Article

Traffic Type Recognition Method for Unknown Protocol—Applying Fuzzy Inference

Sang-Won Kim and Kee-Cheon Kim *

Department of Computer Science and Engineering, Konkuk University, Seoul 05029, Korea;
llllssss94@konkuk.ac.kr

* Correspondence: kckim@konkuk.ac.kr

Abstract: In this paper, we propose a system that can recognize traffic types without prior knowledge of static features such as protocol header information by combining protocol analysis based on an ecological sequence alignment algorithm in a bioinformatics and fuzzy inference system. The algorithm proposed in this paper obtained up to a 91% level of performance at a similar level to several existing algorithms in experiments using datasets containing various types of traffic. In addition, it showed an excellent accuracy of 82.5% or more even under severe conditions that lowered the amount of data to a level of at least 40% or only included data in the middle of the traffic. This shows that the problem of dependence on initial data that frequently occurs in existing machine learning and deep learning-based traffic classification algorithms does not appear in the proposed algorithm. Furthermore, based on the ability to directly extract traffic characteristics without being dependent on static field values, it has secured the ability to respond with a small number of data by taking advantage of the flexibility of the membership function of the fuzzy inference engine. Through this, the applicability to low-power and low-performance environments such as IoT networks was confirmed. In this paper, we describe in detail the theoretical background for constructing such an algorithm and relevant experiments and considerations for actual verification.

Keywords: unknown protocol; fuzzy inference; network management; traffic classification

Citation: Kim, S.-W.; Kim, K.-C. Traffic Type Recognition Method for Unknown Protocol—Applying Fuzzy Inference. *Electronics* **2021**, *10*, 36. <https://doi.org/10.3390/electronics10010036>

Received: 2 December 2020
Accepted: 21 December 2020
Published: 28 December 2020

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the recent commercialization of 5G networks, various types of advanced technologies and services that have previously been difficult to achieve due to various limitations are actualizing. It is true that the commercialization of 5G services is still limited to the NSA (Non-Standalone) level, but the explosive development and growth of derivative services due to this is revealing of the limitations of the existing network management method from various viewpoints such as efficiency and security. In network management, the importance of classification technology according to traffic type has been increasing continuously. It is because the real-time recognition of the types of traffic occurring on the network enables the efficient use of limited resources and enhances flexibility to respond quickly to various changes. In the next-generation network environment, this high-level network management technique can be viewed as essential, not optional. In addition, since previous traffic type classification techniques often establish classification criteria from well-known types of information, the need for a more advanced method to cope with new types is increasing [1]. One of the major points to consider in designing and implementing advanced network management techniques is a change in the form of service provision. Numerous IoT (Internet-of-Things) devices and derivative services are growing very rapidly and are being integrated into existing network environments, and related devices generally do not have enough computing resources to run and use necessary services on their own. For this reason, most rely on the provision of services through network communication, which implies that the role of the network becomes more important.

In addition, it should be considered that the scope of application of IoT devices and services is expanding to areas where delayed operation or malfunction leads to irreparable damage, such as autonomous vehicles and power plants. Based on the recent expansion trend of MEC (Multi-Access Edge Computing), IoT gateway services take a form in which service requests transmitted from numerous devices do not reach the core network performing the actual function but are processed by an edge server located physically close to one another. Therefore, the decision to send requests to the core to either directly process it or ignore it is a very important process for efficient network management [2–4]. In addition, since edge servers and networks are geographically distributed, they are relatively weaker than core networks. This fact implies that to provide the core value of low-latency service, security for this cannot be ignored [2–4]. Research on the real-time traffic classification [5–9] and protocol reverse engineering analysis [10,11] are known as possible solutions to this. In the past, the classification of traffic based on network protocols and securing the visibility of unknown protocols have become major research fields. However, many existing traffic classification studies often used static information such as protocol headers as a criterion, and thus, a considerable amount of adjustment was needed when web services were integrated into one form. In addition, numerous services and traffic are generated from various types of new network nodes, including IoT devices, so it is reasonable to use an approach that recognizes the true characteristics of traffic rather than simple static information to respond to this. In addition, in the case of an encrypted or unknown protocol, the information-based approach is not applicable, so problem solving through the application of protocol reverse engineering analysis technology is appropriate [12].

In this paper, we devised a traffic classification approach for network management in a new form based on the protocol reverse engineering analysis method and the existing traffic classification studies by referring to related studies and technologies to be described later. Going beyond the knowledge-based approach to the existing static features, it seeks to secure the ability to cope with unknown or partially published protocols and to solve the problem that is largely dependent on the quantity and quality of data used for learning. To this end, we propose an approach to extract traffic characteristics through a combination of protocol reverse engineering analysis and existing traffic classification techniques to eliminate dependence on static characteristics. In addition, by introducing a fuzzy inference system as a new method to apply the extracted characteristics to traffic type recognition, we intend to secure the ability to respond to relatively little learning data. These research results are expected to be major in increasing the possibility of realizing functions in low-power and low-performance environments in the future and aim to improve overall network management by enhancing network security and securing efficiency. In this paper, we describe in detail the theoretical background for constructing such an algorithm and relevant experiments and considerations for actual verification. This paper is organized as follows. First, in Section 2, we look at the main technologies developed for network management and introduce traffic classification technologies through bioinformatics analysis and traffic type recognition. Section 3 explains the operation method of the improved algorithm and introduces the theoretical background. In Section 4, the effectiveness of the algorithm is verified through empirical tests on major verification items, and conclusions are drawn in Section 5.

2. Motivational Case Study

2.1. State-of-the-Art Methods

One of the cores of network management techniques being developed to cope with the various requirements of diverse devices and services is the proper identification of traffic occurring in the network. The right identification can be of great help in responding to the requirements of each service and effectively using limited resources. In particular, in environments such as MEC, requests for services with different requirements are often received from various types of devices, so traffic type information is usefully used as a basis for prioritizing network function processing and determining actions [1].

