International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3180

(02/2022)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Future networks

Mechanism of traffic awareness for application-descriptor-agnostic traffic based on machine learning

Recommendation ITU-T Y.3180

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	**Y.3000–Y.3499**
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3180

Mechanism of traffic awareness for application-descriptor-agnostic traffic based on machine learning

Summary

Recommendation ITU-T Y.3180 specifies the mechanism of traffic awareness for application-descriptor-agnostic traffic based on machine learning. This Recommendation specifies the following aspects related to traffic awareness for application-descriptor-agnostic traffic including an overview, general mechanism, used machine learning methods, implementation consideration based on machine learning, report and auxiliary control mechanism for the malicious application-descriptor-agnostic traffic and security considerations.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3180	2022-02-13	13	11.1002/1000/14856

Keywords

Application descriptor agnostic, machine learning, traffic awareness.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3180

Mechanism of traffic awareness for application-descriptor-agnostic traffic based on machine learning

1 Scope

This Recommendation specifies a mechanism of traffic awareness for application-descriptor-agnostic traffic based on machine learning. The scope of this Recommendation includes:

a) Overview of traffic awareness for application-descriptor-agnostic traffic;

b) General mechanism of traffic awareness for application-descriptor-agnostic traffic;

c) Machine learning methods adopted to traffic awareness for application-descriptor-agnostic traffic;

d) Implementation considerations of traffic awareness for application-descriptor-agnostic traffic based on machine learning;

e) Report and auxiliary control mechanisms for the malicious application-descriptor-agnostic traffic;

f) Security considerations.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.200] Recommendation ITU-T X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic reference model: The basic model*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.2704] Recommendation ITU-T Y.2704 (2007), *Security mechanisms and procedures for NGN*.

[ITU-T Y.2770] Recommendation ITU-T Y.2770 (2012), *Requirements for deep packet inspection in next generation networks*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application-descriptor (also known as application-level conditions) [ITU-T Y.2770]: A set of rule conditions that identifies the application (according to clause 3.2.1 of [ITU-T Y.2770]). This Recommendation addresses the application descriptor as an object in general, which is synonym to application-level conditions. It does not deal with its detailed structure such as syntax, encoding, and data type.

3.1.2 application tag [ITU-T Y.2770]: A unique name for an application which is used to indicate the application semantics and is typically used for reporting scenarios.

3.1.3 machine learning (ML) [b-ITU-T Y.3172]: Processes that enable computational systems to understand data and gain knowledge from it without necessarily being explicitly programmed.

3.1.4 network element [b-ITU-T M.60]: It consists of telecommunication equipment (or groups/parts of telecommunication equipment) and support equipment.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 application-descriptor-agnostic traffic: Traffic that cannot be identified by a set of rule conditions or the set of rule conditions corresponding to the traffic are much difficult to be set up.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI	Artificial Intelligence
DL	Deep Learning
DPI	Deep Packet Inspection
FI	Flow Identifier
FF	Flow Feature
GMM	Gaussian Mixture Model
HCA	Hierarchical Cluster Analysis
HSTS	HTTP Strict Transport Security
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP over SSL
IP	Internet Protocol
kNN	k-Nearest Neighbour
ML	Machine Learning
NFV	Network Function Virtualization
NMS	Network Management System
P2P	Peer-to-Peer
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TA	Traffic Awareness
TCP	Transport Control Protocol
TLS	Transport Layer Security
VLAN	Virtual Local-Area Network
VM	Virtual Machine
VoIMS	Voice over Integrated Media System
VoIP	Voice over Internet Protocol

VoLTE Voice over Long-Term Evolution

VoNGN Voice over Next Generation Network

VoP2P Voice over Peer to Peer

XML extensible Markup Language

5 Conventions

In the body of this Recommendation, the words should, and can sometimes appear, in which case they are to be interpreted, respectively, as is recommended, and is/are able to. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative are to be interpreted as having no normative intent.

6 Overview of traffic awareness for application-descriptor-agnostic traffic

An application-descriptor is a set of rule conditions that can identify an application, see [ITU-T Y.2770]. In network scenarios, application-descriptor is a set of rule conditions that can identify a certain packet flow or a certain class of network traffic.

Application-descriptor-agnostic traffic is the class of network traffic which cannot be identified or classified based on an application descriptor, e.g., compressed traffic, various peer-to-peer (P2P) traffic, encrypted traffic, etc.

Because an application descriptor cannot be used to classify it, application-descriptor-agnostic traffic is difficult to be identified through traditional application-descriptor-based technologies or methods such as deep packet inspection (DPI) related technologies.

So, new mechanisms are needed to identify application-descriptor-agnostic traffic, In addition, the aforementioned mechanism should not only rely on traffic features that come from payload of the packet directly. Traffic features from payload are necessary information processed by traditional application identification technologies.

Machine-learning based mechanisms are appropriate for identifying application-descriptor-agnostic traffic because they can work without depending on payload of the packet. Through analysis for multi-dimension features other than packet payload, machine-learning based mechanisms can build up connections between multi-dimension features and a certain kind of network traffic, then machine-learning based mechanisms can identify the network traffic according to the multi-dimension features.

In order to provide identification capability for application-descriptor-agnostic traffic, the following functional elements should be supported by a machine-learning based mechanism:

- Designing and deploying machine learning function.
- Flow feature picking up, designing and representing.

In addition, the following functional elements are also important to the machine-learning based mechanism:

- Machine learning methods that can be used by the machine-learning based mechanism.
- Implementation methods of the machine-learning based mechanism.
- Report and control mechanisms.

7 General mechanism of traffic awareness for application-descriptor-agnostic traffic

7.1 Basic architecture of traffic awareness for application-descriptor-agnostic traffic

7.1.1 Overview of traffic awareness combined with machine learning technologies

It can be seen from clause 6 that it is very difficult to identify application-descriptor-agnostic traffic through traditional traffic awareness methods. In addition, machine learning technologies are quite appropriate for the aforementioned awareness of application-descriptor-agnostic traffic. An important aspect of the machine-learning based traffic awareness mechanism is the designing and deploying machine learning function.

In other words, the critical problem for machine learning technologies to be used in traffic awareness is how to deploy a machine learning function in the traffic awareness context. Generally, there are three kinds of deployment modes:

- traffic awareness with independent machine learning system;
- traffic awareness with embedded machine learning system;
- traffic awareness with hybrid machine learning system.

7.1.2 Architecture of traffic awareness with independent machine learning system

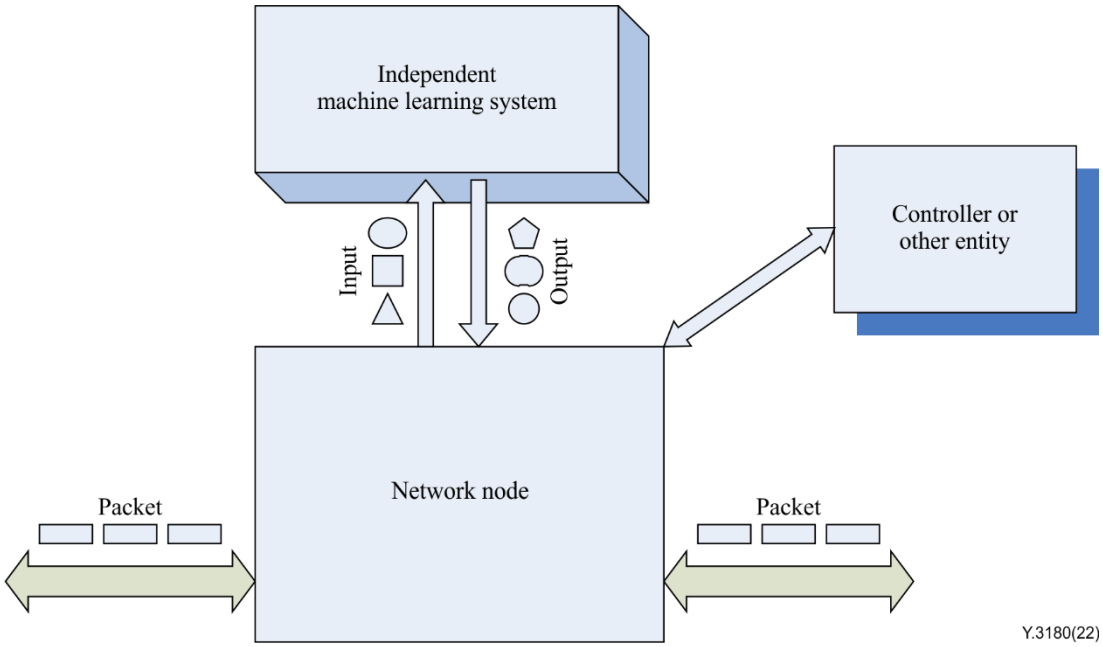

Figure 7-1 – Architecture of independent machine learning system

Figure 7-1 describes a typical architecture for deploying machine learning functions as an independent machine learning system. This deployment mode is proper for the majority of the application scenarios, and especially proper for network nodes that have been fully designed and developed. Network nodes mean the physical network elements used in the network. While deploying the machine functions function as an independent system, the basic structure of the network nodes which are going to be deployed with machine learning functions can remain unchanged.

