quyết định độ lớn của cập nhật.

- r là phần thưởng nhận được sau khi thực hiện hành động a .
 - γ là hệ số chiết khấu (discount factor) để xác định giá trị của tương lai.
 - $\max(Q(s', a'))$ là giá trị Q lớn nhất cho tất cả hành động có thể thực hiện tại trạng thái s' .
4. **Quyết định hành động:** Sau khi cập nhật bảng Q , thuật toán chọn hành động tối ưu tại mỗi trạng thái bằng cách lựa chọn hành động có giá trị Q cao nhất cho trạng thái đó.
 5. **Khám phá và khai thác:** Q-learning thường kết hợp sự khám phá và khai thác. Trong quá trình học, thuật toán có thể thực hiện các hành động ngẫu nhiên để khám phá môi trường và đồng thời khai thác tri thức đã học được từ bảng Q .
 6. **Hội tụ:** Q-learning được đảm bảo hội tụ đến giá trị Q tối ưu khi đủ dữ liệu tương tác với môi trường.

2) Deep Reinforcement Learning (DRL)

- Approximate the value function by leveraging the powerful function approximation property of deep NNs.
 - **Xử lý không gian trạng thái và hành động lớn:** DRL có khả năng học và xấp xỉ hàm giá trị cho các không gian lớn hơn bằng cách sử dụng mạng nơ-ron sâu.
 - **Tăng tốc học tập:** DRL thường học nhanh hơn và có khả năng hội tụ nhanh hơn đến chiến lược tối ưu.

3) RL-based Game Theory

- Strategic interactions among rational decision-makers
- Set of players, a set of strategies and a set of utility functions
- Cooperative game theory and noncooperative game theory
 - Cooperative games: maximize the utility of their coalitions
 - Non-cooperative games: choose strategies individually to maximize their own utility \Rightarrow nodes selfish
- Adaptive learning methods: select their optimal strategies by learning from historical info as network status

V. MACHINE LEARNING IN SDN

- Bring intelligence to the SDN controller
- Existing machine learning efforts to address issues in SDN, such as traffic classification, routing optimization, QoS/QoE prediction, resource management and security

A. Traffic Classification

- Techniques: Port-based approach, Deep Packet Inspection (DPI) and machine learning
 - **Port-based approach:** use TCP and UDP port numbers to determine applications. \Rightarrow not effective
 - **DPI:** matches the payload of traffic flows with predefined patterns to identify the applications that traffic flows belong to.
 - **ML-based:** correctly recognize encrypted traffic and incur much lower computational cost

1) Elephant Flow-aware Traffic Classification

- Identify the elephant flows (long-lived, bandwidth-hungry flows) and the mice flows (short-lived, delay-intolerant flows)
- Control the traffic flows in data centers effectively
- Elephant flow detection strategy: two stages
 - head packet measurement: distinguish suspicious elephant flows from mice flows
 - decision tree: detection method to analyze whether these suspicious elephant flows

2) Application-aware Traffic Classification

- Identify the applications of traffic flows.
- OpenFlow-based SDN system: apply classifier algorithms to classify traffic flows
- **MultiClassifier**: identify applications by combining ML-based classifier and DPI-based classifier.
 - ML-based classifier: reliability larger than a threshold value, MultiClassifier's result
 - Otherwise, DPI-based classification done
 - not return "UNKNOWN": result selected as MultiClassifier's result.
- **SVM algorithm**: classify UDP traffic according to Netflow records
- **Mobile application classification**: Atlas framework
 - Crowd sourcing approach: collect ground truth data from end devices
 - Collected data used to train the decision tree.
 - deep NN: train a 8-layer deep NN model.
 - **TrafficVision**: SDN-enabled wireless edge network
 - Collecting, storing and extracting flow statistics and ground-truth training data from end devices and access devices.
 - A decision tree classifier is used to identify the application name
 - A k-NN classifier is applied to identify the flow types

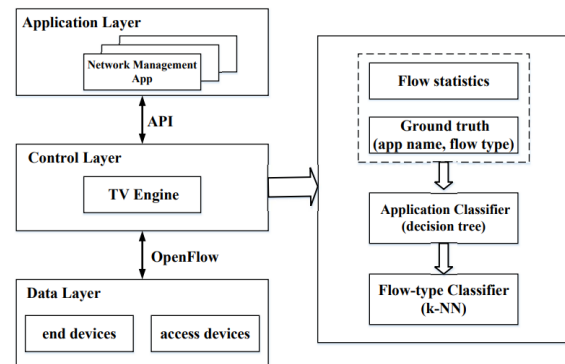


Fig. 12. High-level SDN-enabled TrafficVision architecture and the workflow of TV Engine [148]. TV Engine has three major tasks: (1) Collecting flow statistics and ground-truth training data from end devices and access devices. (2) A decision tree classifier is applied to identify application name. (3) A k-NN classifier is applied to identify flow types.