As a factor that makes it difficult to identify the traffic type, the presence of private or partially disclosed protocols cannot be ignored. For a long time, private or partially open protocols, including encrypted data, have caused difficulties in network management from various perspectives [10–16]. For this reason, existing studies that classify traffic types, our main purpose, have generally been conducted based on information obtained from the static structure of packets targeting some specific protocols. Machine learning-based traffic classification algorithms such as K-means [17,18], DBSCAN (Density-based spatial clustering of applications with noise) [18], Naïve Bayes [19–21], and DT (Decision Tree) Classifier [22,23] utilize characteristic information such as packet occurrence interval, number of packet bytes, and connection duration, and advanced studies have been developed recently through artificial neural network models such as CNN (Convolutional Neural Network) [24–26] applying deep learning technology. Table 1 shows information on major existing studies. In recent network conditions, machine learning-based K-means, DBSCAN, Bayes, and DT methodologies are greatly ineffective due to encrypted data. Accordingly, state-of-the-art studies using 1-D CNN [25] and RNN (Recurrent Neural Network) [24] models are attracting attention in order to cope with encrypted or unknown protocols. However, the approach using the artificial neural network model inevitably requires a high level of computing resources because the entire packet bytes are pre-processed as an image, which can be said to be an obstacle to actual network application. In addition, according to a recent research report, deep learning-based methodologies tend to rely on initial traffic data, and in this case, there is a problem that a very large amount of data must be kept in the device [3]. This is a very big drawback for low-power and low-performance environments such as IoT AP (Access Point). In this respect, the ability to recognize the purpose of traffic—that is, the type of service—can be of great value without being bound by the protocol. If it is classified so as to provide an appropriate service based on the characteristics of actual traffic, rather than relying on information obtained from the packet header for determination as in the existing method, it will have a low overhead and can greatly widen the range of responsiveness. In addition, it is possible to theoretically improve network management efficiency, as detailed classification according to use is possible even within the same protocol [27].

Table 1. State-Of-Art Traffic Classification Method Summary.

Data Type	Approach	Datasets	Target Protocols
Plain data	K-means [18]	Auckland IV [28]	Web, P2P, FTP, etc . . .
	DBSCAN [18]	Auckland IV [28]	Web, P2P, FTP, etc . . .
	Naïve Bayes [19]	Self-collected	FTP, SSH, SMTP, WWW, DNS, etc . . .
	Decision Tree [22]	Self-collected	FTP, Telnet, SMTP, DNS, HTTP
Encrypted data	1-D CNN [25]	ISCX dataset [29]	Visualized Image
	2-D CNN [25]	ISCX dataset [29]	Visualized Image

2.2. Protocol Analysis Algorithm for Unknown Protocol

2.2.1. Protocol Reverse Engineering

In the field of network security, attention has been paid to several vulnerabilities derived from protocols that are partially public or not at all, such as private protocols [10,11]. The corresponding protocol reverse engineering analysis is generally implemented in two types. First one is a network trace-based type that directly analyzes network packets, and the second is an execution trace-based type that collects and analyzes additional information like the program execution records at the upper network layer such as an application level. In recent studies that have achieved remarkable results, network trace-based analysis is mainstream. The output of protocol reverse engineering analysis consists of a PF (Protocol Format) containing syntax information and a PFSM (Protocol Finite State Machine) containing semantics [30,31]. Protocol reverse engineering analysis basically has the goal of securing visibility by grasping the syntax and semantics of unknown protocols from the perspective of security monitoring, and this has been resolved through

PF and PFSM, respectively. However, the value of these technologies and outputs is not simply limited to security monitoring. Being able to grasp the communication structure of unknown protocols has opened up considerable possibilities in various fields, and it is being used and applied in various network fields such as research to secure the availability between heterogeneous protocols and is increasing its value [32–35]. In this paper, it was used to understand the syntax and semantics of the unknown protocol target.

2.2.2. Bioinformatics Algorithm for Protocol Analysis

In this paper, we focused on the application research of the sequence alignment algorithm used in bioinformatics among protocol reverse engineering analysis studies. In bioinformatics, an algorithm using information-based and statistical approaches has been developed to solve the sequence alignment problem [36,37]. The Needleman–Wunsch algorithm [36], the most representative algorithm, compares the target sequences among biological sequences, covering the entire range from front to back. In the algorithm process, alignment is performed through a score matrix, and an example is shown in Figure 1. Through this process, the Needleman–Wunsch algorithm can divide the entire sequence data into a series of small subsequences.

Figure 1. The above example shows the execution result of the Needleman–Wunsch algorithm. The two sequences below the grid are the result of global alignment. The result of the Needleman–Wunsch algorithm provides an optimal alignment result by adding gaps to all target sequences used as inputs.

Such sequence alignment algorithms have generally been applied in various bioinformatics fields and developed centering on protein structure and DAN (deadenylating nuclease) sequence. The work of Beddoe [38] presented a method of applying the advantages of these bioinformatics sequence alignment algorithms to protocol reverse engineering analysis. Based on the results of this research, Bossert, G., Guihéry, F., and Hiet, G.’s Automated Protocol Reverse Engineering Analysis Study, Gascon, H., Wressnegger, C., Yamaguchi, F., Arp, D., and Rieck, K.’s black-box fuzz test, and other research achievements in the field of protocol reverse engineering analysis were affected. These studies tend to focus on obtaining security achievements such as securing monitoring visibility through the reverse engineering analysis of closed or partially disclosed protocols. Currently, related research achievements have reached a considerable level and are leading to practical application,

and derivative studies in the security field such as the detection of protocol anomalies and the network field such as securing compatibility between heterogeneous protocols are major.

As shown in Figure 2, a recent study by Tack-Hyun Jung and two others [39], which was conducted based on the results of previous research in the field of protocol reverse engineering analysis and traffic classification, shows the efforts to apply the traffic state estimation methodology through protocol reverse engineering analysis to the network management field. The researchers discover the syntax and semantics of unknown protocols through protocol reverse engineering analysis and propose a series of network management methodologies that obtain important state information for network management [39]. Finally, using the result of the protocol reverse engineering analysis, the syntax and semantics of unknown protocols were identified, and through this, a utilization plan for securing the availability between heterogeneous protocols was suggested. We noted that the protocol reverse engineering analysis technology was used to present a possibility for improving network management performance and efficiency. We found out there is a limitation in that a method that applies actual network management based on analyzed information cannot be specifically presented.