Because the basic structure of the network nodes remain unchanged, only small and minor changes can occur to the network nodes. Under such circumstances, a small agent is needed by the network node to exchange input and output information with the independent machine learning system.

The input between network nodes and an independent machine learning system includes:

- Machine learning rules: some specification for the independent machine learning system, and the machine learning rules is used to tell the independent machine learning system how to use the input information to generate output information. It can be originated from the network controller or the network node.
- Training data: a data set including input data, decision and evaluation score, The evaluation score is the evaluation result for the decision. The training data can be originated from the controller or the network node. It can also be generated by the independent machine learning system itself.
- Input data: some data provided to the independent machine learning system in order to acquire some corresponding decision. It is originated from the network node.

The output refers to the decision information that can make the network node undertake some special actions.

Logically, there are three external interfaces (except for the interfaces between the network node and the independent machine learning system) concerning the network node:

- Packet in interface: a logical interface used for receiving data packets, generally, there is a packet in interface corresponding to each physical interface in the network node.
- Packet out interface: a logical interface used for sending data packets, generally, there is also a packet out interface corresponding to each physical interface in the network node.
- Interface to controller: a logical interface used to exchange information with the controller in order to carry out the control functions for the network node.

7.1.3 Architecture of traffic awareness with embedded machine learning functions

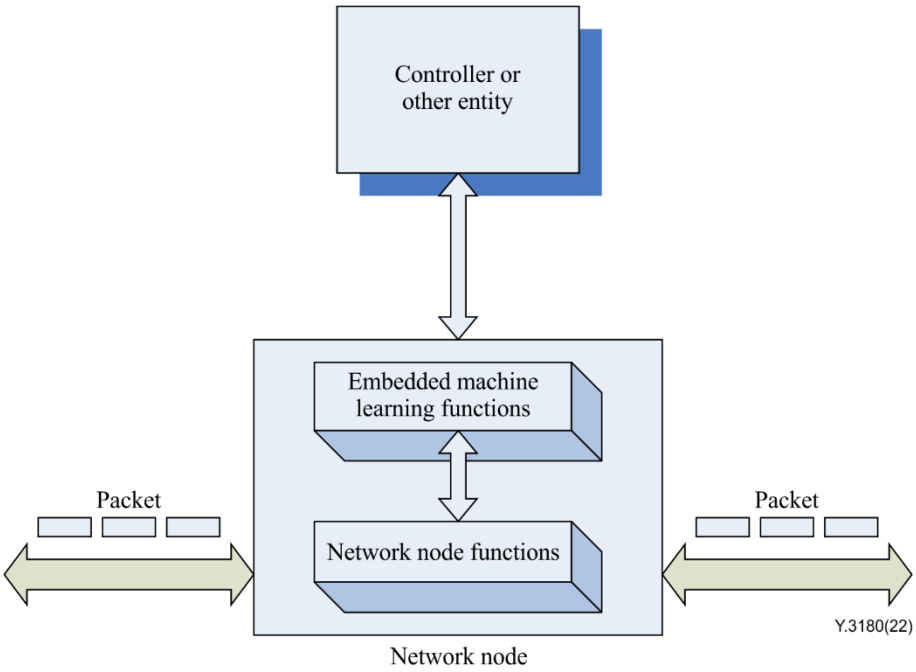

Figure 7-2 – Architecture of embedded machine learning functions

Figure 7-2 illustrates a kind of architecture for deploying machine learning functions in embedded mode. This deployment mode is appropriate for the network nodes that are under developing and can be restructured. While deploying machine learning functions in embedded mode, there are substantial changes concerning the basic structure of the network node.

Logically, there are also three external interfaces concerning the network node, and those external interfaces are the same as the external interfaces described in clause 7.1.2.

7.1.4 Architecture of traffic awareness with hybrid machine learning system

Figure 7-3 illustrates another kind of architecture for deploying machine learning functions in hybrid mode. That is to say, machine learning functions are deployed by combining independent machine system and embedded machine functions. The advantages of this deployment mode are that the two-level machine learning function will be more powerful and flexible.

As to this deployment mode, the level 2 machine learning system is functionally equivalent to an independent machine learning system while the level 1 machine learning system is functionally equivalent to the embedded machine learning system.

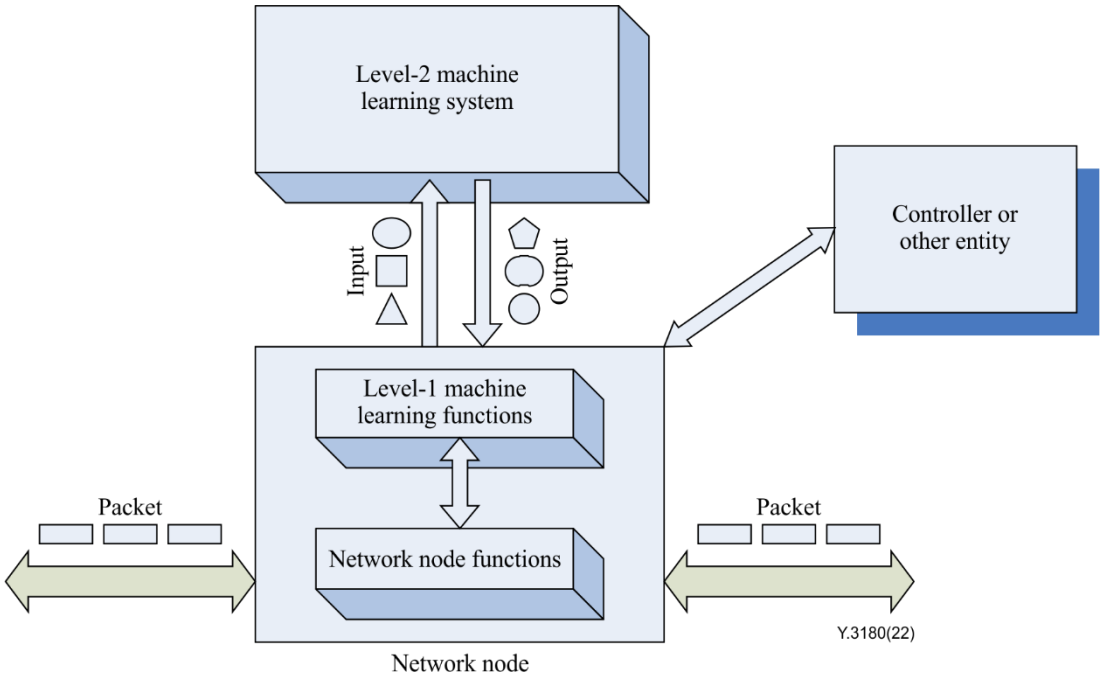

Figure 7-3 – Architecture of hybrid machine learning system

Logically, there are also three external interfaces (except for the interfaces between the network node and the independent machine learning system) concerning the network node, and these external interfaces are the same as the external interfaces described in clause 7.1.2.

7.2 Architecture for independent machine learning system

7.2.1 Overview of independent machine learning system

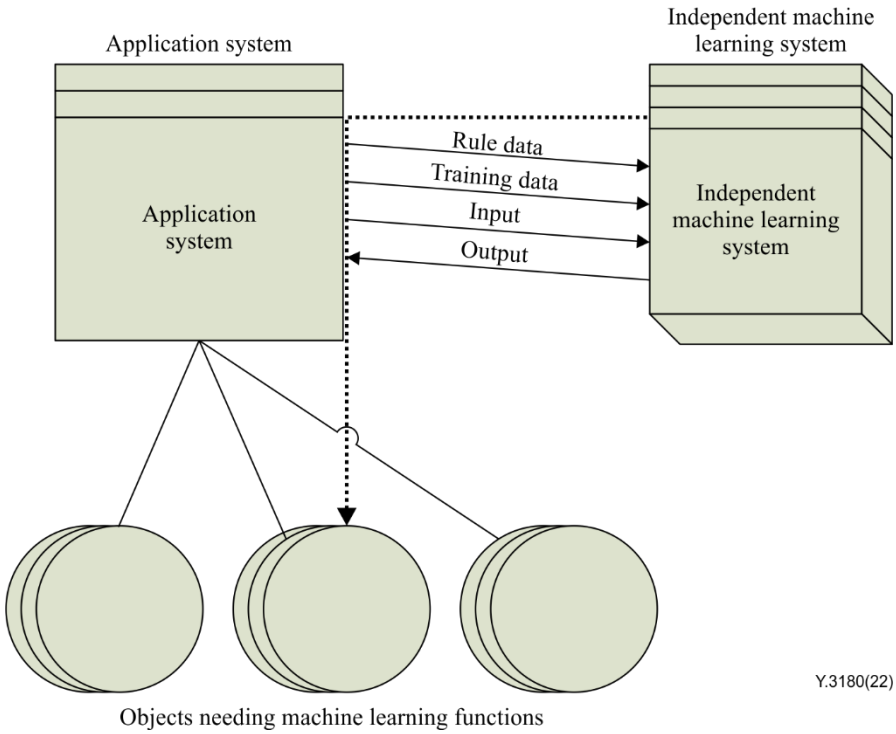

Figure 7-4 – Model for relations between application system and independent machine learning system

Figure 7-4 depicts a model of how the independent machine learning system cooperates with the application system that uses the machine learning functions. An application system is a system that needs to use machine learning functions, for example, the network node described in clause 7.1.

