Towards Interval-Valued Fuzzy Approach to Video Streaming Traffic Classification

Eduardo Maroñas Monks*, Bruno Moura Paz de Moura*, Guilherme Bayer Schneider*,
Hélida Salles Santos†‡, Adenauer Correa Yamin* and Renata Hax Sander Reiser*

*Federal University of Pelotas (UFPEL) - Pelotas, Brazil
Laboratory of Ubiquitous and Parallel Systems (LUPS/CDTEC)
{emmonks, bmpdmoura, gbschneider, adenauer, reiser}@inf.ufpel.edu.br

†Universidade Federal do Rio Grande (FURG) - Rio Grande, Brazil
Centro de Ciências Computacionais (C3)
‡Universidad Publica de Navarra (UPNA) - Pamplona, Spain
Institute of Smart Cities (ISC)
helida@furg.br

***Abstract*—This paper contributes to classifying video streaming traffic by exploring concepts of Interval-Valued Fuzzy Logic. This approach extends related works by considering the uncertainties generated by variations in network conditions and the parameter imprecision affecting the behavior of network flow, which increases the complexity to reach high accuracy in identifying the network traffic. Some evaluations using the interval-valued logic approach for video streaming traffic classification are presented using application tools and datasets to validate the proposal.**

***Index Terms*—Interval-Valued Fuzzy Logic, Classification, video streaming**

I. Introduction

Network traffic classification (NTC) is a fundamental process for several areas of study and research related to computer networks, such as information security, quality of service (QoS), accounting and service differentiation [1].

NTC is an essential technique for managing Information Technology (IT) infrastructure in companies, mainly Internet Service Providers (ISP), and also a relevant area of study and research [2]–[4].

In NTC, many factors may imply uncertainties and inaccuracies. We highlight the unpredictability of the occurrence of problems in the media, fluctuations in communication channels and computational resources used to maintain network convergence, overload on communication channels, configuration errors, and natural disasters. In the same way, these uncertainties and inaccuracies occur when obtaining the sensor metrics on the network resources, as they suffer disturbances and different variations.

In short, traffic classification consists of mapping network traffic into categories, such as normal or altered traffic, by type of applications (*streaming*, browsing, VoIP, Chat) or by identification of applications such as Youtube, Netflix, Telegram [5]–[7].

More specifically, some challenges that make traffic ranking important are: (i) to assist in diagnosing problems, locating network devices with issues, services with wrong configurations that consist in points of failure that cause packet losses and network errors; (ii) to contribute in security, detecting attacks, misuse of network resources, traffic anomalies; (iii) to identify applications and protocols based on Quality of Service (QoS), for managing network resources, differentiating traffic and analyzing the behavior of applications and protocols [7].

About 70% of network traffic on the Internet nowadays is composed by video *streaming* [8]. This type of traffic has specific characteristics, despite sharing the same protocols used for other types of services such as browsing or file downloading. Therefore, network management becomes complex and, consequently, the classification of streaming traffic within the large volume of packet flows.

In this sense, the challenge arises in providing techniques to classify network traffic, addressing the uncertainties and inaccuracies resulting from two sources: (i) the network environment; and (ii) the variables treated in the traffic classification process.

The use of NTC methods grows every day, encouraging challenges in the area to quickly and efficiently perform the classification. So, the following research questions stand out:

a) Do network traffic classifiers consider uncertainties to factors such as Packet Length Mean (PLM), Fwd Packet Length Std (PLS), and Backward InterArrival Time Total (BIAT)?
b) Are the network traffic classifiers equipped with consistent techniques for dealing with uncertainties associated with computer network environments?
c) Do the network managers have precise knowledge about what type of traffics is occurring in their networks?

This work was partially supported by CAPES, PQ/CNPq (309160/2019-7), PqG/FAPERGS (21/2551-0002057-1) and FAPERGS/CNPq PRONEX (16/2551-0000488-9).

d) Are the network traffic classifiers equipped with consistent techniques for dealing with encrypted traffic?