Figure 2. The figure above shows the architecture of the network state inference algorithm through the application of protocol analysis. Through protocol analysis, syntax and semantics of unknown protocol are found, and through this, important state information for network management is obtained [39].

2.3. Fuzzy Inference System

The fuzzy inference system can mathematically calculate reasonable conclusions for problems whose judgment criteria are ambiguous based on fuzzy theory as shown in Figure 3. Since the environment around us is full of ambiguous probabilities, those systems can be applied in the field of real-time problem solving such as reinforcement learning. In fact, in the field of game artificial intelligence, it is showing good results by making a reasonable conclusion as much as possible based on many variables of the surrounding environment at every moment [40]. Our team focused on refining the ability of the fuzzy inference system to make quick judgments in this ambiguous judgment environment. When determining the type of network traffic, there are many cases where there is no clear criterion for determining whether the amount of traffic is excessive or the average packet size of the traffic. The fuzzy inference system can show strength because it can express the degree to these criteria.

Figure 3. The figure above shows an example of a fuzzy set of a fuzzy inference system and its membership function. As can be seen in the figure, the fuzzy inference system has a method of defining all the degree values for each fuzzy set for one input and combining them to determine. (a) is triangular, and (b) is a trapezoidal model.

However, it is difficult to define detailed criteria for most algorithms, including fuzzy inference systems, deep learning, and machine learning, given little data. In a fuzzy inference system, a kind of probabilistic score is returned for an ambiguous characteristic through a membership function, but when there is little data, the membership function tends to be overly conservative. The research team developed a method to secure flexibility by partially improving the creation of these membership functions. In fact, through this, we have secured the ability to check numerically the degree of how far out of the learning data is even for inputs outside the learning data. Furthermore, we propose a new approach to solving the inefficient use of resources and dependence on the initial data of state-of-the-art algorithms based on deep learning. The proposed method secures the ability to cope with unknown protocols by utilizing existing studies on the analysis of unknown or closed protocols. Based on this, by applying the improved fuzzy inference system developed by our team, we developed an algorithm that shows high performance even with a small amount of data.

3. Traffic Type Recognition Algorithm

3.1. Contribution of Our Work

In this paper, we propose a traffic-type recognition method that applies the fuzzy inference system for the efficient realization of appropriate management for various types of traffic, as shown in Figure 4. Protocol analysis usually uses two methods to analyze unknown protocols. One is an execution trace-based method, which infers through the operation contents of an application related to traffic, and the other is a network trace-based method that collects and analyzes actual packets to infer protocol. Our algorithm extracts feature directly analyzed from traffic regardless of whether it is a well-known or unknown protocol through network trace-based analysis. Through estimation, similarities to traffic with extracted features can be recognized.

Figure 4. The figure above briefly shows the composition and relationship of each element constituting the traffic type recognition algorithm proposed in this paper. The overall algorithm is largely divided into two parts: train, target.

Existing deep learning and machine learning-based studies inevitably require a lot of learning data and have recently been found to depend on initial traffic [3]. Therefore, there arises a problem of maintaining the initial data of a lot of traffic. It is not suitable for the low-power and low-performance characteristics of the rapidly growing IoT environment. In addition, in a situation where the coexistence of traffic types with various characteristics within a single protocol increases, the criteria for making decisions for efficient network management are becoming very ambiguous. In this paper, by improving the membership function definition method of the fuzzy inference system, we intend to secure the ability to respond flexibly even when incomplete reference data are given [40].

3.2. Fuzzy Rule Generation

Protocol analysis using the sequence alignment algorithm [36] of bioinformatics identifies the structure of network packets and provides characteristic information. The series of processes is shown in Figure 5. Figure 5 shows a method of grasping the characteristics of traffic in whole traffic type recognition algorithms, creating a criterion for judgment and enhancing the membership function of the fuzzy set that constitutes the criterion. This method has the advantage of being able to quickly understand and respond to new traffic types when they occur.

Figure 5. The figure above shows a method of grasping the characteristics of traffic during the traffic type recognition process to create a standard, and to enhance the membership function of the fuzzy set. This method has the advantage of quickly grasping the characteristics of new types and coping with them, and it is possible to respond appropriately through the flexibility of the fuzzy inference system even in the situation of little learning data, which is a weakness of learning algorithms.

The protocol analysis module that performs this operation utilizes the Apriori algorithm and the bioinformatics algorithm. Through the Apriori algorithm, keywords representing common characteristics of target data are derived. Based on the derived keywords, a global sequence alignment process and a local sequence alignment process are performed to establish a standard that allows efficient distinction between protocol structures. Criteria candidates that can be obtained through protocol sequence alignment analysis were selected as shown in Table 2. As a process of extracting several features of target data within the Feature_Extraction process of Algorithm 1, the above sequence

alignment analysis is performed, and the result is used as an input value for generating a fuzzy value for generating a fuzzy IF–THEN rule in the next step.

Algorithm 1: Rule Generation.

Input: Packet Data **n**

Loop (packet):

Traffic flow $\leftarrow$ **Pre_Process**(**n**) # extract traffic flow from raw packets

Loop(Traffic flow):

$x_1, x_2, x_3, x_4, x_5 \leftarrow$ **Feature_Extraction**(Traffic flow) #—Static Keywords Number, Static Keywords Length, Payload Size Average, Fields Size Average, Packet Interval Average

$A_1, A_2, A_3, A_4, A_5 \leftarrow$ **Fuzzifier**(x_1, x_2, x_3, x_4, x_5) # generate membership function and calculate fuzzified values for each membership function

$I_n \leftarrow$ **Rule_Generator**($A_1, A_2, A_3, A_4, A_5, f_n$) # Fuzzy If Then Rule

End

Table 2. Features used for rule generation.