An independent machine learning system means machine learning functions are completely or partially implemented by an independent entity. The independent machine learning system accepts some rules and training data from outside of the independent machine learning system, and generates new training data based on the aforementioned rules and data. Each training data includes a group of input data, decision information and status information, and the optimal training data corresponding to the input data is stored as knowledge. When the input data enters the independent machine learning system, the independent machine learning system chooses the optimal knowledge as the origin of decision information and outputs the decision information in the knowledge.

Generally, the rule information and training data have a similar structure and they include the following information:

- input information;
- decision information;
- status information.

The input information for the rule and the training data includes a group of data types.

The decision information includes a group of operation types and one or more operation options corresponding to the operation types.

The status information includes one or more status types and weight/evaluation scores of every status type. Weight and evaluation scores share the same field and they are used by machine learning rules and training data respectively. When the status information corresponds to a machine learning rule, weight is used to represent the influence factor of the status type for the final evaluation result. When the status information corresponds to a training data, the evaluation score is used to represent the evaluation result of the status type for the decision information in the training data.

For example, as to machine learning rule R has a group of status types S1, S2 … and Sn, then S1, S2 … and Sn has a weight of W1, W2 … and Wn respectively. W1, W2 … and Wn are non-negative real numbers and should obey the following formula:

$$W1 + W2 + \ldots + Wn = 1$$

On the other hand, as to training data T has a group of status types S1, S2 … and Sn, then S1, S2 … and Sn has evaluation scores E1, E2 … and En respectively. E1, E2 … and En are real numbers between 0 and 100: En represents the evaluation score about status type Sn to the decision information in T.

Clause 7.2.4 provides a detailed illustration of machine learning rules and training data.

It is noted that the aforementioned level 2 machine learning system has the same structure and function as the independent machine learning system.

Figure 7-4 describes how the application system uses the machine learning functions provided by the independent machine learning system, and can be summarized in the following steps:

1) The application system connects with the independent machine learning system.
2) The application system defines some machine learning rules for the independent machine learning system.
3) The independent machine learning system updates the rules database.
4) The application system sends some training data based on machine learning rules to the independent machine learning system.
5) The independent machine learning system stores the training data and generates more training data based on simulations and existing training data.
6) The application system sends some input data to the independent machine learning system.
7) The independent machine learning system evaluates the training data and acquired corresponding knowledge.
8) The application system sends some input data to the independent machine learning system.
9) The independent machine learning system stores the input data.
10) The independent machine learning system looks up the knowledge based on the input data and acquired decisions corresponding to the input data.
11) The independent machine learning system sends the decisions to the application system.

7.2.2 Logical structure of independent machine learning system

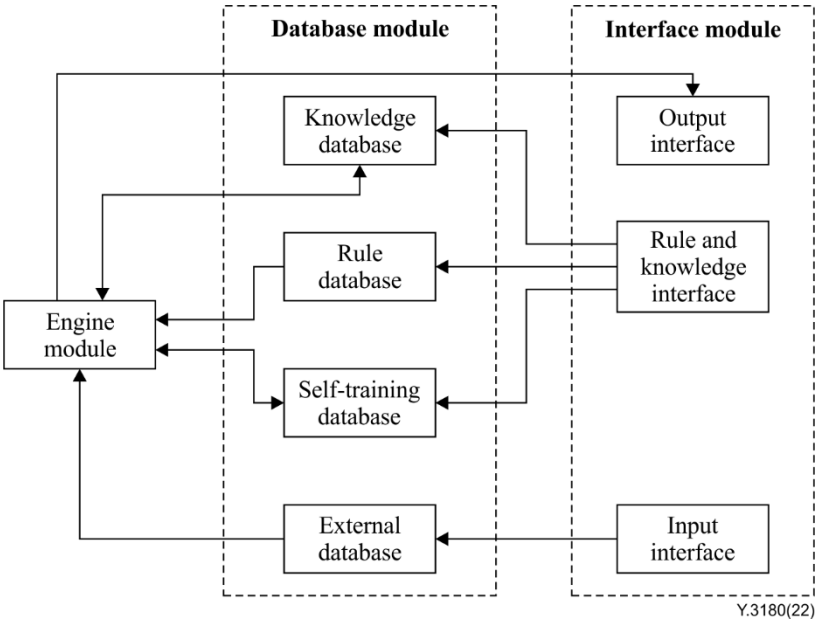

Figure 7-5 – Logical structure of the independent machine learning system

Figure 7-5 describes the logical structure of the independent machine learning system. From Figure 7-5, it can be seen that the independent machine learning system is comprised of three main blocks:

- Interface module: used to provide the interface for external communication. That is to say, the interface module receives machine learning rules and training data from outside of the machine learning system, and also receives input data which needs a corresponding decision.
- Machine learning engine module: used to implement machine learning algorithms. The machine learning engine module can generate new training data based on received machine learning rules and training data. Then, it can acquire knowledge based on training data. When the interface module receives input data which needs a decision, it will choose the optimal knowledge and output the optimal decision information.
- Database module: used to store various data used by machine learning functions. The stored data includes machine learning rules and training data. The machine learning rule includes input information, decision information and status information. The input information includes multiple data types. The decision information includes multiple action types and each action type includes one or more action options. The status information of the machine learning rule includes multiple status types and every status type has a weight that can be used to make decisions.

As to training data, the input information in it includes a group of data types. The training data also includes decision information and status information. The decision information includes a group of action types and every action type has a determined action option. The status information includes a group of status types corresponding to the input information and every status type has a corresponding evaluation score (evaluation result corresponding to the status type).

The process for the machine learning engine module to generate new training data based on received machine learning rules and training data includes the following two steps: First, simulated data is automatically generated based on the machine learning rule, and the simulated data has the same structure as the training data, but the simulated data does not have a determined evaluation

score. Then, based on the evaluation score of the received training data, the simulated data can get the evaluation score through self-learning and become a new training data.

The machine learning engine module chooses the optimal training data corresponding to a set of input information as the acquired knowledge and stores the knowledge in the corresponding data base. This process includes three steps: First, calculating the evaluation score of every status type through the evaluation score of the status type multiplying its status weight (e.g., En multiplies Wn); Then, the general evaluation score corresponding to the training data is acquired by summing up the evaluation score of all status types; Finally, the training data with the highest general evaluation score is chosen as the knowledge.

7.2.3 Interface specification for independent machine learning system

From clause 7.2.1, it can be seen that there are three interfaces within the independent machine learning system, and the following describes the specifications for those interfaces:

Rule and knowledge interface: the application system that needs machine learning functions defines machine learning rules through the interface to the independent machine learning system. In addition, the application system sends external historical knowledge (for example, knowledge from other networks) through the interface to the independent machine learning system. Moreover, the application system sends training data through the interface to the independent machine learning system.

Input interface: the application system sends input data that needs decisions through the interface to the independent machine learning system.

Output interface: the independent machine learning system sends decisions corresponding to the input data to the application system through the interface.

All the three interfaces can be independent physical interfaces or logical interfaces virtualized by an identical physical interface.