3) QoS-aware Traffic Classification

- Identify the QoS classes of traffic flows: classify traffic flows according to their QoS requirements

- QoS-aware traffic classification system: leveraging semi-supervised learning algorithm and DPI.
 - DPI: label a part of traffic flows of known applications.
 - labeled training dataset used by semisupervised learning algorithms (Laplacian SVM): classify the traffic flows of unknown applications.
 1. **Bước lọc thông tin:** Thuật toán sử dụng dữ liệu huấn luyện không được gán nhãn để xác định các mối tương đồng hoặc cấu trúc ẩn trong dữ liệu. Nó có thể sử dụng ma trận Laplacian để biểu diễn mối tương đồng giữa các mẫu dữ liệu. Ma trận Laplacian là một phần quan trọng trong đồ thị phân loại và giúp xác định cấu trúc của dữ liệu.
 2. **Phân loại:** Dựa trên cấu trúc và thông tin học được từ dữ liệu huấn luyện không được gán nhãn, Laplacian SVM sẽ cố gắng phân loại các luồng lưu lượng mạng mới vào các nhóm đã được học.
- ⇒ categorized into different QoS classes, high classification accuracy

4) Analysis

- Elephant flow-aware traffic classification applied in data centers. Data center: schedule traffic flows rapidly
- Application-aware traffic classification applied fine-grained network management
- QoS-aware traffic classification: used by network operators to optimize network resource allocation for traffic flows according to their desired QoS.
- DPI: label traffic flows, incurs high computational cost
- Supervised >< semi-supervised learning (small part of labeled data ⇒ fine-grained traffic classification.)
- Increasing of dimension and volume of training datasets: classification accuracy can be improved.
- deep learning algorithms more suitable for processing large high-dimensional training datasets
 - deeper neural network has higher complexity

B. Routing Optimization

- Routing: control the routing of traffic flows by modifying flow tables in switches ⇒ optimize the routing of traffic flows
- Shortest Path First (SPF) algorithm and heuristic algorithms: routing optimization
 - SPF: routes packets according to simple criteria (hop-count or delay)
 - best-effort routing protocol: not best-effort routing protocol
 - Heuristic: high computational complexity
 1. **Tính gần đúng:** Heuristic algorithms tập trung vào việc tìm kiếm các giải pháp gần đúng chấp nhận được thay vì tìm kiếm giải pháp tối ưu chính xác. Điều này cho phép chúng hoạt động nhanh hơn và thường dùng trong các tình huống mà việc tìm kiếm toàn bộ không gian giải pháp quá tốn kém.
 2. **Tiêu chuẩn và hàm mục tiêu:** Heuristic algorithms thường dựa trên một số tiêu chuẩn hoặc hàm mục tiêu để đánh giá chất lượng của giải pháp. Chúng cố gắng cải thiện giá trị của tiêu chuẩn này thông qua các bước tối ưu hóa.
 3. **Không bảo đảm tối ưu tuyệt đối:** Một trong những hạn chế của heuristic algorithms là chúng không đảm bảo tìm ra giải pháp tối ưu tuyệt đối cho mọi tình huống. Thay vào đó, chúng cố gắng tìm giải pháp tốt nhất có thể trong khoảng thời gian và tài nguyên cho phép.

4. **Ví dụ về heuristic algorithms:** Có nhiều loại heuristic algorithms, và mỗi loại được thiết kế cho một loại vấn đề cụ thể. Ví dụ về heuristic algorithms bao gồm thuật toán tìm kiếm bằng cách sử dụng nguyên tắc tìm kiếm xung quanh (local search), thuật toán tìm kiếm bằng sự thử nghiệm (trial-and-error), và thuật toán tối ưu hóa dựa trên quần thể (population-based optimization) như thuật toán tối ưu hóa mề mống (genetic algorithms) và thuật toán tối ưu hóa kiến thức của kiến (ant colony optimization).

- Machine learning algorithms: give near-optimal routing solutions quickly, not need an exact mathematical model of underlying network

1) Supervised Learning-based Routing Optimization

- Optimize routing, the network and traffic states:
input of the training datasets \Rightarrow heuristic algorithms: output
- architecture with supervised ML-based meta layer:
solve dynamic routing problem
 - Training dataset: input of heuristic algorithm & its corresponding output \Rightarrow obtain real-time heuristic-like results.
- Dynamic routing framework: NeuRoute
 - LSTM: used to estimate future network traffic.
 - Input: network state & estimated network traffic
 - Output: corresponding routing solution calculated by heuristic algorithms

\Rightarrow train the neural network model: obtain the real-time heuristic-like results.

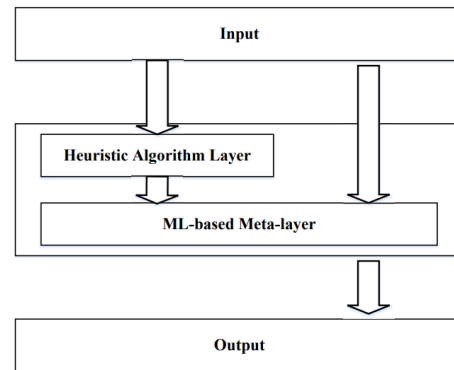


Fig. 13. A dynamic routing architecture [150]. The architecture consists of a ML-based meta-layer and a heuristic algorithm layer. The input of heuristic algorithm and its corresponding output are the training dataset of the ML-based meta-layer. After the training phase, the ML-based meta-layer can make the optimal routing decisions directly and independently.