Feature(abbe)	Description
Static Keywords Number (SKN)	The number of static keywords appeared in traffic type
Static keywords Length (SKL)	Average lengths of static keyword for each traffic type
Payload Size Average (PSA)	Average size of payload included in a traffic type
Fields Size Average (FSA)	Average of each field length between a pair of keywords
Packet Interval Average (PIA)	Average time interval of each traffic type

Algorithm 1 shows a method of extracting protocol information using a bioinformatics sequence alignment algorithm and processing it into a fuzzy If–Then rule. There are several important key points in the creation of an If–Then rule. The most important part is understanding that static keywords and the characteristics of each dynamic field are given as scopes. In the If–Then rule, this information must be converted into a constant, so the problem of selecting a method for conversion occurs. We consider three selection methods in this paper. In the simplest range, the method of using the average value of the obtained range and the method of using the median value were considered. In addition, in order to properly reflect the results of the sequence alignment analysis, we chose a method of estimating the membership function directly through the mean and variance of the resulting range distribution.

Finally, there is a problem of selecting the type of membership function to be estimated, and the study was conducted in consideration of the (1) Gaussian model, (2) Triangular model, and (3) Trapezoidal model, which are represented by the formulas specified below. The Gaussian model is a method of measuring the degree of each fuzzy set in the form of a normal distribution and is expressed as a normal distribution equation as shown in Equation (1). The triangular model literally represents each fuzzy membership in a triangular shape, and the trapezoidal model represents a trapezoidal shape. These two types can be used when the distribution of data is uneven or contains discrete features, can be used to increase accuracy, and are expressed as equations representing straight lines for each section as in Equations (2) and (3). In this study, these models were appropriately selected and applied according to the characteristics of each feature.

$$\mu_{\text{gaussian}}(x) = \exp\left(-\left(\frac{x - m}{\sigma}\right)^2\right) \quad (1)$$

$$\mu_{\text{triangular}}(x) = \begin{cases} 0 & : x = \alpha \\ \left(\frac{\lambda}{\beta-\alpha}\right)(x-\alpha) & : \alpha \leq x < \beta \\ \lambda & : x = \beta \\ \left(\frac{\lambda}{\beta-\gamma}\right)(x-\gamma) & : \beta \leq x \leq \gamma \\ 0 & : x = \gamma \end{cases} \quad (2)$$

$$\mu_{\text{trapezoidal}}(x) = \begin{cases} 0 & : x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & : \alpha \leq x < \beta \\ 1 & : \beta \leq x \leq \gamma \\ \frac{x-\gamma}{\lambda-\gamma} & : \gamma < x \leq \lambda \\ 0 & : x > \lambda \end{cases} \quad (3)$$

$$\alpha < \beta < \gamma < \lambda$$

To estimate the fuzzy membership function, it is necessary to set an appropriate category for the input value and the result value. We basically define the result value as a value that indicates how well the input condition meets the criteria indicated by the If-Then rule. For the input value, three categories of high, match, and low were constructed by calculating the reference point according to the constant value estimation method. Based on the results of extracting the five characteristics specified in Table 2 from the data learned for the generation of membership functions, two methods to estimate high and low membership were defined: by calculating the first and third quartiles and simply using maximum and maximum values. In the end, a mixture of two methods is applied. When the maximum and minimum values are used as criteria for high and low, the membership functions of trapezoidal and triangular models can be used, which can help compensate for the imperfections of the training data.

In addition, we have devised a new definition method that increases the flexibility of the membership function used in the fuzzy inference system. Through this, the ability to respond to a small number of data can be secured through the application of the fuzzy reasoning system, and it can have strengths in vague judgments about the level of service demand that frequently occurs in network management [40]. With this improvement, the advantage obtained through the introduction of the fuzzy inference system can be maximized because even when the learned standard is deviated, the degree of deviation can be measured and determined flexibly. If the statistical data extracted from the training data are simply used as a criterion for judgment, it is difficult to produce adequate performance when the initial data are small, and a large amount of data are required. This problem can be avoided by padding to some extent the analysis result through estimation of the fuzzy membership function. In actual implementation, for the padding setting, a method of adding and subtracting the minimum and maximum values using multiples of the parent standard deviation was used. From time to time, the range can be further expanded, such as a factor of 2 or 3, which increases flexibility and risk. Based on this, it can be configured so that it can respond appropriately to the major weaknesses of the learning algorithms by taking advantage of the flexibility of the fuzzy inference system even in a situation where little learning data are given. Figure 6 shows an example of this method.

The final IF-THEN rule adopted the zero-order method of the Takagi-Sugeno (TS) fuzzy model in order to shorten the decision time and minimize the required computing power. A constant value is used as the result, and it is set as confidence in the judgment, which is the target result of this study.

$\mathcal{R}_l$: if *skn* is A_1 AND *skl* is A_2 AND *psa* is A_3 AND *fsa* is A_4 AND *pia* is A_5 then $f_l = P_l$

The If-Then rule created such as $\mathcal{R}_l$ can be used to recognize the traffic type by using the fuzzy inference value for each rule as a probability. In this case, various advanced defuzzing algorithms such as the weighted average defuzzification method can be applied to these rules and estimated values, and through this, the level of fuzzy membership

for each feature can be comprehensively determined, thereby providing a stronger basis for judgment.

Figure 6. The figure above shows an example of adding and subtracting multiples of the population standard deviation to the minimum and maximum values for the Padding setting. In some cases, the range can be further expanded, such as a factor of 2 or a factor of 3, thereby increasing flexibility and risk.

3.3. Traffic Type Recognition

Using the If–Then rule created through this process, the confidence value of determining whether the input data are the corresponding type is calculated for the traffic types learned by the model for the input data. The fuzzy membership level value defined in the If–Then rule is calculated by forming a fuzzy set for each feature for the membership functions of the traffic type to be determined. Therefore, we can define the t-norm and s-norm (t-conorm) operators by considering “and” and “or” operations of fuzzy sets for the features defined in the If–Then rule as intersection and union, respectively. In this paper, the most representative t-norm operator, the algebraic product operator, is used, and s-norm uses the maximum. Each definition is as follows (4) and (5).

$$\text{Algebraic Product : } T_{ap}(a, b) = ab \text{ [t - norm]} \quad (4)$$

$$\text{Maximum : } S_{max}(a, b) = \max(a, b) \text{ [s - norm]} \quad (5)$$

There are various ways of interpreting the If–Then rule in practice, and since this paper uses the product inference engine, appropriate Mamdani implications are used. In the product inference engine, the result of each rule is synthesized as a union combination. Through this, it is possible to secure simplicity in analysis that may require complex calculations. Accordingly, it is possible to perform an integrated calculation for the case where there are various combinations expressing the characteristics of a specific traffic type.