Considering these contexts and some works such as [3], [4], which use Fuzzy Logic to model uncertainties in the NTC process, this work aims to present an approach named FuzzyNetClass, exploring Fuzzy Logic for classifying network traffic, focusing on video streaming classification.

The paper is structured as follows. Related works are presented in Section II. Section III introduces basic concepts of type-2 fuzzy logic (T2FL). Section IV deals with the conceptual foundations of NTC. Section V presents CicFlowMeter tool used to extract attributes from network traffic flows. In Section VI, details of the *FuzzyNetClass* component and its conceptions are discussed, including database, fuzzification, rule base, inference, and defuzzification. Section VII describes the experimental evaluation. Finally, Section VIII presents the conclusions and future work.

II. Related Works

Related works using fuzzy logic on network classification are briefly reported in this section, and their characteristics are summarized in Table I.

Liang and Mendel proposed to use type-2 fuzzy logic classifiers (FLCs) to classify video traffic using compressed data [9]. In [10], an interval type-2 FLC was proposed to achieve a superior delivered video quality compared with existing traditional controllers and a T1 FLC fuzzy logic congestion controller (FLC). Rizzi et al. [11] used Min–Max neuro-fuzzy networks trained by PARC algorithm and compared it with a popular classification system based on machine learning. In [12], network traffic classification and anomaly detection methods based on traffic time series analysis using fuzzy clustering techniques were developed.

In [13], Shalaginov and Franke described the ongoing study and first results of the application of the Neuro-Fuzzy (NF) model to support large-scale forensics investigation in the domain of Network Forensics. In [14], three different data mining algorithms were discussed as part of the proposed solution for network fault classification: K-Means, Fuzzy C Means, and Expectation Maximization. In [15], Ducange et al. proposed to tackle the traffic classification problem by using multi-objective evolutionary fuzzy classifiers (MOEFCs).

In [2], Abdullah and Al-Hashmi proposed a novel evolving fuzzy system to discriminate anomalies by inspecting the network traffic. The results have proved the appropriateness of time series exploring fuzzy engine for network classification. In [3], it was proposed a new supervised hybrid machine-learning approach for ubiquitous traffic classification based on multicriteria fuzzy decision trees with attribute selection.

In [16], Iglesias et al. presented a network attack classification based on plain linear decision trees and fuzzy decision trees. Shifa et al. [4] proposed a Fuzzy-logic Threat Classification (FTC) model as the basis of a method to auto-detect three different confidentiality levels for streamed videos from heterogeneous mobile devices via web edge servers. In [17], Parfenov et al. work aimed at developing a fuzzy inference system to classify abnormal network traffic and identify current attacks by type using circa 550 fuzzy rules extracted from a decision tree.

Based on the papers above, summarized in Table I, our innovative strategy applies the interval-valued fuzzy logic approach to model uncertainty and imprecision on video streaming traffic classification.

III. Foundations of Fuzzy Logic

Lotf Zadeh introduced T2FL in 1975 as an extension of the traditional FL [18] modeling the inherent uncertainties related to the antecedent and consequent membership functions, enabling the manipulation of imprecise terms throughout its fuzzy inference system [19].

Type-2 fuzzy sets (T2FS) emerged when no procedure was available to select the crisp membership degree $\mu_A(x)$ of an element $x \in \chi$ in a fuzzy set A, meaning that it is not a single real value [20]. These sets are handy in situations where there exists uncertainty about the degrees, forms or parameters of the membership functions [21], providing potential strategy on the treatment of uncertainties in information models based on multiple-criteria obtained from distinct specialists and/or extracted from simulators [22].

In this proposal, Interval Type-2 Fuzzy Logic (IT2FL), based on T2FS theory, is suggested for the treatment of uncertainties, allowing to attribute an interval as the membership degree of an element x in a fuzzy set A [23]. Thus, extending the Fuzzy Set (FS) theory, IT2FS theory is able to model vagueness with an additional ability, considering imprecision (non-specificity) as another important aspect of uncertainty, reflecting this uncertainty by the length of the interval-valued membership degree.