7.2.4 Information structure for independent machine learning system

7.2.4.1 Information structure of machine learning rule

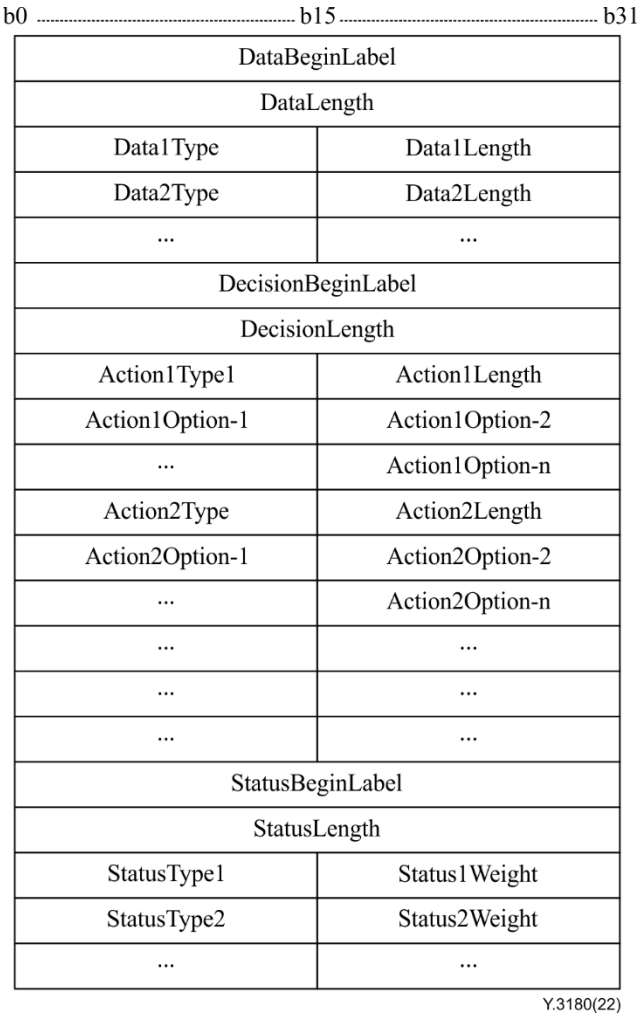

Figure 7-6 – Information structure of machine learning rule

Figure 7-6 illustrates the information structure of the machine learning rule. The information structure includes the following three parts:

- Input data structure: defines structure for input data that needs decisions, and it includes a group of data with various data types and various data lengths. In Figure 7-6, input data refers to the data from 'DataBeginLabel' to 'DecisionBeginLabel'.
- Decision structure: defines the structure for the decision that needs generating by the independent machine learning system. It should be emphasized that the decision corresponds to the input data and the decision comprises one or more actions. In addition, every action has one or more action options. In Figure 7-6, decision information refers to the data from 'DecisionBeginLabel' to 'StatusBeginLabel'.
- Status structure: defines the structure for status information used to evaluate the decision. There are one or more status items with various status types and different weights. In Figure 7-6, status information refers to the data from 'StatusBeginLabel' to the end of the rule.

7.2.4.2 Information structure of training data

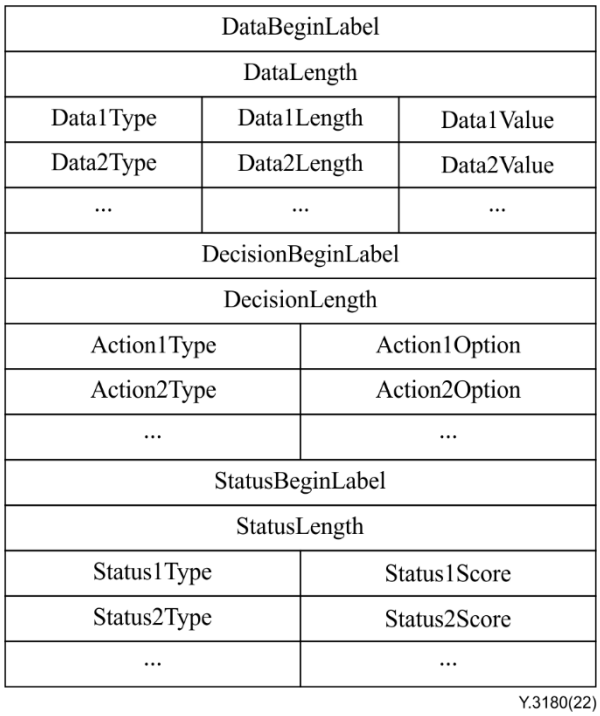

Figure 7-7 – Information structure of training data

Figure 7-7 illustrates the information structure of training data. The information structure is similar to the information structure of the machine learning rule and also includes the following three parts:

- Input data: input data that needs a decision, and it includes a group of data items and each data item includes data type, data length and data value. In Figure 7-7, input data refers to the data from 'DataBeginLabel' to 'DecisionBeginLabel'.
- Decision information: the action group corresponds to the input data and the decision comprises of one or more actions. In addition, every action has one deterministic option. In Figure 7-7, decision information refers to the data from 'DecisionBeginLabel' to 'StatusBeginLabel'.
- Status information: is used to evaluate the decision information. There are one or more status items and every status item has an evaluation score. The general evaluation score is the sum of the evaluation score of every status item multiplying weight of the status item. In Figure 7-7, status information refers to the data from 'StatusBeginLabel' to the end of the training data.

7.3 Protocol layer mechanism of traffic awareness for application-descriptor-agnostic traffic

Protocol layer means the lower-layers of the protocol stack, see OSI/ISO [ITU-T X.200], that is to say the protocol layer includes the layers from layer 2 to layer 4, and the protocol layer features involve all layer 2, layer 3 and layer 4 headers. For example, if layer 2 is Ethernet, layer 3 is the IP layer, layer 4 is the TCP layer, these protocol layer features include the layer 2 destination address, the layer 2 source address, type/priority, virtual local-area network (VLAN), the layer 3 source address, the layer 3 destination address, protocol type, TCP source port, TCP destination port, etc.

Together with other traffic features, the protocol layer features can be used to identify the traffic type.

In order to use the protocol layer features, it is necessary to represent the protocol layer features as a certain formation in order to facilitate the following processing. A feature set [f_1, f_2, … f_n] is used to identify the protocol layer features. Every member from the set is a type-length-value triple tuple. The aforementioned type is a layer 2, layer 3 or layer 4 header field (e.g., layer 3 destination address, layer 3 source address, etc.), the aforementioned length is number of bytes needed to represent the header field and the aforementioned value is the concrete data for the header field.

So, a complete protocol feature set F[f_1 (f_{1t}, f_{1l}, f_{1v}), f_2(f_{2t}, f_{2l}, f_{2v}), … f_n(f_{nt}, f_{nl}, f_{nv})] can be used to describe the protocol layer feature corresponding to the packet, so F can be thought of as the packet-level protocol feature set. Then, F1, F2 … and Fn can be used to describe the protocol layer features of packet 1, packet 2… and packet n of the network flow respectively, and [F1, F2, … Fn] is called the flow-level protocol feature set.

Generally, based on the flow-level protocol layer feature set, the network traffic can be divided into the following classes described in clause 7.3.1, clause 7.3.2 and clause 7.3.3.

7.3.1 Single flow-level protocol feature set traffic

Sometimes the network traffic has only one network flow that can be identified by a single flow-level protocol feature set. This kind of network traffic is easily represented and processed as to the protocol layer feature.

7.3.2 Multiple flow-level protocol feature set traffic

Usually the network traffic has more than one network flow, and the multiple flow-level protocol feature sets are needed to identify this network traffic. It is noted that these protocol feature sets are independent each other.

7.3.3 Protocol and data separated traffic

There are some other kinds of network traffic which consist of two or more flows, and it is possible that one (or more) flow is used for protocol negotiation meanwhile another flow is used for data transportation. These kinds of traffic are called protocol and data separated network traffic.

Under such circumstances, it is much more difficult to identify the traffic type. Because protocol features of the data transportation flow are from the protocol negotiation flow, the aforementioned protocol feature would change with time.

Concerning these kinds of network traffic, multiple flow-level protocol feature sets are also necessary to identify them. In addition, the flow protocol feature set related to the data flow is dependent on the flow protocol feature set related to the protocol flow.

7.4 Application layer mechanism of traffic awareness for application-descriptor-agnostic traffic

The application layer means the higher layer of the protocol stack, and generally the application layer is on the top of layer 4. Application layer features are those features from the application layer, in other words, application layer features come from the payload of the packet.

As to application-descriptor-agnostic traffic, because there is usually not a deterministic data format to be used to describe the application layer, so it is much difficult to use a data set similar to the feature set for the protocol layer to identify the application layer.

However, some statistic features can be used to identify the application features, clauses 7.4.1, 7.4.2 and 7.4.3 present several commonly-used probability statistic features.

7.4.1 Time related features

Application layer data transportation is a time related sequence, and transportation of every block of application data has a time related parameter. All of the time parameter can be used to form the time related features.

7.4.2 Space related features

Application layer data transportation is also space related, in other words, every block of application data would possibly need deterministic store space. So, every block of application data has a space related parameter, and all of the space parameter can be used to form the space related features.

7.4.3 Protocol related application layer features

Sometimes some application protocols or rules are used to describe the application layer data, that is to say, application layer data transportation is also related to some application protocols (e.g., HTTP, Restapi, XML, etc.). Every network flow of application data would possibly use one or more special protocols. So, the aforementioned protocol can be used as protocol related features.

7.5 Reliability and availability mechanism for independent machine learning system

It can be seen from clauses 7.1 and 7.2 that the independent machine learning system is important for applying machine learning methods in identifying application-descriptor-agnostic network traffic.