Algorithm 2 shows a series of processes that perform an appropriate response to the input traffic based on the characteristics of each traffic type analyzed from the viewpoint of actual network management. The process of obtaining the membership level by extracting the characteristics of the target traffic proceeds in the manner described in Algorithm 1. In the fuzzy inference engine, the product inference engine calculates a confidence value indicating the degree of conformity for each traffic type. The process of inferring the degree of conformity through the If–Then rule is shown in Figure 7, in which the whole case proceeds in a Modus Ponens manner. A schematic diagram of this process is shown in Figure 8, and Algorithm 2 takes place in the Traffic Recognition part of the figure.

Algorithm 2: Traffic Recognition.**Input:** Target traffic **T**, Fuzzy rule sets **R**, threshold **H****Output:** max(Traffic type) AND confidence**def Fuzzy_Inference_Engine**(traffic, degree, rules): **for each** rule in rules: Conclusion $\leftarrow$ **Defuzzifier**(rule, traffic) # list of defuzzied values for each rule Confidence $\leftarrow$ **Weighted_Average**(degree, Conclusion) **return** Confidence**F** [x_1, x_2, x_3, x_4, x_5] $\leftarrow$ **Feature_Extraction**(**T**) #—Static Keywords Number, Static Keywords Length, Payload Size Average, Fields Size Average, Packet Interval Average **D** [A_1, A_2, A_3, A_4, A_5] $\leftarrow$ **Fuzzifier**(x_1, x_2, x_3, x_4, x_5)**Loop**(**R**): Confidence $\leftarrow$ **Fuzzy Inference Engine**(**F**, **D**, rules) Decision $\leftarrow$ **max**(Confidence, Decision)**IF** Confidence > **H**: **Declare_Traffic_Type**(Confidence)**ELSE:** **Report** (Target Traffic) # Do default network management action and report**END**

Figure 7. A structure that judges the degree of conformance to a specific traffic type of target traffic through the If–Then rule. This is a series of processes that combine several judgment rules. The rules shown above are the case of including all features.

Figure 8. The figure above shows the traffic type recognition process. The actual traffic recognition process uses a methodology to determine the maximum confidence value for all traffic type rules of the Membership Farm based on fuzzy inference.

Finally, confidence is obtained through the result of fuzzy inference through the If Then rule, and based on this, the type of traffic is recognized and determined. At this time, the process of defuzzing is necessary to calculate the actual confidence. Since each fuzzy set is defined as many as the number of criterion for determining whether the traffic type is correct, several conclusions arise, and this is the process of analyzing the degree of conformance we want by synthesizing them.

$$f = \frac{\sum_r \mu^{A^r}(x) * f_r}{\sum_r \mu^{A^r}(x)}, \quad f \text{ is conclusion} \quad (6)$$

$r \in \text{rule farm}, A \in \text{fuzzy set}, \text{all inputs} \in x$

We use the weighted average method that averages the conclusions obtained through inference for all rules as weights and is calculated as shown in Equation (6). The weighted average defuzzifier is known as a very efficient algorithm in terms of computing resources and time and is used in many places. The overall components and processing flow are shown in Figure 9.

Figure 9. It is a schematic diagram of Algorithm 2. It shows processes that the membership level of the input traffic data is measured for the fuzzy set and membership function that exist for each criterion, and the membership level values of each feature are aggregated and determined through the weighted average defuzzification method. The results are expressed as confidence, which indicates similarity to the type of traffic in question.

4. Experimental Research

4.1. Datasets

This paper focuses on the theoretical background and experimental results to examine the performance of the proposed algorithm. In order to ensure the reliability of the experimental results, we adopt well-known public dataset called the ISCXVPN2016 provided by the Information Security Centre of Excellence (ISCX) and the Canadian Institute for Cybersecurity (CIC) based at University of New Brunswick in Fredericton [29]. The ISCX dataset contains six types of network packet data and provides general data and VPN encrypted data for each type. The detailed information of the dataset is shown in Table 3.

Table 3. ISCXVPN2016 dataset summary.

Category	Description	NonVPN	VPN
Chat	Hangout, facebook, skype chat, etc.	141,180	86,226
Email	SMTP-based mail packets and Gmail	87,307	21,581
File Transfer	Skype file transfer packets	1,427,699	378,089
P2P	Torrent packets	108,519	422,098
Streaming	Netflix, Spotify, Youtube, etc.	738,827	1,514,288
VoIP	Hangout voice connection packets	1,959,190	2,413,699

4.2. Experiment Design

In order to confirm the performance level of the proposed algorithm, we conduct comparative analysis with the latest studies. Specifically, experiments were conducted with the same data for the K-means clustering algorithm, 1D CNN, 2D CNN, and DAGMM (Deep Autoencoding Gaussian Mixture Model). For the experiment, three types of fuzzy sets for each input feature were constructed: low, match, and high. The fuzzy membership function uses a method of defining a min–max average Gaussian model through a new technique that applies padding of 3 sigma. An example of the membership function composed of these settings is shown in Figure 10.

Figure 10. Membership function with the padding method applied. This membership function is calculated for all input features. Through this, the membership degree for the input of the target traffic is then calculated.

Figure 10 denotes a membership function for PIA (Packet Interval Average) to which the padding method is applied. This membership function is calculated for all input features such as SKL (Static Keyword Length) and SKN (Static Keyword Number). Through this, the membership level for the input value of the subsequent target traffic is calculated.