Definition 1: [21] A T2FS A is characterized by a type-2 membership function $\mu_A(x, u)$ and given as follows:

$$A = \{(x, \mu_A(x, u)) : x \in \chi, u \in J_x \subseteq [0, 1]\}. \quad (1)$$

A T2FS assigns to an element in the universe χ a mapping $A(x)\colon [0, 1] \to [0, 1]$. A T2FS can also be given as

$$\{(x, A(x, t)) : x \in \chi, t \in [0, 1]\}$$

when $A(x, .)\colon [0, 1] \to [0, 1]$ is given as $A(x, t) = A(x)(t)$, for every $x \in \chi$, $t \in [0, 1]$. In particular, in Type-1 fuzzy sets (T1FS) $A(x)$ is a real number in $[0, 1]$, for every $x \in \chi$.

Definition 2: [23] When all $\mu_A(x) = 1$, then A is an interval type-2 fuzzy set (IT2FS), corresponding to

$$A(x) = \{(u, 1) : u \in J_x \subseteq [0, 1]\}, \forall x \in \chi.$$

Observe that Interval-valued Fuzzy Sets (IVFS) [24] are a particular case of T2FS. If A is an IT2FS, $A(x) = [\underline{A}(x), \overline{A}(x)]$, $\forall x \in \chi$. In addition, let A, B be IT2FS, the corresponding complement, union and intersection are also IT2FS given as:

$$A_C(x) = [1 - \overline{A}(x), 1 - \underline{A}(x)];$$
$$A(x) \cup B(x) = [\max(\underline{A}(x), \underline{B}(x)), \max(\overline{A}(x), \overline{B}(x))];$$
$$\mu_{A \cap B}(x) = [\min(\underline{A}(x), \underline{B}(x)), \min(\overline{A}(x), \overline{B}(x))], \forall x \in \chi.$$

TABLE I
RELATED WORKS.

Work	NTT	FLGA	ENT	FLE
[9]	MPEG (VBR) video	Fuzzy classifiers (type-1 and type-2) Singleton and Nonsingleton Interval		T1FL/T2FL
[10]	Video streaming	Interval type-2 FLC		T2FL
[11]	NT flows	Min–Max neuro-fuzzy networks trained by PARC algorithm	✓	Neuro-Fuzzy
[12]	Network traffic classification and anomaly detection	fuzzy C-means, Fukuyama and Sugeno index, Xie and Beni index, separation and compactness index		Clustering
[13]	Anomalous and malicious network traffic	Neuro-Fuzzy (NF), Self-Organizing Maps (SOM), Mean Absolute Error (MAE), Relative Absolute Error (RAE) and Mean Absolute Percent Error (MAPE)		Neuro-Fuzzy
[14]	Network fault classification	K-Means, Fuzzy C Means, and Expectation Maximization		Clustering
[15]	NT flows	Multi-objective evolutionary fuzzy classifiers (MOEFCs)	✓	MOEFCs
[2]	Anomalies by inspecting NT	Time Series Evolving Fuzzy Engine (TiSEFE)		T1FL
[3]	Anomalous and malicious NT	Hybrid approach combining decision tree learning and a fuzzy multicriteria classification method	✓	Hybrid
[16]	Classify abnormal NT and identify current attacks by type	Multiclass Fuzzy Classification, neuro-fuzzy classification	✓	Fuzzy Decision Trees
[4]	Real-time video streaming	Fuzzy Threat Classifcation (FTC) model	✓	T1FL
[17]	Identification of attacks in NT	Triangular membership functions for fuzzification and create the rule base from the C.45 decision tree	✓	Neuro fuzzy/T1FL
FNC	Video streaming	Interval-Valued Fuzzy	✓	T2FL

Network Traffic Types (NTT) Fuzzy Logic General Approach (FLGA) Encrypted Network Traffic (ENT) Fuzzy Logic Extensions (FLE) *FuzzyNetClass* (FNC) Network Traffic (NT) T1FL (Type-1 Fuzzy Logic) T2FL (Type-2 Fuzzy Logic)