2) RL-based Routing Optimization

- RL algorithms: solve decision-making problems, optimize routing
- Controller: agent - network: environment - state space: composed of network and traffic states - action: routing solution
 - Reward: optimization metrics (network delay)
- Cognitive Routing Engine (CRE): find the optimal overlay paths between geographically dispersed data centers, by using random NN and RL
- QoS-aware Adaptive Routing (QAR): enable time-efficient adaptive packet forwarding by utilizing RL algorithm
 - Reward: based on traffic types and users' applications
- DRL model: select the optimal routing paths for all source-destination pairs given the traffic matrix to minimize the network delay

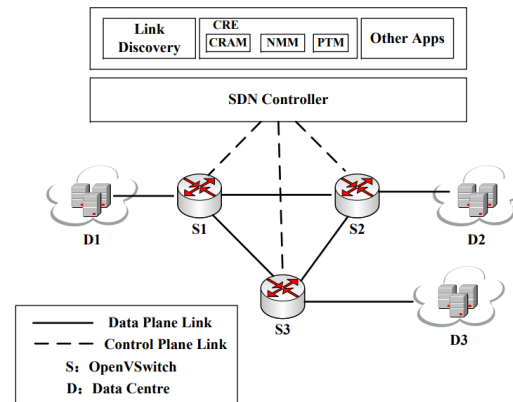


Fig. 14. The SDN-enabled overlay network with CRE [153]. CRE consists of three main modules: CRAM, NMM and PTM [154]. CRAM stands for Cognitive Routing Algorithm Module, NMM for Network Monitoring Module, PTM for Path-to-OF Translator Module. CRAM uses random NN and RL to find network paths which maximize the objective function set by the tenant's traffic engineering policy. NMM is in charge of collecting required network state information to update the random NN in the CRAM. PTM is responsible for converting the optimal paths found by the CRAM into the appropriate OpenFlow messages, which guide the SDN controller to reroute the network traffic.

3) Traffic Prediction

- Predict the trend of traffic volume by analyzing historical traffic info
- SDN controller: make traffic routing decisions in advance and distribute the proactive routing policies to forwarding devices in the data plane to guide traffic flow routing in the near future.
 - take appropriate actions before traffic congestion occurs
 - facilitate proactive provision of network resources
- Load balance strategy: optimize path load.
 - Four flow features (i.e., transmission hop - số bước truyền, transmission latency, packet loss rate and bandwidth utilization ratio): predict load of each path using neural network model.
 - least loaded path: selected as the transmission path of new traffic flows.
- LSTM-based framework - NeuTM: predict network traffic matrix
 - Train the LSTM model: traffic data from the GEANT backbone network

4) Others

- Routing protocol in environments with strict compliance requirements: efficient risk-based swarm routing protocol
- First, k-means algorithm: cluster network traffic into several clusters of risk ratios in an off-line mode
- Then, Ant Colony Optimization (ACO): select paths with minimized privacy exposure and compliance risks for a given data transmission session in an on-line mode.

5) Analysis

- **Supervised learning algorithms:** obtain the optimal heuristic-like routing solutions \Rightarrow shortcomings: acquisition of labeled training datasets
- **RL:** not need labeled training datasets, optimization targets (e.g., energy efficiency, throughput and delay) adjusted flexibly through different reward functions.
- **Traffic prediction:** promote implementation of routing pre-design, reduce transmission delay in the data plane by modifying switches' flow tables in advance.
- **Optimal neural network architecture:** measurement of training time and learning performance

C. QoS/QoE Prediction

- QoS parameters: network-oriented metrics, used by network operators to assess network performance
- SDN: centralized architecture, collect statistics from the switches at per port & per flow granularity levels

1) QoS Prediction

- Key Performance Indicators (KPIs): packet size, transmission rate and queue length, etc.
- Quantitative correlations between KPIs and QoS parameters: improve the QoS management by predicting QoS parameters according to KPIs.
- QoS parameters (continuous data): problem - regression task \Rightarrow need supervised learning
- Estimate the network delay given the traffic load and the overlay routing policy
- Delay estimation: two different models (i.e., traditional M/M/1 network model and neural network model)
 - NN-based estimator has better performance than M/M/1 model
 - M/M/1 model is easier for humans to understand.
- Two-phase analysis mechanism: improve the QoS prediction
 - Decision tree: discover correlations between KPIs and QoS parameters.
 - Linear regression ML algorithm: perform root cause analysis and discover each KPI's quantitative impact \Rightarrow predict traffic congestion and provide recommendations on QoS improvement.
- Application-aware QoS estimation: two learning methods (i.e., random forest and regression tree)
 - Video on-Demand (VoD) application (i.e., frame rate and response time): collected operating system-granularity statistics, port-granularity statistics and flow-granularity statistics
 - forward-stepwise-selection technique: reduce the feature set size + low level of QoS estimation error

2) QoE Prediction

- QoE: subjective metric
- QoE metric: Mean Opinion Score (MOS), divides QoE values into five levels, including excellent, good, fair, poor and bad.
 - Obtain: number of users are invited to rate the quality of a service