On the one hand, the independent machine learning system brings many advantages such as strong processing capability, high flexibility and good extensibility. On the other hand, because the independent machine learning system is the core component and because the correct decision mainly comes from it, its reliability and availability should be carefully taken into account.

Redundant methods usually can help to improve the reliability and availability of a system. Certainly redundant methods can also be beneficial to the independent machine learning system. Redundant methods use two or more components that can realize the same functions respectively. For example, if each of component A, B and C can realize function F, all components are used to realize F. When a certain component A is out of order and output from it is wrong, outputs from the other components B and C can output the same correct result. It can guarantee the final output is correct by choosing the output from B and C.

There are mainly two classes of redundant methods as follows:

- Homogeneous redundant method;
- Heterogeneous redundant method.

As for the homogeneous redundant method, all components used in the method have an identical design.

On the other hand, all the components used in the heterogeneous redundant method have different designs. The heterogeneous redundant method is much better than the homogeneous redundant method, because the identical design means that all components can easily have the same mistakes. So, the heterogeneous redundant method is a good solution to improve the reliability and availability of the independent machine learning system.

As designs for components used in the heterogeneous redundant method are different, then, network function virtualization (NFV) technologies are very beneficial. It is difficult and uneconomical to design different hardware for those components. NFV can make full use of a generic hardware platform and facilitate every component to easily load different software. The following is the general design method to implement the heterogeneous redundant method based on NFV technologies:

- Designing a group of heterogeneous process methods, and those methods can be implemented in the virtual machines (VMs).
- Through machine learning methods, building up a knowledge base that is used to store links between inputs and process methods.
- As for a certain input, a set of proper process methods can be acquired by searching the aforementioned knowledge base.
- The set of proper process methods are used to handle the input and a group of outputs can be acquired.
- The best output can be chosen from the above output group.

It is noted that each of the aforementioned process methods is implemented in a VM and process methods implemented in different VMs are substantially different from each other. The selected set of process methods are partial of all process methods.

Deployment of VMs is not limited by physical systems or devices. In other words, a group of VMs can be deployed in a single physical machine, two or more physical machines or in a cloud-based context.

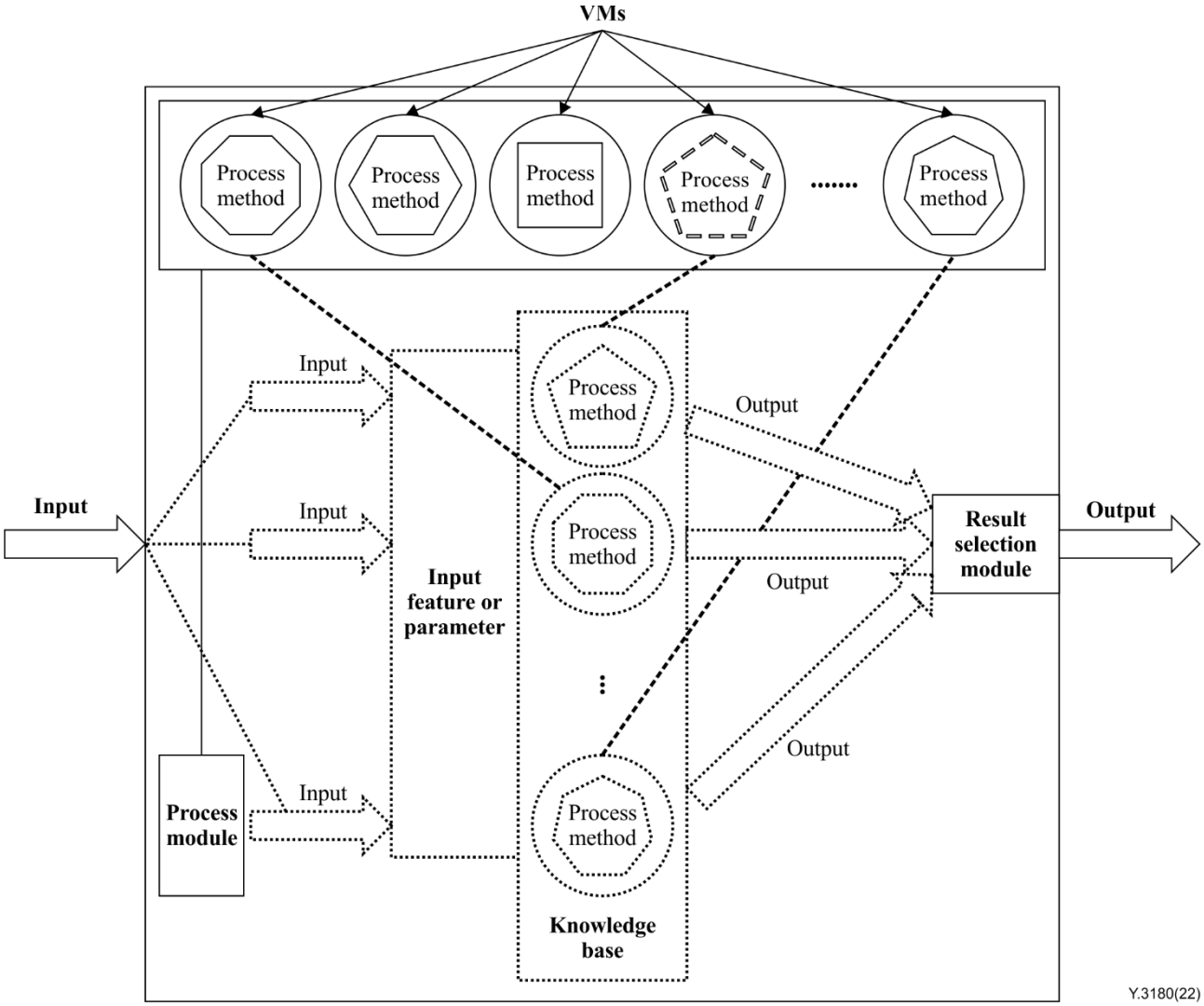

Figure 7-8 – Heterogeneous redundant method based on NFV

Figure 7-8 describes the above heterogeneous method. In the figure, there are a group of process methods that can run in VMs and can be selected based on input. The input will be sent to all

selected process methods and a group of outputs can be acquired. A result selection module is responsible for choosing the best output from the group of outputs.

8 Machine learning methods used for application-descriptor-agnostic traffic awareness

8.1 Overview of machine learning methods used for traffic awareness

There are many machine learning methods that can be used to implement traffic awareness functions. Both the independent machine learning system/level 2 machine learning system and the embedded machine learning system/level 1 machine learning system can implement these machine learning methods if its resources are enough to support running of these machine learning methods.

The following machine learning methods described in clause 8.2 and clause 8.3 are recommended to be used in both the independent machine learning system/level 2 machine learning system and embedded machine learning system/level 1 machine learning system. It is noted that candidate machine learning methods that can be used to implement traffic awareness functions are not limited to those methods described in clause 8.2 and clause 8.3.

8.2 Supervised learning methods

8.2.1 Support vector machine method applied to traffic awareness for application-descriptor-agnostic traffic

A support vector machine (SVM) is a supervised learning algorithm that can be used for binary classification or regression. A support vector machine constructs an optimal hyper-plane as a decision surface such that the margin of separation between the two classes in the data is maximized. Support vectors refer to a small subset of the training observations that are used as support for the optimal location of the decision surface.

If traffic awareness functions need to be realized based on support vector machine method, the following two premises should be taken into account:

- The result of traffic awareness method should be 'yes' or 'no'. In other words, what the traffic awareness method should do is not identifying what the traffic is but judging whether the traffic is type A or not.

The features used for training need to be determined. That is to say, every group of training data can make the traffic awareness method draw a clear conclusion: whether the traffic is A or not.