In this paper, we denote $A(x) = X, B(x) = Y, \forall x \in \chi$, U as the set of all real intervals in the unit interval $[0, 1]$ and $\mathbb{U}$ as the set of interval type-2 fuzzy values. The partial order on $\mathbb{U}$ is the product order [25] given as:

$$X \leq Y \text{ iff } \underline{X} \leq \underline{Y} \text{ and } \overline{X} \leq \overline{Y}.$$

A system based on IT2FL can estimate input and output functions by using heuristic and interval techniques. Figure 1, graphically illustrates the inference system architecture based on IvFL. Its main blocks are briefly described as follows:

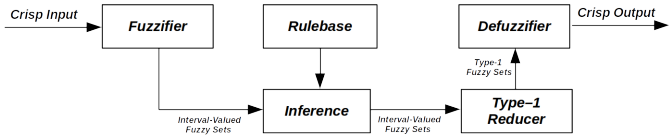


Fig. 1. Interval-Valued Fuzzy Inference System Architecture

1 **Fuzzification Interface (Fuzzifier)**: The fuzzification process based on IVFL is performed according to the nature and definition of a type-2 fuzzy set. It associates an input value with an interval function and not simply with a single value in U. In other words, the uncertainty regarding the input membership function is inserted into the inference mechanism. Thus, for each IVFS A, an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \chi^n$, for $n \in \mathbb{N}^*$ is related to a pair of vectors in $\mathbb{U}^n$ given as follows:

$$\left(\overline{A}(x_1), \overline{A}(x_2), \ldots, \overline{A}(x_n)\right), \left(\underline{A}(x_1), \underline{A}(x_2), \ldots, \underline{A}(x_n)\right).$$

2 **Rule Base (RB)**: Composed of rules classifying linguistic variables (LVs) according to the IVFS;

3 **Logic Decision Unity (Inference)**: It executes inference operations between the input data and the rules defined in the RB to obtain performance by the system action;

4 **Defuzzification**: Considering two main stages of IVFS:

(i) **Type-1 Reducer** transforms an IVFS into a fuzzy set, that is, it provides the best fuzzy set that represents the IVFS, satisfying the following premise: when all uncertainties disappear, the result of the Interval-valued Fuzzy Ruled Based System (IV-FRBS) is reduced to a Fuzzy Ruled Based System (FRBS) [26];

(ii) **Defuzzifier** provides an output given as the average of the extremes $\underline{Y}$ and $\overline{Y}$, expressed as:

$$y = \frac{\underline{Y} + \overline{Y}}{2} = \frac{\underline{A}(x) + \overline{A}(x)}{2}, \forall x \in \chi, \qquad (2)$$

and corresponding to the lower and upper bounds, related to the image by the membership function A applied to an element x in the universe χ. They are calculated using the iterative method of Karnik and Mendel (KM algorithm) [27].

The defuzzification step can still be obtained using a conventional method such as the centroid, as the final value of an inference system performance.

IV. Fundations of Network Traffic Classification

This section addresses the main topics related to the fundamentals for classifying network traffic. These topics are necessary for a better understanding of this work proposal.

A. Network Flows

The information from the packet control headers determines the relationship between them, enabling the construction of a packet flow (*flow*). The flow consists of five fields, formed by the source IP address, destination IP address, transport layer protocol, source port address, and destination port address [28].

The flow is considered the same when the forming fields remain equal. The flow can be analyzed in a bidirectional or unidirectional way, impacting the packet communication direction. From this premise, it is possible to collect and analyze other relevant information to classify the network traffic, such as the number of bytes generated, the length of time, and the time difference between each packet.

B. Network Traffic: Strategies for Classification

The literature review points out that the different strategies for traffic classification underwent an evolution from the complexity of protocols and services [5] [6]. The first method used was to identify information contained in the headers directly, for example, by relating the communication port to the type of application or protocol.

Another method used was the analysis of the behavior of *hosts*, analyzing the destinations of the generated packets and flows. With the advancement of network applications and protocols, mainly P2P, the use of packet payload analysis started to be used to relate patterns with preexisting protocol signatures [29].

With the widespread use of encryption in application traffic, analysis by packet content has become an inaccurate strategy to classify network traffic [30].