4.3. Accuracy of Proposed Algorithm

Confidence measurement experiments for six traffic types (Chat, Email, File Transfer, P2P, Streaming, and VoIP) were conducted through the configured traffic type recognition algorithm. First, a classification experiment was conducted for Chat type and Email type, which have distinct characteristics. The confidence estimated through fuzzy rules and membership functions for Chat type data represents the level at which the input data can be determined to be Chat type. As a result of the experiment, the accuracy was 81% for the Chat type and 89% for the Email type.

However, as can be seen from Figure 11, for other types of packets, there are quite a few cases where the Chat type membership level was measured to be the highest, and the confidence level was quite high at the level in the late 1960s. Figure 11 shows the result of inferring confidence as a fuzzy rule for the Chat type. The Y axis represents the

measured confidence value, and the X axis represents the time order of traffic. The graph at the top shows the actual Chat type traffic results, and the graph at the bottom shows the misclassified results. A quantitatively calculated accuracy was found to be at the 81% level, and the range of measured values stably appeared from 70 to 69.4 in correct classification. It can be seen that the misclassified values are scattered over a large range of 70 to 67 and show a distinct difference.

Figure 11. Confidence value trend of packets whose membership level is the maximum for chat type among all data. (a) shows the distribution of the actual chat type data. (b) shows the distribution of the misclassified data.

Figure 12 shows the same experiment result for the Email type. It can be seen that the stability of the measured value is a bit more unstable in the Email type, and there are a few cases that the membership level is measured very low, ranging from 0 to 10. However, since the classification result was correct, the accuracy was measured higher at 88.6%. This is because even if the membership level is low, the maximum value among the measured values for all types comes from the rules for the Email type. Nevertheless, the reason why the measured value itself is low is because only one IF–THEN rule is set for each type. By adding the characteristics found in one type in the form of IF–THEN rules, it is possible to make the membership level measure high.

Figure 12. Confidence value trend of packets whose membership level is the highest for email type among all data. (a) shows the distribution of actual email type data. (b) shows the distribution of the misclassified data.

Based on these results, experiments were also conducted on the other four types (File Transfer, P2P, Streaming, VoIP). These types basically have the characteristic of continuously sending a lot of data, and the characteristics of a single packet are very similar due to the network communication limits such as the size of the MTU (Maximum Transfer Unit). In fact, in the experiment on the File Transfer type, the accuracy is relatively low at the 75% level. At this time, if you look at the distribution of the Confidence value in Figure 13, it was determined in a very narrow range. In fact, PP (Positive-Positive) is 99%, and it appears that it captures the characteristics of the traffic type very well. The results for all types are shown in Table 4.

Figure 13. Confidence value trend of packets whose membership level is the maximum for the file transfer type among all data. (a) shows the distribution of actual file transfer type data. (b) shows the distribution of the misclassified data.

Table 4. Accuracy for each traffic type.

Chat	Email	File	P2P	Stream	VoIP
81%	88.6%	75%	86.6%	67.3%	90.9%

Although there are variations depending on the type, the overall classification experiment was conducted by synthesizing the fuzzy sets and rules for each type, and as a result, it showed high accuracy levels of up to 91%.

4.4. Benchmark Comparison

The proposed algorithm was aimed at ensuring accuracy at the level of machine learning or deep learning-based approaches that require high computing resources while being able to respond to low-power and low-performance environments. Therefore, using ISCX data, performance benchmarks with typical machine learning and deep learning-based approaches were conducted as shown in Table 5 [1,41]. The experimental results showed a performance difference of about 6.5% from the experimental results using the 1D and 2D CNN models, but it should be taken into account that the study using the CNN model was based on static information obtained in advance. In addition, it appears that it has a very small difference in performance from the advanced DAGMM-based approach that combines two neural network models.

Table 5. Benchmark with state-of-the-art methods.

Approach	Method	Accuracy (Avg.)
Static Feature Include Protocol [18]	K-means [17]	28.2+
	(Liu, Y et al.) Network Traffic Classification Using K-means Clustering, IMSCCS 2007	28.2+
Deep Learning via Protocol	2D Convolutional Neural Network [25]	87.5+
	(Wang, Wei, et al.) End-to-end encrypted traffic classification with one-dimensional convolution neural networks, IEEE ISI 2017	87.5+
	1D Convolutional Neural Network [25]	90.0+
Entropy Estimation	(Wang, Wei, et al.) End-to-end encrypted traffic classification with one-dimensional convolution neural networks, IEEE ISI 2017	90.0+
	Deep Autoencoding Gaussian Mixture Model (DAGMM) – streaming intensity recognition [41]	89.7
Packet Sequence Alignment Analysis	(Kim, S. et al) Research on the traffic type recognition technique for advanced network control using Floodlight, IMCOM 2020	89.7
	Fuzzy inference-based protocol analysis traffic recognition algorithm [This Paper]	82.5+

In addition, K-means showed very low accuracy in the same experimental environment and considering the fast-learning speed and low computing resource consumption of the proposed algorithm, it can be seen that the protocol analysis and the approach using fuzzy inference show sufficient performance. In fact, it can be seen that the training time is very fast compared to the deep learning models that take more than 500 sec at the level of 0.0051 sec in the same environment, and it can be seen that the required computing resources are small. In addition, considering that the fuzzy sets and rules used in this

experiment are simply set as much as possible in consideration of the worst environment, there is sufficient possibility of performance improvement that takes advantage of the improved environment in actual application.

4.5. Inference Optimization

It has been described earlier that the algorithm of this paper has been improved to exhibit proper performance even when a small amount of data is given through fuzzy inference. To confirm this, an experiment was conducted to adjust the size of the input data. At this time, the data was reduced to 80%, 60%, and 40%, excluding the initial and last parts of the total traffic, and the change was confirmed. As shown in Figure 14, experimental results found that there was no noticeable decrease in performance, but that the confidence value slightly changed as the traffic characteristics were estimated slightly differently as only the data in the middle part of the traffic flow was used.

Figure 14. Accuracy trend according to the amount of learning traffic. Even when a small number of data including only the middle part of the total traffic flow is used, there is no change in positive-positive (PP) and Confidence, but there is a trend of increasing false-positive (FP) as shown by the arrow.