- ML (supervised learning): effective method to learn the relationship between QoS parameters and the QoE values
 - Video streaming services: estimate the MOS value by network parameters (e.g., RTT, jitter, bandwidth and delay) \Rightarrow adjust video parameters (e.g., resolution, frame per second and bitrate)
 - Four ML algorithms (i.e., DT, neural network, k-NN and random forest): predict QoE values based on video quality parameters
 - Two metrics, Pearson correlation coefficient and RootMeanSquare-Error, are applied to assess the performance
1. **Tham số chất lượng video (SSIM và VQM):** SSIM (Structural Similarity Index) và VQM (Video Quality Metric) là các tham số được sử dụng để đo lường chất lượng video. Chúng thường được sử dụng để **xác định mức độ tương đồng giữa video gốc và video được truyền đi**. SSIM và VQM là các chỉ số số học được tính dựa trên **phân tích cấu trúc và đặc điểm của hình ảnh trong video**.
 2. **Đánh giá hiệu suất:** Để đánh giá hiệu suất của các thuật toán dự đoán, hai chỉ số được sử dụng:
 - **Pearson Correlation Coefficient:** Một chỉ số đo lường mối quan hệ tuyến tính giữa giá trị dự đoán và giá trị thực tế. Giá trị của Pearson Correlation Coefficient nằm trong khoảng từ -1 đến 1, với 1 đại diện cho mối quan hệ tuyến tính hoàn hảo.
 - **Root-Mean-Square Error (RMSE):** Một chỉ số đo lường sai số giữa giá trị dự đoán và giá trị thực tế. RMSE thể hiện sự chênh lệch trung bình giữa dự đoán và thực tế.

3) Analysis

- QoS prediction: discover the quantitative correlations between KPIs and QoS parameters \Rightarrow regression task
- QoE prediction: discover the quantitative correlations between QoS parameters and QoE values. \Rightarrow classification task

\Rightarrow supervised learning techniques

- Semi-supervised learning algorithms: also effective for QoE prediction
- Prediction accuracy of random forest is higher \gg complexity of regression tree is lower

TABLE IV
ML-BASED ROUTING OPTIMIZATION SOLUTIONS IN SDN.

| Ref. | Objective | Learning model | Complexity | Brief summary |
|-------|---------------------------|----------------|------------|---|
| [150] | Routing optimization | Neural network | Fair | Using ML-based meta-layer to obtain real-time heuristic-like traffic routing results |
| [151] | Routing optimization | Neural network | Fair | A ML-based dynamic routing framework called NeuRoute |
| [152] | Routing optimization | RL | Fair | Using RL to select the optimal forwarding paths to minimize network cost considering delay, loss rate and bandwidth |
| [155] | Routing optimization | RL | Fair | Using RL to select the optimal forwarding paths to maximize QoS-aware reward |
| [156] | Routing optimization | DRL | High | Using DRL to select the optimal forwarding paths to minimize network delay |
| [158] | Traffic prediction | Neural network | Fair | Using NN model to predict traffic load which is used to calculate the optimal routing decisions in advance |
| [159] | Path load prediction | Neural network | Fair | Using NN model to predict the load of each path |
| [160] | Traffic matrix prediction | LSTM | High | Using LSTM to predict network traffic matrix |
| [162] | Traffic clustering | k-means | Low | A ML-based data privacy preservation routing protocol in SDN |

D. Resource Management

- Efficient network resource management: improve network performance
- SDN separates the control plane from the data plane: maximize the utilization of network resources

1) Data Plane Resource Management

- Three types of resources: networking, caching and computing resources.
- Networking resources: spectrum (tần số), bandwidth and power - deliver data through networks to meet QoS/QoE requirements

- Caching resources: store the frequently requested data at devices in data plane. \Rightarrow reduces data transmission delay & decreases duplicate data transmission.
- Computing resources (**Tài nguyên tính toán**): limited computing resources and battery capacity \Rightarrow offload computing tasks - Edge Computing technologies
- Two scenarios in SDN: single-tenancy and multitenancy SDN network.
 - Single: logically centralized controller control all resources in the data plane
 - Multi: resources in the data plane are shared by multiple tenants (người dùng). \Rightarrow tenants use their own SDN controllers to control their isolated resources in the data plane.

a) Resource Allocation in Single-tenancy SDN Network

- Integrated framework: enhance performance of software-defined virtualized Vehicular Adhoc Network (VANET) - **allocate networking, caching and computing resources dynamically**. \Rightarrow joint optimization problem
 - DRL algorithm: solve the **complex optimization problem** and obtain the **resource allocation policy**.
- Roadside Unit (RSU) cloud architecture: instantiate, replicate, and migrate (tạo ra, sao chép và di chuyển) services dynamically
 - RL-based heuristic approach: select configurations (cấu hình mạng) that **minimize long-term reconfiguration cost**
- Content distribution optimization: improve user satisfaction.
 - RL-based context-aware content distribution scheme to improve the content distribution QoS: **select the optimal protocol** (i.e., TCP/IP or Content Centric Networking (CCN)), **based on network status** (content distribution time, network throughput and network overhead)
 - **Intelligent streaming architecture - SDNHAS**: help HTTP Adaptive Streaming (HAS) players decide the optimal bitrate and quality
 - RL algorithm: leverage bitrate and quality decisions in each cluster
- Cache-enabled Unmanned Aerial Vehicles (UAVs) in a Cloud Radio Access Network (C-RAN): optimize users' QoE and UAVs' transmit power
 - ESN: type of recurrent NNs, predict content request distribution and mobility pattern of each user. \Rightarrow user-UAV association, each UAV's location and the content to cache at each UAV