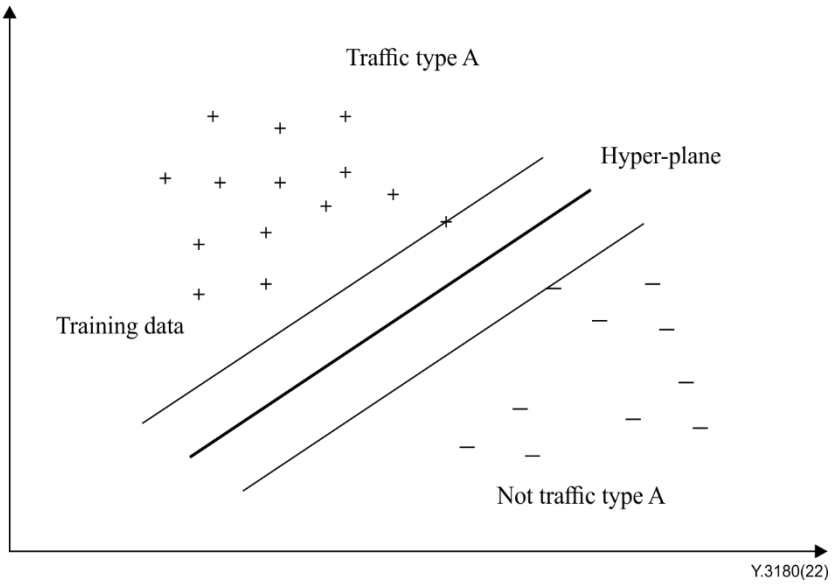

Figure 8-1 – Traffic awareness based on support vector machine

Figure 8-1 shows a diagram for traffic awareness based on the support vector machine method. After the above two premises are met, the following process can be used to realize the traffic type judging functions:

- Selecting a group of traffic features to construct a feature vector.
- Based on the traffic feature vector, simulating and generating a set of training data.
- Based on the training data, computing and finding a proper hyper-plane.
- Use real training data and simulated training data to make the hyper-plane evolve the optimal one.
- When the traffic is input, collecting the traffic features, judging whether the traffic type is A based on the traffic features.
- Confirming and verifying the judged result.
- Adjust the hyper-plane based on the confirmed result and loss computing.

8.2.2 Learning method base on k-nearest neighbours

The k-nearest neighbours algorithm (k-NN) is a non-parametric machine learning method which can be used in a classification method. The output of k-NN is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small).

It is sure that k-NN can be applied in traffic awareness when the candidate network traffic classes are known. When a certain network traffic enters the machine learning system, the machine learning method can find several traffic classes that the target traffic possibly belongs to, then the k-NN algorithm can be used to find the most appropriate traffic class.

8.2.3 Learning method base on Naive Bayes

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. They are among the simplest Bayesian network models but coupled with kernel density estimation, they can achieve higher accuracy levels.

The machine learning system for traffic awareness can also use the Naive Bayes based methods. A group of features of network traffic can be extracted and described as a number of variables by the machine learning system. When a certain network traffic enters the machine learning system, the machine learning can use the value of the variables from the network traffic to judge the appropriate traffic class.

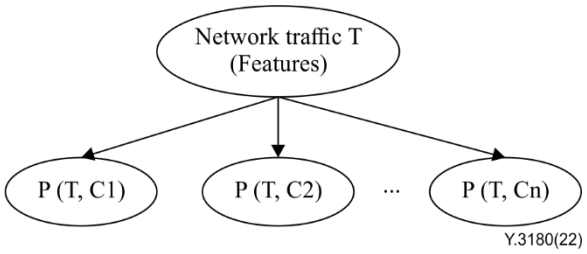

Figure 8-2 – Traffic awareness based on Naive Bayes

8.2.4 Learning method base on decision tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A decision tree based method can be also adopted by the machine learning system for traffic awareness. In reality, it can be thought of as a simple solution for the machine learning system.

An improved method for a decision tree is a random forest based method. It can be thought as an enhanced decision tree based method. Figure 8-3 shows traffic awareness based on a decision tree.

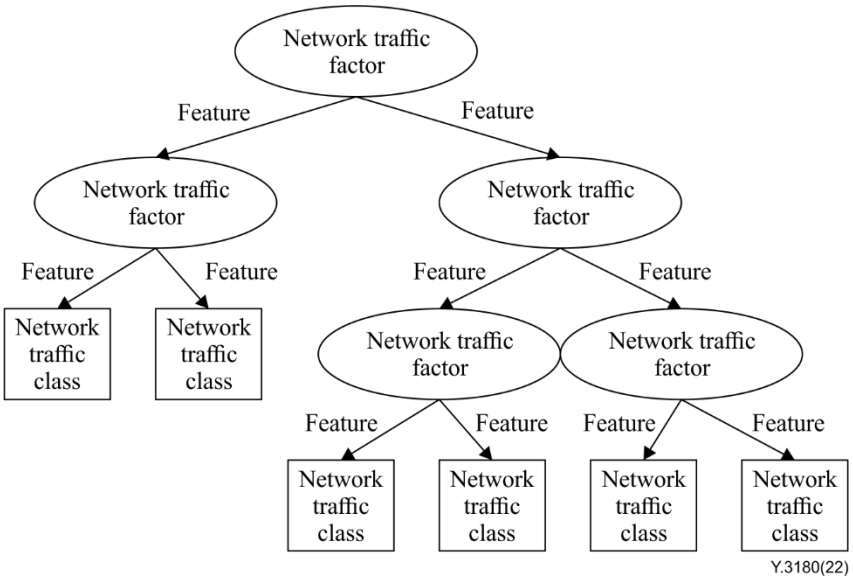

Figure 8-3 – Traffic awareness based on decision tree

8.2.5 Deep learning method applied to traffic awareness for application-descriptor-agnostic traffic

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised. Deep machine learning is the synonym of deep learning.

Deep learning is also a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input.

Different from the support vector machine method, deep learning can not only judge whether the traffic is type A or not but can also point out what the traffic type is.

Figure 8-4 specifies a general model for traffic awareness based on deep learning. The left layer is the input layer meanwhile the right layer is the output layer. All the other layers are hidden layers. To realize traffic awareness based on deep learning, the following process is necessary:

- Selecting a group of traffic features.
- Based on the traffic features, set up a deep learning model.
- Based on the traffic features, inputting or generating training data.
- Based on the training data, improve the deep learning model.
- When the traffic is input, collecting the traffic features.
- Inputting the traffic features into the deep learning machine.
- Deep learning machine outputs the identified traffic type.

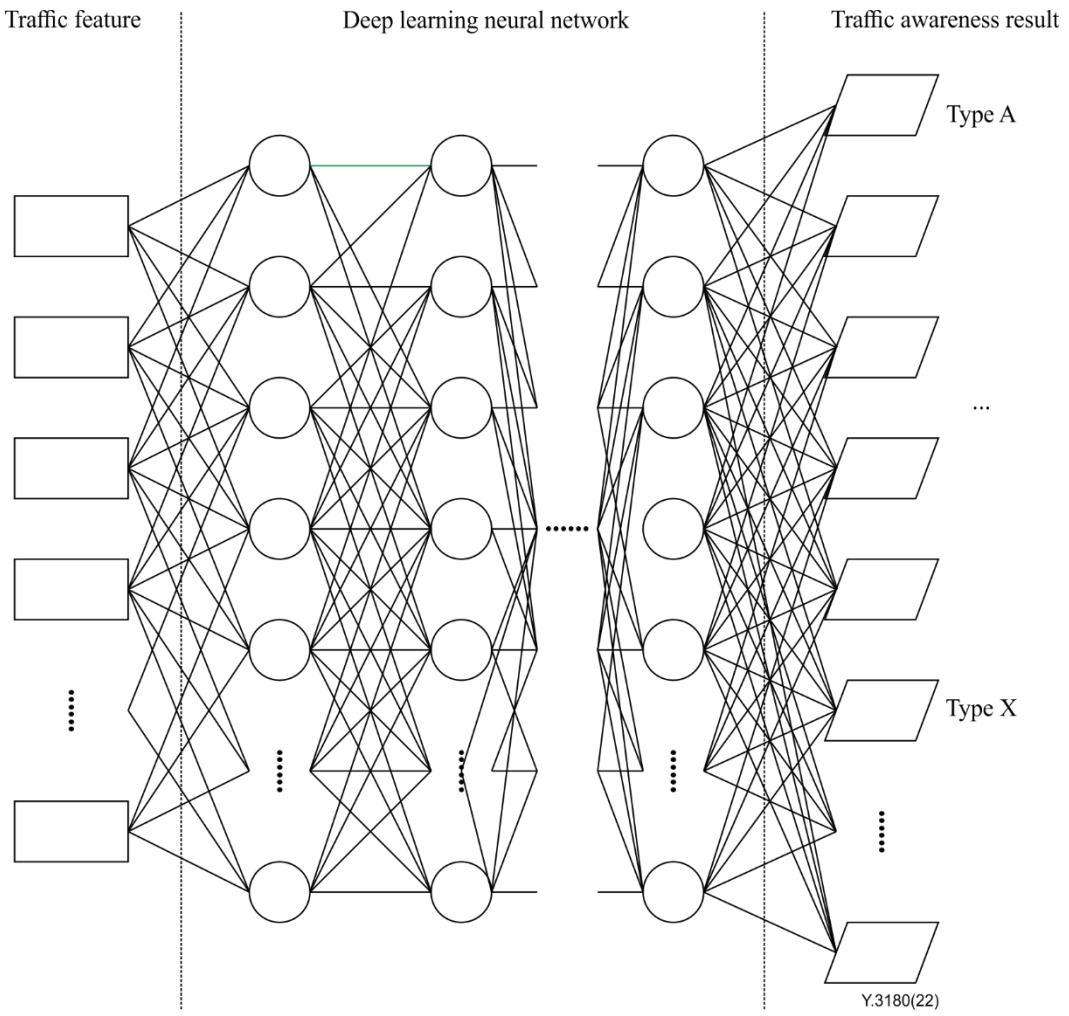

Figure 8-4 – Traffic awareness based on deep learning

8.3 Unsupervised learning methods

8.3.1 K-means clustering method

K-means clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centre or cluster centroid).