In addition, when looking at the specific trend of false-positive (FP) performance indicators, it is judged to be affected to some extent from the fact that an overall upward trend appears. Since the data used to compose IF-THEN rules have changed, the judgment criteria have changed, and the confidence value has changed slightly. Considering that the FP value gradually increases, and the PP value maintains a certain level for the same test data, it can be seen that the judgment criteria are expanded more flexibly as the quality and quantity of data to be provided decreases. Therefore, it can be seen that the target traffic type is still well classified, but the ability to distinguish other types of traffic is slightly inferior. However, even though it is a result of using a small amount of data, it shows satisfactory performance. In particular, considering that only the data in the middle part of the traffic flow is used, it is judged that the dependence on the initial flow data appearing in many modern algorithms is low.

The padding method developed to create this advantage is determined as a multiple of the standard deviation and affects the actual recognition accuracy. To verify this, an additional experiment was conducted to examine the estimation result of the membership function that can be obtained through the change of the padding value. As shown in Figure 15, it can be seen that as the σ value increase, the number of data judged as the confidence-zero level changes.

Figure 15. Confidence measurement trend according to padding setting. As the σ value increases, the estimation becomes more flexible. As shown in the actual graph, it can be seen that the number of completely excluded subjects whose confidence is measured to be 0 changes.

5. Discussion

This paper proposes an algorithm that realizes an appropriate level of traffic type recognition without a heuristic knowledge base by applying a protocol analysis technique using a sequence alignment algorithm in the bioinformatics field, going beyond the traditional protocol-based classification method of existing traffic classification studies. In this study, we tried to achieve the goal while maintaining the composition at the lowest possible level: the performance of the latest research level, the ability to cope with a small number of data, the removal of dependence on the initial data, and the removal of dependence on static information. Through the actual experiment, recognition accuracy of up to 91% was obtained, and it was confirmed that the performance was similar to that of the latest research conducted with the same data. These experimental results show that the algorithm proposed in this paper solves those problems while securing a certain level of performance.

Since the proposed method can cope with unknown protocols, it can contribute to strengthening the overall network security, and by using the classification result for actual network management, it is expected to improve the overall network management performance through securing efficiency. This type of approach follows the approach of protocol reverse engineering analysis technology in the field of network security and is very suitable for application to the MEC-based next-generation network environment that requires the management of numerous traffic types with various characteristics. In addition, as a result of reducing the amount of traffic data used to generate fuzzy rules and membership functions and changing the temporal location of the extracted data within the entire traffic flow, the performance dependence on the initial data is very low compared to other algorithms. These characteristics can be said to be suitable for low-power and low-performance environments such as IoT devices, which are gradually expanding as they require less resources of network equipment when considering actual network application. It presents a new direction to overcome the problem of dependence on existing initial data.

However, the fact that there are yet many setting options such as fuzzy set definition, fuzzy rule definition, judgment threshold setting, and padding value of membership function means that a lot of adjustments are needed for the algorithm to operate smartly by itself. In addition, it is clear that the composition of the data tested in this study is not easy, but it is necessary to consider how much more advanced it should be in terms of applying it to the actual environment. In addition, it is clear that it has confirmed that it has the ability to respond to the initial traffic data, but it is necessary to verify it in various domains later. Still, it is necessary to consider that the experimental setup in this paper was rather

simpler than the general level. In the future, optimization studies on important parameters such as experimental and judgment threshold values in an MEC-like environment based on NG-SDN (Next-Generation Software-Defined-Networking) are required.

Author Contributions: Conceptualization, S.-W.K.; methodology, S.-W.K.; software, S.-W.K.; validation, S.-W.K.; formal analysis, S.-W.K.; investigation, S.-W.K.; resources, K.-C.K.; data curation, S.-W.K.; writing—original draft preparation, S.-W.K.; writing—review and editing, K.-C.K.; visualization, S.-W.K.; supervision, K.-C.K.; project administration, K.-C.K.; funding acquisition, K.-C.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant NRF-2017M3C4A7083678, through the Next-Generation Information Computing Development Program, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. 2020R1A2C1101392).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [29].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nguyen, T.T.T.; Armitage, G. A Survey of Techniques for Internet Traffic Classification Using Machine Learning. *IEEE Commun. Surv. Tutor.* **2008**, *10*, 56–76. [\[CrossRef\]](#)
2. Hu, Y.-C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile Edge Computing—A key Technology Towards 5G. *ETSI White Paper* **2015**, *11*, 1–16.
3. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile Edge Computing: A Survey. *IEEE Internet Things J.* **2018**, *5*, 450–465. [\[CrossRef\]](#)
4. Pham, Q.-V.; Fang, F.; Ha, V.N.; Piran, M.J.; Le, M.; Le, L.B.; Hwang, W.-J.; Ding, Z. A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art. *IEEE Access* **2020**, *8*, 116974–117017. [\[CrossRef\]](#)
5. Dainotti, A.; Pescapé, A.; Claffy, K.C. Issues and Future Directions in Traffic Classification. *IEEE Netw.* **2012**, *26*, 35–40. [\[CrossRef\]](#)
6. Szabo, G.; Szabo, I.; Orincsay, D. Accurate Traffic Classification. In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Espoo, Finland, 18–21 June 2007; pp. 1–8.
7. Karagiannis, T.; Papagiannaki, K.; Faloutsos, M. BLINC: Multilevel Traffic Classification in the Dark. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Philadelphia, PA, USA, 22–26 August 2005; pp. 229–240.
8. Sun, G.; Chen, T.; Su, Y.; Li, C. Internet traffic classification based on incremental support vector machines. *Mobile Netw. Appl.* **2018**, *23*, 789–796. [\[CrossRef\]](#)
9. Singh, H. Performance analysis of unsupervised machine learning techniques for network traffic classification. In Proceedings of the 5th International Conference on Advanced Computing & Communication Technologies, Rohtak Haryana, India, 7–8 February 2015; pp. 401–404.
10. Sija, B.D.; Goo, Y.-H.; Shim, K.-S.; Hasanova, H.; Kim, M.-S. A survey of automatic protocol reverse engineering approaches methods and tools on the inputs and outputs view. *Secur. Commun. Netw.* **2018**, *2018*. [\[CrossRef\]](#)
11. Cui, W.; Kannan, J.; Wang, H.J. Discoverer: Automatic protocol reverse engineering from network traces. In Proceedings of the 16th USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2007; pp. 199–212.
12. Aceto, G.; Ciunzio, D.; Montieri, A.; Pescapé, A. Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation Lessons Learned and Challenges. *IEEE Transit. Netw. Serv. Manag.* **2019**, *16*, 445–458. [\[CrossRef\]](#)
13. Wang, P.; Chen, X.; Ye, F.; Sun, Z. A survey of techniques for mobile service encrypted traffic classification using deep learning. *IEEE Access* **2019**, *7*, 54024–54033. [\[CrossRef\]](#)
14. Höchst, J.; Baumgärtner, L.; Hollick, M.; Freisleben, B. Unsupervised traffic flow classification using a neural autoencoder. In Proceedings of the IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 9–12 October 2017; pp. 523–526.
15. Lotfollahi, M.; Siavoshani, M.J.; Zade, R.S.H.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2020**, *24*, 1999–2012. [\[CrossRef\]](#)
16. Rezaei, S.; Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [\[CrossRef\]](#)
17. Yingqiu, L.; Wei, L.; Yunchun, L. Network traffic classification using K-Means clustering. In Proceedings of the 2nd International Multi-Symposium of Computer and Computational Science (IMSCCS), Iowa City, IA, USA, 13–15 August 2007; pp. 360–365.
18. Erman, J.; Arlitt, M.; Mahanti, A. Traffic classification using clustering algorithms. In Proceedings of the SIGCOMM Workshop Mining Network Data (MineNet), Pisa, Italy, 15 September 2006; pp. 281–286.