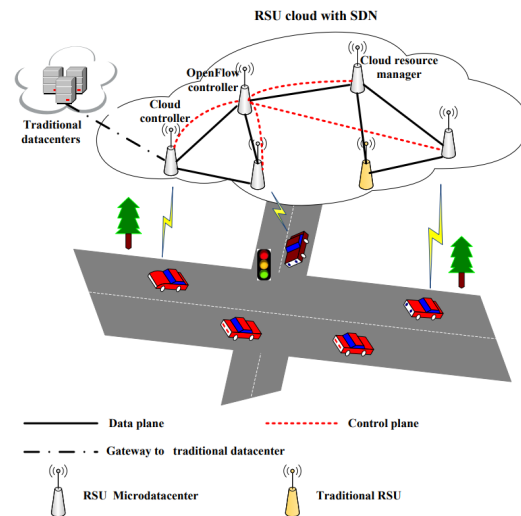


Fig. 15. RSU cloud architecture [172]. RSU cloud is composed of traditional RSUs and RSU microdatacenters. Traditional RSUs as fixed roadside infrastructures can perform Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communications. RSU microdatacenters are the fundamental components of the RSU clouds, which host services to satisfy the demands of mobile vehicles. The difference between traditional RSUs and RSU microdatacenters is that additional hardware and software components are deployed in RSU microdatacenters to provide virtualization and communication capabilities.

b) Resource Allocation in Multi-tenancy SDN Network

- **Goal:** allocate data plane resources among multiple tenants to maximize their utility
- RRH assignment issue among Mobile Network Operators (MNOs) in C-RAN
 - RRH assignment: non-cooperative game-theoretic problem
 - players - RRHs, action - MNO selection, utility of each player - maximize its Received Signal Strength (RSS) level. \Rightarrow regret-matching based learning algorithm used by each player to select the optimal set of MNOs.
- Computation offloading issue in MEC.
 - Offloading problem: noncooperative game-theoretic problem
 - Players - MEC servers, two actions of each player - active/inactive, utility of each player - minimize its energy consumption
- Spectrum sharing problem in LTE and WiFi coexistence system: non-cooperative game-theoretic problem
 - LTE & WiFi: belong to different operators & have their own controllers \Rightarrow share the same unlicensed spectrum resources
 - Players - two controllers, action - amount of shared spectrum resources, utility - maximize the spectrum resource utilization
 - Each round: decision tree algorithm to predict the opponent's network status
- Network function assignment issue in SDN/NFV system (network function market): two stage Stackelberg game
 - servers and users: sellers and buyers of network functions
 - Servers - leaders, users - followers \Rightarrow make their optimal decisions to maximize their utility and benefit
 - RL algorithm: applied to converge to the equilibrium.



Giải thích:

- Nhà khai thác Mạng Di động (Mobile Network Operators - MNOs)
- **Regret-matching:** Đây là một kỹ thuật trong lý thuyết trò chơi và học tăng cường (reinforcement learning). Khi một người chơi tham gia vào một trò chơi nhiều lần và phải lựa chọn hành động trong mỗi lượt chơi, thuật toán regret-matching giúp người chơi học từ kết quả của các lần chơi trước đó để đưa ra quyết định tối ưu trong tương lai.
- **Regret:** Regret là sự tiếc nuối hoặc lỗ hổng trong quá khứ mà một người chơi có thể cảm thấy khi chọn một hành động cụ thể trong mỗi tình huống. Nếu một người chơi đã chọn một hành động nhưng sau này nhận thấy rằng có một hành động khác tốt hơn, thì sự khác biệt giữa kết quả tối ưu và kết quả đã đạt được được gọi là "regret" cho lượt chơi đó.
- **Regret-matching algorithm:** Thuật toán này sử dụng thông tin về các "regrets" tích lũy từ các lượt chơi trước để điều chỉnh việc chọn hành động trong tương lai. Người chơi tự động chọn hành động có "regret" thấp hơn, dựa trên quy tắc giảm thiểu sự tiếc nuối. Dần dần, theo thời gian, thuật toán học cách chọn hành động tối ưu dựa trên lý thuyết trò chơi và thông tin nhận được từ kết quả của các lần chơi trước đó.

c) Admission Control (AC)

- **Goal:** manage a large number of service requests by **accepting or rejecting new incoming requests** according to the resource availability.
- **ML-based method:** select the most proper AC algorithm from a pool of online algorithms (experts)

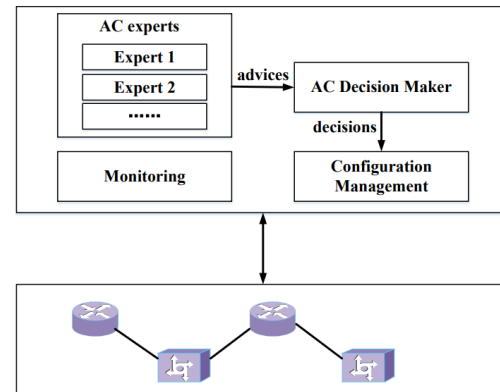


Fig. 16. A ML-based AC method in the SDN controller [178], [179]. Each expert is an online AC algorithm, such as Greedy, Agrawal [180] and AAP-pd [181], [182]. AC Decision Maker selects the best decision from all experts according to varying traffic conditions.