The machine learning system for network traffic awareness can use the K-means clustering method under such circumstances:

- The network traffic class space is unknown or the network traffic class space is not appropriate for the target network traffic to be identified.
- The feature of the network traffic can be quantization.
- The quantized features can be used to represent the network traffic accurately.

8.3.2 Hierarchical clustering method

Hierarchical clustering also called hierarchical cluster analysis or (HCA), is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

- Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

- Divisive: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

The machine learning system for traffic awareness can use this kind of method to build traffic hierarchy in order that the succeeding traffic awareness action can be based on supervised methods.

8.3.3 Gaussian mixture model method

A Gaussian mixture model is a probabilistic model for representing the presence of subpopulations within an overall population. Formally a Gaussian mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population.

When the machine learning system for traffic awareness cannot accurately identify a certain kind of network traffic, the Gaussian mixture model is the proper method to be applied. By initiating multiple Gaussian distributions and recursively adjusting the parameters of the Gaussian distribution, the more optimized result can be acquired.

9 Implementation consideration of traffic awareness for application-descriptor-agnostic traffic based on machine learning

9.1 Implementation for flow feature based methods

9.1.1 Overview of the flow feature based methods

From clause 6, it can be seen that it is very difficult to identify application-descriptor-agnostic traffic directly. Then, a feasible and appropriate method of traffic awareness is taking not a single packet but a group of correlated packets into account. In other words, not only packet features but also flow features are used to identify the network traffic. This is so called the flow feature based method.

The flow feature based method logically includes the following main functions:

- Feature extraction function.
- Feature synthesization function.
- Feature representation-transformation function.
- Feature data identification function.

9.1.2 Feature extraction function

Just like the description in clause 7.4, there are many classes of features about network traffic such as time-related features, space-related features, and so on.

It is noted some of the aforementioned features can be extracted directly from the packets such as protocol related features meanwhile some other features, such as time-based features, need to be acquired by extraction, transformation and statistics.

9.1.3 Feature synthesization function

Generally, not a single kind or class of the feature but rather all kinds or classes of the features should be taken into account to realize awareness of the network traffic, so, it is necessary to synthesize all kinds or all classes of the features. Through synthesization, a set of traffic features can be acquired. Table 9-1 is an example for the feature set, it is noted that not only those features listed in the table should be used, instead, all features related the network traffic can be taken into account.

Table 9-1 – An example for the synthesized feature values

Feature name	Feature identification	Feature value	note
Graded traffic feature	X01	Packet length/a constant N	Packet length measured by N bytes
Time distribution feature	X02	Packet number/Time	Packet number per millisecond
Layer four feature	X03	Protocol type of IP header	The l4 protocol of the TCP/ IP protocol stack
Source l4 port	X04	Source L4 port	The l4 source port of the TCP/ IP protocol stack
Destination l4 port	X05	Destination L4 port	The l4 destination port of the TCP/IP protocol stack
Source IP address	X06	CRC16 of source IP address	Computed CRC16 for source IP address
Destination IP address	X07	CRC16 of destination IP address	Computed CRC16 for destination IP address
L2 source address	X08	CRC16 of source layer 2 address	Computed CRC16 for source layer 2 address
L2 destination address	X09	CRC16 of destination layer 2 address	Computed CRC16 for destination layer 2 address
…	…	…	…
…	…	…	…

9.1.4 Feature representation-transformation

All data about traffic features described in clause 9.1.3 is diverse and hard to be processed by the machine learning methods described in clause 8. Therefore, it is necessary to transform those data to some representation that is easy to handle by machine learning methods

For example, if a machine learning method is targeted to image, it is feasible and useful to transform traffic feature data to an image.

The following is an example method to transform traffic feature data to an image. Each feature value illustrated in clause 9.1.3 can be represented by a binary data (for example, a 16-bit binary data). In the meantime, one point of an image can be represented by a binary data. Therefore, if all feature values represented with a binary data are combined, a similar image can be generated. Figure 9-1 is an sample of the aforementioned image.

X0101	X0102	X0103	X0104	X0105	X0106	X0107	X0108
X0109	X0110	…	…	X01N	PAD	PAD	PAD
X0201	X0202	X0203	X0204	X0205	X0206	X0207	X0208
X0209	X0210	…	…	…	X01M	PAD	PAD
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
Xj01	Xj02	Xj03	Xj04	Xj05	Xj06	Xj07	Xj08
Xj09	Xj10	…	…	…	…	XjL	PAD

Figure 9-1 – An sample for the image transformed from the traffic feature data

9.1.5 Feature data identification function

Through machine-learning related methods, we can set up the knowledge which describes the relationship between the feature data and the traffic class. When a certain network traffic needs to be identified, the traffic class can be acquired through this knowledge.

Table 9-2 describes an example of the aforementioned knowledge.

Table 9-2 – An example for the knowledge

Feature data description	Class of the network traffic	Possible correct rate
feature data 1	class A	95%
feature data 1	class B	60%
feature data 1	class C	30%
feature data 2	class A	50%
feature data 2	class C	70%
feature data 3	class D	80%
feature data 3	class E	60%
feature data 3	class F	90%
feature data 3	class G	95%
feature data 3	class H	30%
feature data 4	class G	80%
feature data 4	class H	70%
feature data 5	class D	90%
feature data 5	class E	100%
feature data 5	class F	85%
…	…	…
…	…	…

9.2 Implementation methods based on analysis for payload features

9.2.1 Methods based on analysis for live payload

Methods based on analysis for live payload are not the same as traditional payload based methods because they aim at identifying the traffic without an application descriptor. On the one hand, some application-descriptor agnostic traffic can be converted to application-descriptor-awareness traffic such as the situation where the traffic is encrypted but the key and the encryption algorithm are known. On the other hand, although the traffic cannot be converted to common traffic, some relevant data packets related to the traffic can be analysed so that the hand-shake protocol packets are common packets.

9.2.2 Methods based on analysis for stochastic feature of the payload

Methods based on analysis for stochastic feature of the payload do not use the payload directly, but they use the stochastic features (for example, probability of a data pattern, entropy of the payload in the packet) of the payload. The payload can be thought of as common data, through analysing stochastic attributions about the data, the corresponding traffic can be classified.

9.3 Implementation of methods based on analysis of traffic behaviour feature

9.3.1 Methods based on analysis for behaviour features of applications

The methods based on analysis for behaviour features of applications try to identify the network applications from the point of view of the host. The target of these methods is classifying the network traffic into known applications such as P2P, Web and so on. Realization of these methods is usually based on analysis for correlation of the host and the protocols used by the host.

9.3.2 Methods based on analysis for time-space distribution of packets

Methods based on analysis for time-space distribution of packets are blind to the contents of data packet or network traffic. The main information used by these methods is time-space distribution features of data packets. Those features include packet size and packet frequency, etc.

9.4 Hybrid methods based on analysis for multiple features

In reality, it is effective to identify the network traffic by using multiple methods simultaneously. For example, it is a good solution to synthetically use methods based on live payload and methods based on analysis for time-space distribution of packets.

10 Report and auxiliary control mechanism for the malicious application-descriptor-agnostic traffic

10.1 Report mechanism for malicious application-descriptor-agnostic traffic

When some application-descriptor-agnostic traffic is identified as malicious traffic such as a network attack, a network virus and so on, it is necessary to report the malicious traffic to corresponding functional entities such as controller or network management system (NMS).

The report process will be different when the deployment mode for the machine learning system is different.

10.1.1 Report based on independent machine learning system

Logically, the independent machine learning system can be thought to be located outside of the network, so it cannot usually report to the controller or other entities directly, though the malicious application-descriptor-agnostic traffic is discovered by the independent machine learning system.

The report process based on independent machine learning system is as follows:

First, the independent machine learning system sends the information about the malicious application-descriptor-agnostic traffic to a network node.

Second, the network node reports this information to the controller or other entities.

The red arrows in Figure 10-1 indicate the report process.