19. Moore, A.W.; Zuev, D. Internet traffic classification using Bayesian analysis techniques. In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Banff, AB, Canada, 6–10 June 2005; Volume 33, pp. 50–60.
20. Zhang, J.; Chen, C.; Xiang, Y.; Zhou, W.; Xiang, Y. Internet traffic classification by aggregating correlated naive Bayes predictions. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 5–15. [\[CrossRef\]](#)
21. Auld, T.; Moore, A.W.; Gull, S.F. Bayesian neural networks for Internet traffic classification. *IEEE Trans. Neural Netw.* **2007**, *18*, 223–239. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Cohen, E.; Lund, C. Packet classification in large ISPs: Design and evaluation of decision tree classifiers. *ACM SIGMETRICS Perform. Eval. Rev.* **2005**, *33*, 73–84. [\[CrossRef\]](#)
23. Tong, D.; Qu, Y.R.; Prasanna, V.K. Accelerating Decision Tree Based Traffic Classification on FPGA and Multicore Platforms. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 3046–3059. [\[CrossRef\]](#)
24. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. *IEEE Access* **2017**, *5*, 18042–18050. [\[CrossRef\]](#)
25. Wang, W.; Zhu, M.; Wang, J.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 43–48.
26. Vu, L.; Bui, C.T.; Nguyen, U. A deep learning based method for handling imbalanced problem in network traffic classification. In Proceedings of the 8th International Symposium on Information and Communication Technology, Nha Trang, Vietnam, 7–8 December 2017; pp. 333–339.
27. Aitel, D. *The Advantages of Block-Based Protocol Analysis for Security Testing*; Immunity Inc.: Miami, FL, USA, 2002.
28. WAND Network Research Group. WITS: Auckland IV. 2001. Available online: <https://wand.net.nz/wits/auck/4/> (accessed on 27 August 2020).
29. Canadian Institute for Cybersecurity. VPN-nonVPN Traffic Dataset (ISCXVPN2016). 2017. Available online: <http://www.unb.ca/cic/research/datasets/vpn.html> (accessed on 27 August 2020).
30. Narayan, J.; Shukla, K.S.; Clancy, C.T. A survey of automatic protocol reverse engineering tools. *ACM Comput. Surv.* **2015**, *48*, 1–26. [\[CrossRef\]](#)
31. Duchene, J.; Guernic, C.L.; Alata, E.; Nicomette, V.; Kaaniche, M. State of the art of network protocol reverse engineering tools. *J. Comput. Virol. Hacking Techn.* **2018**, *14*, 53–68. [\[CrossRef\]](#)
32. Caballero, J.; Song, D. "Automatic protocol reverse-engineering: Message format extraction and field semantics inference". *Int. J. Comput. Telecommun. Netw.* **2013**, *57*, 451–474. [\[CrossRef\]](#)
33. Caballero, J.; Poosankam, P.; Kreibich, C.; Song, D. Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering. In Proceedings of the 16th ACM conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 621–634.
34. Bossert, G.; Guihery, F.; Hiet, G. Towards automated protocol reverse engineering using semantic information. In Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, Kyoto, Japan, 4–6 June 2014; pp. 51–62.
35. Gascon, H.; Wressnegger, C.; Yamaguchi, F.; Arp, D.; Rieck, K. Pulsar: Stateful black-box fuzzing of proprietary network protocols. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Singapore, 8–10 August 2017; pp. 330–347.
36. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [\[CrossRef\]](#)
37. Smith, T.F.; Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197. [\[CrossRef\]](#)
38. Beddoe, M.A. *Network Protocol Analysis Using Bioinformatics Algorithms*; ToorCon: San Diego, CA, USA, 2004.
39. Jung, T.; Song, W.; Kim, K. State Inference Method through Reverse Engineering for Network Protocols. *J. Korean Inst. Commun. Inf. Sci.* **2019**, *44*, 1535–1546.
40. Schwartz, H.M. *Multi-Agent Machine Learning: A Reinforcement Approach*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
41. Kim, S.; Lee, T.; Kim, K. Research on the traffic type recognition technique for advanced network control using Floodlight. In Proceedings of the 14th International Conference on Ubiquitous Information Management and Communication (IMCOM), Taichung, Taiwan, 3–5 January 2020; pp. 1–6.