2) Control Plane Resource Management



Giải thích:

- Trong ngữ cảnh của mạng và máy chủ ảo hóa (virtualization), hypervisor (hoặc Virtual Machine Monitor - VMM) là một phần mềm hoặc thiết bị phần cứng có nhiệm vụ quản lý và chạy các máy ảo (virtual machines - VMs) trên một máy chủ vật lý. Hypervisor cho phép nhiều máy ảo chia sẻ tài nguyên phần cứng của máy chủ một cách an toàn và hiệu quả.
- Cụ thể, hypervisor tạo ra môi trường ảo để các máy ảo chạy trên máy chủ vật lý, và nó quản lý việc phân phối tài nguyên phần cứng (như CPU, bộ nhớ, ổ cứng) cho các máy ảo. Điều này cho phép nhiều máy ảo hoạt động đồng thời trên cùng một máy chủ, mỗi máy ảo chạy hệ điều hành và ứng dụng của riêng nó mà không ảnh hưởng đến các máy ảo khác.

a) Control Plane Resource Allocation

- Multi-tenancy SDN network (FlowVisor and OpenVirteX): network hypervisors introduced between data plane & control plane \Rightarrow allow tenants to use their own controllers to control their isolated resources.
- Limited computing resources (e.g., CPU): how to guarantee communication between each tenant's data plane & control plane
 - ML algorithms: optimize resource allocation.
- Resource monitor tool: monitor CPU consumption
 - Benchmarking tool hvbench: measure the control message rate.
 - Collected data: train three different regression learning models to learn the mapping between the CPU consumption and the control message rate

- Trained mapping models: whether network hypervisors are overloaded from measured control message rate.
- Available computing resources of hypervisors are fluctuant: to detect the resource change, CPU consumption information and the control message rate information are collected continuously
 - SVM: compare the collected data and the current mapping model
 - collected data not fit the current model → big resource change occurred → mapping model re-trained

b) Controller Placement

- Controller placement problem: long distance between the controller and switches
- Heuristic algorithms: approach to solve this problem
- Shortcomings: high computational complexity
- Supervised learning algorithms (decision tree, neural network and logistic regression): effective to solve the controller placement problem.
 - Input: traffic distribution - output: corresponding controller placement solutions of heuristic algorithms
 - Predicted solutions: optimal solutions/initial solutions of heuristic algorithms

3) Others

- WiFi-Direct network: estimate the number of active UEs
 - remaining transmission time (i.e., the Estimated Time of Arrival (ETA)) during the file transmission in SDNbased wireless networks: help the clients to schedule their application-layer actions
- Autonomic network management in software defined 5G systems: Autonomic Manager (AM) provide the network intelligence
 - **Diagnoser, Decision-Maker (DM) and Action Enforcer (AE)**: three main functional blocks in AM
- Service Level Agreements (SLA) management in SDN: use LSTM to calculate next values of the system features at time $t + 1$ given the numerical values of these features at time t .
 - Forecasted values of features are taken by decision tree algorithm: **determine the most likely SLA Violation (SLAV)**
- Statistical machine learning approach: **increase the robustness** of SDN-based Wireless Sensor Network (WSN)

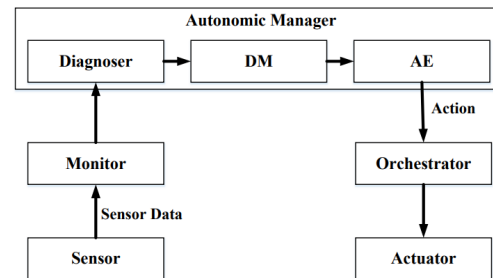


Fig. 17. The intelligent control loop in SELFNET framework [191]. The sensors in the network generate sensor data. The monitor analyzes and aggregates the sensor data to detect network problems. The diagnoser is capable of diagnosing the root cause of network problems reported by the monitor. Based on reasons of these problems, a set of corrective and preventive methods are decided by the Decision-Maker (DM). Action Enforcer (AE) is in charge of providing implementable actions to be enforced in the network infrastructure. The orchestrator and actuator are responsible for executing the implementable actions

- Interference prediction: controller can modify network state to **keep the system stable** and to **prevent network malfunctions**

4) Analysis

- Data plane resource allocation: decision-making task \Rightarrow RL (single-tenancy SDN network) and ML-based game theory (multi-tenancy SDN network)
- Control plane resource allocation: considered by network hypervisors to **allocate their limited resources** to multiple tenants.
- Mapping between the **resource consumption and control message rate**: regression task.
- Mapping task: supervised learning techniques \rightarrow dynamic scenarios: **trained mapping model** needs to be **updated periodically**.

TABLE VI
ML-BASED RESOURCE MANAGEMENT SOLUTIONS IN SDN.