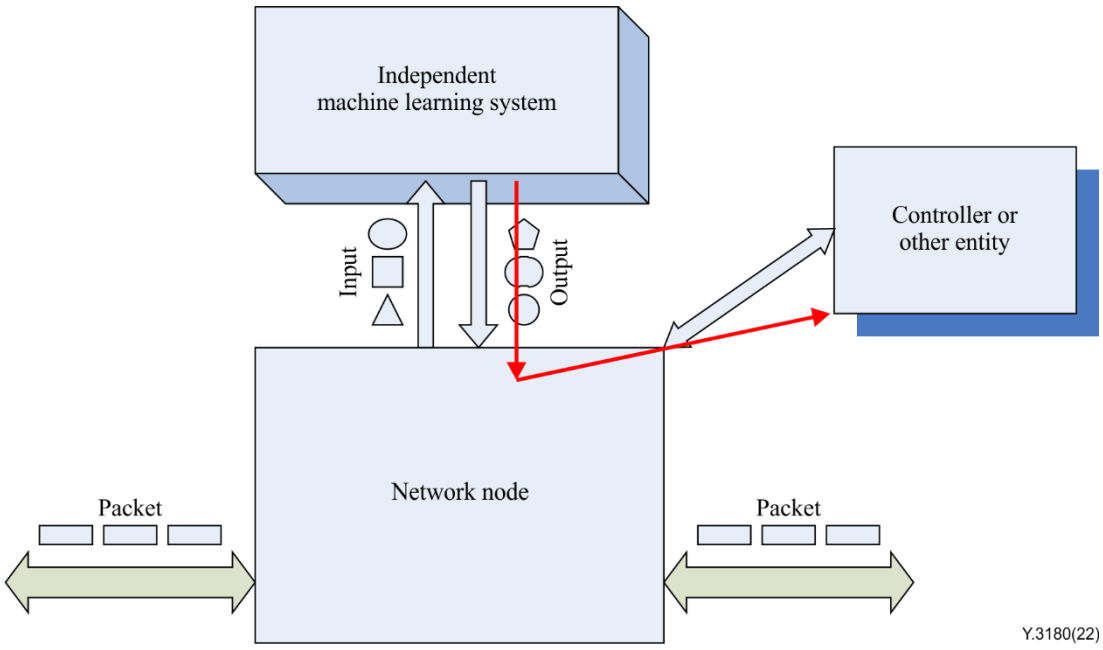

Figure 10-1 – Report process based on independent machine learning system

10.1.2 Report based on embedded machine learning system

Under such circumstances, the malicious application-descriptor-agnostic traffic is discovered by an embedded machine learning system within the network node itself. Then the embedded machine learning system can report the information to the controller or other entities directly. The red arrows in Figure 10-2 indicate the report process.

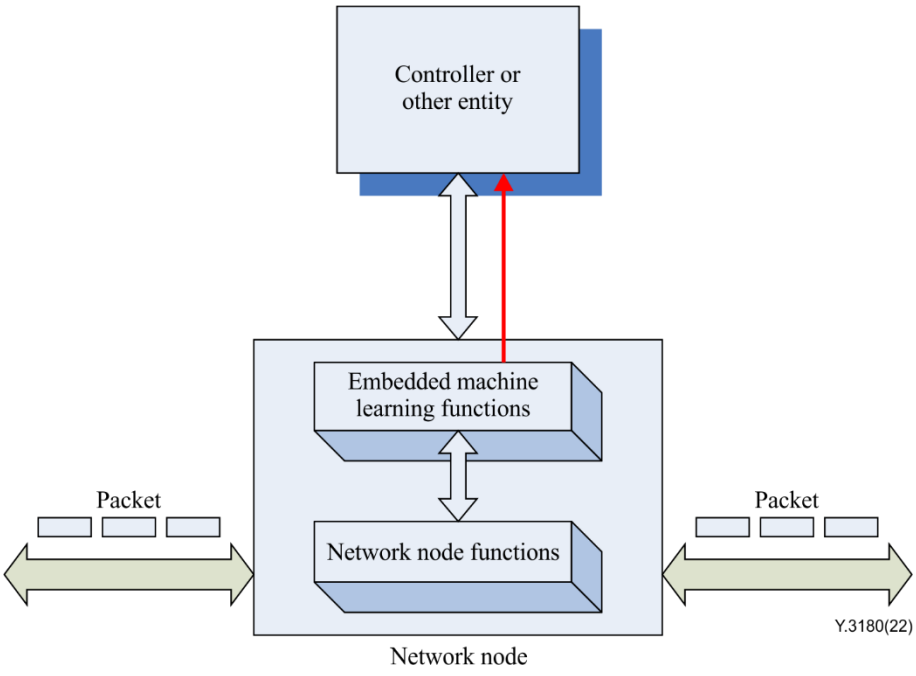

Figure 10-2 – Report process based on embedded machine learning system

10.1.3 Report based on hybrid machine learning system

In the hybrid machine learning system the malicious application-descriptor-agnostic traffic is discovered by either a machine learning system outside of the network node or a machine learning function in the network node. No matter which system is responsible for identifying the traffic, the machine learning function in the network node is responsible for reporting. If the malicious application-descriptor-agnostic traffic is discovered by the level 2 machine learning system, the level 2 machine learning system sends the information about the malicious application-descriptor-agnostic traffic to the level 1 machine learning system first, then the level 1 machine learning system reports this information to the controller or other entities. If the malicious application-descriptor-agnostic traffic is discovered by the level 1 machine learning system, the level 1 machine learning system reports this information to the controller or other entities directly.

The red arrows in Figure 10-3 indicate the report process.

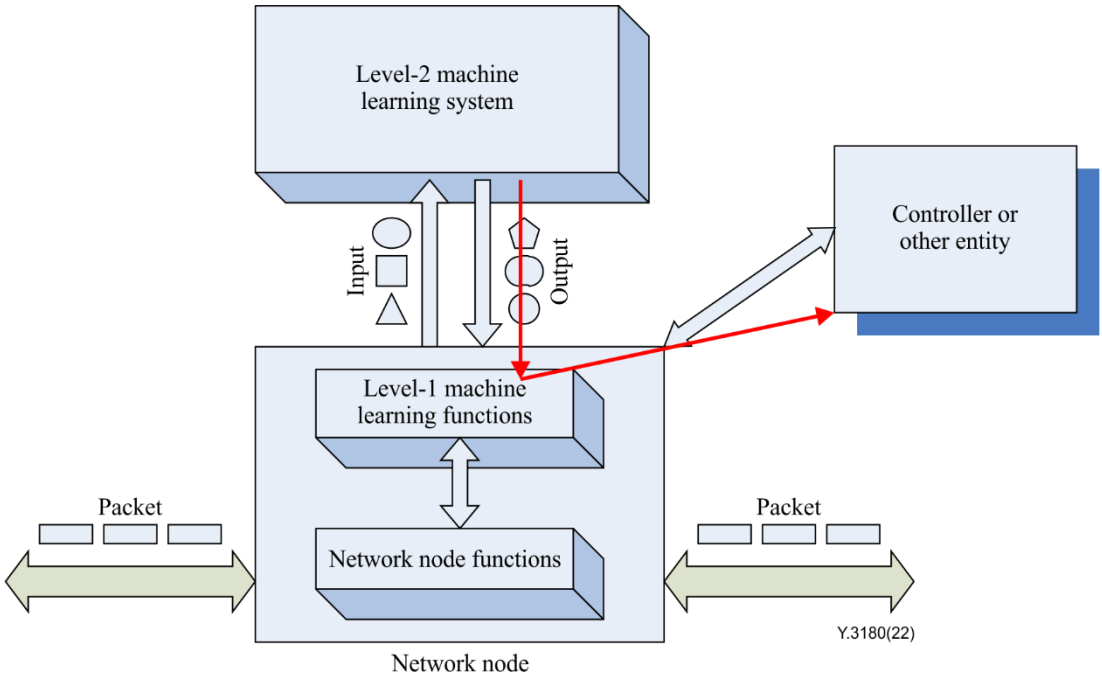

Figure 10-3 – Report process based on hybrid machine learning system

10.2 Auxiliary control mechanism for malicious application-descriptor-agnostic traffic

When some application-descriptor-agnostic traffic is identified as malicious traffic, it is recommended for the network to restrict the malicious traffic as soon as possible after reporting the malicious traffic to corresponding functional entities such as controller or network management system. So, the controller will set up some policies for the network node in order that the network node can block the malicious traffic from being further forwarded and doing further harm to the network.

11 Security considerations

The mechanisms specified in [ITU-T Y.2704] address the security requirements of this Recommendation. In addition, entities and the information pertaining to the mechanism specified in this Recommendation should be under protection against threats described in [ITU-T Y.2704].

Moreover, when the mechanisms specified in this Recommendation are implemented with help from the functional elements outside of the network, then the information exchanged between those functional elements and functional elements within the network should be under protection.

Bibliography

[b-ITU-T M.60] Recommendation ITU-T M.60 (1993), *Maintenance terminology and definitions*.

[b-ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020*.

[b-IETF RFC 5246] IETF RFC 5246 (2008), *The Transport Layer Security (TLS) Protocol*.

[b-IETF RFC 6797] IETF RFC 6797 (2012), *HTTP Strict Transport Security (HSTS)*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022