| Ref. | Objective | Learning model | Complexity | Brief summary |
|-------|--|--|------------|---|
| [170] | Data plane resource allocation | DRL | High | Using DRL to jointly allocate networking, caching and computing resources in software-defined vehicular networks to maximize network operators' comprehensive revenue |
| [171] | Data plane resource allocation | DRL | High | Using DRL to jointly allocate networking, caching and computing resources in smart cities to maximize network operators' comprehensive revenue |
| [172] | Service reconfiguration | RL | Fair | Using RL to select the optimal configuration strategy to minimize the long-term reconfiguration cost in VANET |
| [173] | Content delivery optimization | RL | Fair | Using RL to select the optimal content delivery protocol (i.e., TCP/IP or CCN) to maximize the content distribution QoS |
| [174] | HAS traffic delivery optimization | RL | Fair | Using RL to select the optimal bitrate and quality to maximize users' QoE when delivering HAS media content |
| [177] | Cache-enabled UAVs deployment optimization | Recurrent NN | High | Using ESN, a type of recurrent NNs, to predict the content request distribution and mobility pattern of each user |
| [131] | RRH allocation | Regret-matching-based game theory | Fair | Utilizing regret-matching-based game theory for RRH allocation among MNOs to maximize RSS levels |
| [132] | Computation offloading in MEC | RL-based game theory | Fair | Using RL-based game theory method to solve the energy-efficient MEC server activation problem |
| [133] | Spectrum sharing | DT-based game theory | Fair | Using decision tree to predict the opponent's network status, based on which the controllers make their optimal decisions to maximize the spectrum resource utilization |
| [134] | Network function assignment | RL-based game theory | Fair | Using RL-based Stackelberg game theory to optimize the network function assignment between servers and users |
| [185] | Hypervisor resource utilization estimation | Regression learning model | Low | Using regression learning models to learn the mapping between CPU consumption and control message rate |
| [186] | Hypervisor resource change detection | SVM | Fair | Using SVM to calculate whether the recently collected data fits the current mapping model |
| [187] | Controller placement | Decision tree, neural network, logistic regression | Fair | Using supervised learning algorithms to obtain the controller placement solutions in real time |
| [189] | Estimating the number of active nodes | Naive Bayes, SVM, k-NN | Fair | Different ML approaches to predict the number of active nodes in a Wi-Fi-Direct network |
| [192] | SLA management | LSTM, decision tree | High | Using LSTM to predict the values of system parameters, and using decision tree to determine the most likely SLA violation |
| [193] | Increasing robustness of WSN | ML approaches | Fair | Using ML techniques to predict interference patterns over time to increase the robustness of SDN-based WSN |

E. Security

- Intrusion detection: important element for network security
- Intrusion Detection System (IDS): monitor the events in a network system and identify possible attacks
- Two types of IDSs: signature-based IDS and anomaly-based IDS
- Signature-based IDS: humans **create the signatures of known attacks**, **compares these traffic flows** against the known signatures to identify possible malicious activities.
 - Shortcomings: signature update difficult, incurs high time consumption
- Anomaly-based IDS: uses the collected data related to the behavior of legitimate users to create a model, traffic flows are compared with the model.
- anomaly based IDS: flow-based traffic identification based on flow-granularity information - packet header information
- Supervised learning algorithms: applied for intrusion detection (model to identify normal activities and intrusions)
 - Feature reduction done to **reduce the dimensionality of dataset input**: Feature selection (choose a subset of appropriate features from all flow features) and feature extraction (extracting a set of new features from the original features)

1) Coarse-grained Intrusion Detection

- Goal: classify traffic flows as normal and abnormal classes.

- Composed of data preprocessing, predictive data modeling, and decision making and response subsystem
 - Forward feature selection strategy: select appropriate feature sets
 - Decision tree and random forest algorithms: detect malicious activities
 - Reactive routing: install different flow rules for different flow types
- HMM-based Network Intrusion Detection System (NIDS): use five selected flow features (i.e., length of the packet, source port, destination port, source IP address and destination IP address) to determine the maliciousness
- ATLANTIC framework: perform anomaly traffic detection, classification and mitigation jointly in two phases: a lightweight phase and a heavyweight phase
 - Lightweight phase uses information theory to calculate deviations in entropy of flow tables.
 - The heavyweight phase utilizes SVM algorithm to classify abnormal traffic.
- Intrusion detection and mitigation architecture (IoT-IDM): composed of five key modules (Device Manager, Sensor Element, Feature Extractor, Detection, and Mitigation)
- Four ML algorithms (i.e., decision tree, BayesNet -be, decision table and Naive Bayes): predict potential malicious connections and vulnerable hosts
- Deep NN model: input layer, three hidden layers and an output layer is trained based on the NSL-KDD dataset
 - OpenFlow protocol: modify switches' flow tables in order to prevent attacks
- Deep recurrent NN in an anomaly-based IDS & a [Gated Recurrent Unit Recurrent Neural Network \(GRU-RNN\)](#) algorithm (train by [six flow features - duration and protocol type](#)): detect intrusion

2) Fine-grained Intrusion Detection

- Fine-grained classification of network traffic and identify different types of attacks.
- Improved behaviour-based SVM: categorize network attacks.
 - Decision tree: feature reduction approach to outrank raw features and select the most qualified features (input data to train SVM classifier)
- NDAE: combines the deep learning approach (feature reduction) and random forest (traffic classification and intrusion detection)

3) DDoS Attack Detection

- DDoS attack: exhaust system resources by [simultaneously sending a large number of fake requests](#) using many puppet machines
- NOX-based SDN network: controller collects traffic flow feature info from OpenFlow switches.
 - SOM: perform DDoS attack detection according to the collected traffic flow features
- IDS: consists of two modules - Signature IDS (s k-NN, Naive Bayes, k-means and k-medoids: classify traffic flows as normal and abnormal) and Advanced IDS (check packets with anomalous behaviors)
- Recurrent NN and convolutional NN: an input layer, a forward recursive layer, a reverse recursive layer, a fully connected hidden layer and an output layer. → feature reduction and DDoS attack detection.

- SDN-based DDoS detection system: extracts 68 flow features from the collected network traffic, including 34 features from TCP flows, 20 features from UDP flows, and 14 features from ICMP flows → apply deep learning model

4) Others

- HMM is applied to capture state information of user behaviors and identify whether a network connection is legitimate or not → if not, firewall can block access of that connection in time

5) Analysis

Finegrained intrusion detection requires a more complex labeled training dataset.

KDD99 and NSL-KDD (better): two widelyused datasets for IDS research.

Deep learning is a widely-used feature extraction & better due to its multiple-level feature representation capability.

VI. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

A. High-quality Training Datasets

Relationship between the training dataset size, the network characteristics, and the performance of the machine learning models

Public datasets: initiate activities for the publication of datasets in networking AI field.

B. Distributed Multi-controller Platform

- Increasing of network size and the number of flows: controller faces scalability issue
- Distributed multi-controller platform: composed of a logically centralized root controller (global view) and several local controllers (only control part of switches)
- RL models: guide the local controllers to process inter-domain traffic flows directly
- Reability: use RL algorithm to maximize network utility by selecting an optimal controller to process the flow table update request.

C. Improving Network Security

- Generative Adversarial Network (GAN): solve the problem by predicting new attacks.
- Comprised of two neural networks. Generator: generates new data, discriminator: evaluating the new data for authenticity according to real training dataset.

D. Cross-layer Network Optimization

- SDN, controller: collect the cross-layer information from all different layers

E. Incrementally Deployed SDN

- Only parts of network traffic is controlled by the controller: how to perform effective traffic engineering and optimize resource allocation

- SDN controller communicates with other traditional network nodes to exchange link weights, available bandwidth and topology information

VII. SOME BROADER PERSPECTIVES

A. Software Defined Network Function Virtualization

- Network Function Virtualization (NFV): virtualizing network functions and decoupling network functions from the underlying specialized hardware. → automate network configuration, provision and management.
- Markov Decision Process (MDP) and Bayesian learning method: optimize the dynamic resource allocation for NFV components
- Service Function Chaining (SFC) in NFV: create service chains dynamically based on the resource usage condition to enable efficient service providing.

B. Software Defined Edge Computing

- Role: offloading of computation-intensive tasks from the less capable devices to the powerful edge servers
- Supervised learning algorithms, especially neural networks: obtain the optimal heuristic-like offloading decisions

C. Software Defined Optical Networks

- Enable networking applications to use underlying optical network infrastructure dynamically and efficiently, by leveraging the ability of logically centralized control in SDN.
- Improve the design and planning of optical networks: QoT prediction (use k-NN and random forest) prior to the deployment of a new lightpath
- Fault management: statistical machine learning techniques are used for failure localization by predicting each link's failure probability

D. Software Defined Internet of Things

- Internet of Things (IoT) architecture: use ML models
 - Reduce network bandwidth consumption and improve the system response time: edge computing is often used to pre-process raw data
- Network security: deep NN is used to perform anomaly detection at edge servers.

E. Software Defined Vehicular Networks

- Provide comfort and convenience for drivers and passengers, [improve traffic efficiency & enhance vehicle road safety](#) by sharing real-time traffic information among vehicles
- Unicast, multicast and broadcast are three routing approaches
 - Multicast and broadcast are not energy efficient
 - Unicast: divided into two categories: topologybased (data forwarding decisions based on link information among vehicles) and position-based routing (decrease the communication hops by selecting the neighbor vehicle that is closest to the destination vehicle)

- ML techniques such as LSTM can be used to predict the movement of vehicles and the topology change of vehicular networks

F. Software Defined Mobile Networks


- Meet challenges from the heterogeneous wireless environments, the complexity of network management, the increasing mobile traffic demand and diverse service requirements in 5G mobile networks: ML techniques have been utilized to deploy more intelligence in mobile

RESEARCH AND NOTES

Bài báo (survey):

https://hustedu.vn-my.sharepoint.com/:b:/g/personal/trang_lh205234_sis_hust_edu_vn/Efy89eDQ9J1K1ldjqFUfJ98BoPPmeUhQYSzoCo3jxr3nxw?e=owxYoX

Tham khảo:

GitHub - khanhngocdam/IOT23: Network intrusion detection with machine learning
Network intrusion detection with machine learning - GitHub - khanhngocdam/IOT23: Network intrusion detection with machine learning
 <https://github.com/khanhngocdam/IOT23?fbclid=IwAR2RnrAzOHhPfbUCN2v2Rixy3XLgdRFe3iQVx6stqAqJEge2dAoVZ-WfR0>

khanhngocdam
IOT23
Network intrusion detection with machi
Contributors 0 Issues 0 Stars 0

FEEDBACKS

Gợi ý của thầy

- Node mạng biên: Vấn đề định tuyến (user-host) ⇒ định tuyến đúng từ server đến host (travel packet)
- Add mạng biên: giả lập server, thiết kế topology
- Lấy nguồn traffic follow ntn: dịch chuyển trong mạng (source → destination), bắt traffic của các dịch vụ khác (dùng bộ dữ liệu, dịch vụ mới) → dataset đang giới hạn (bắt các dịch vụ toàn cầu)



Chú ý:

- Tìm hiểu offloading factors cho mạng biên như thế nào
- Xây dựng topology
- Xây dựng dữ liệu

Công việc hoàn thiện

- ☒ Đọc bài survey, tìm hiểu
- ☒ Cài đặt liên quan
- ☐ Cài đặt mininet: Giả lập một mạng thiết lập giữa mininet với onos
- ☐ Đọc 2 bài báo về triển khai mạng, cách định tuyến routing