C. Video Streaming

Network video streaming is characterized by sending [31] chunks. Chunks are data segments sent according to network conditions and available resources on the client and server. Due to the possibility of changing the video quality during transmission and the use of protocols such as HTTPS, HTTP/2, and QUIC the application of traditional methods for identifying and classifying traffic becomes less effective [32].

Current video streaming players use two effective techniques: buffer and quality adaptation to optimize video streaming traffic. A buffer is used on the client-side to pre-store video data to compensate for network fluctuations and to survive short-term network outages.

Additionally, in order to meet the currently available long-term average Internet bandwidth, the playback quality is adapted to the available network bandwidth concerning the video encoding bitrate. When both techniques are used together with the Hypertext Transfer Protocol (HTTP) for information exchange, this is called HTTP Adaptive Streaming (HAS) [31]. These adaptations to optimize video streaming applications turn the network traffic classification more complex and with a greater amount of uncertainty.

Among the most popular video streaming applications, we have YouTube and Netflix, using MPEG DASH. YouTube, for example, has used HTML5 as the default playback option since early 2015, and as part of this, DASH in HTML5 is used wherever possible (e.g., IE11, Chrome, Safari). In MPEG DASH, the video content is provided in differently coded segments at the server-side, and the client requests the appropriate segments to stream the video. The client selects the segments to match the available end-to-end bandwidth between the client and content server.

The transmitted segments are played, and the video quality is adapted to this principle, depending on the bandwidth. The adaptation is made on the client-side by the adaptation logic. An adaptation logic can pursue many objectives where, generally, the main goal is to avoid a negative impact on video playback for the user.

V. CICFlowMeter

CICFlowMeter[1] is a network traffic flow generator and analyzer. Its choice for the conception of this proposal was due to its internationally wide adoption.

It can be used to generate bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions. Hence, more than 70 statistical network traffic features such as Flow duration, Number of packets, Number of bytes, Length of packets, among others, can be calculated separately in the forward and backward directions.

Additional functionalities include: (i) selecting features from the list of existing features, (ii) adding new ones, and (iii) controlling the duration of flow timeout. The application's output is the comma-separated values (CSV) format file with six columns labeled for each network flow (FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort, and Protocol) with more than 80 network traffic analysis features.

A. Attribute Selection

WEKA tool and the algorithm for selecting attributes CfsSubsetEval [33] were used to select the most relevant attributes for classifying network traffic, together with the BestFirst search method using the default parameters suggested by the tool. The results were analyzed and filtered with a network specialist aid.

Figure 2 shows the procedure applied to discover the most relevant attributes for classifying video streaming traffic. The datasets[2] used to perform the attribute discovery were generated from the authors' captures and publicly available datasets.

The selected attributes were "Packet Length Mean", "Fwd Packet Length Std", and "Bwd IAT Total". Packet Length Mean attribute displays the average value of the packet size

[1]https://github.com/ahlashkari/CICFlowMeter

[2]https://github.com/emmonks/datasets

in the network flow. Fwd Packet Length Std attribute displays the standard deviation of the average value of the packet size in the flow upload direction. Bwd IAT Total attribute shows the total time between packets (InterArrivalTime) in the flow download direction.

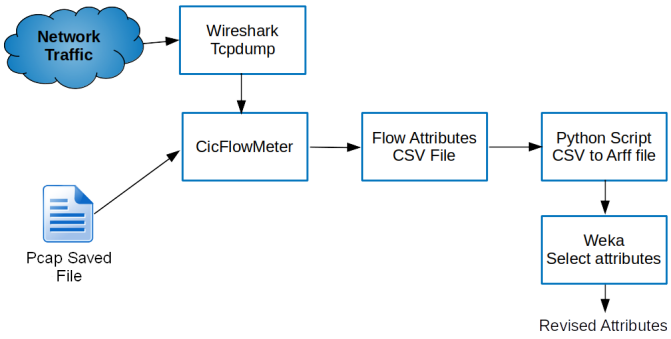


Fig. 2. Procedure for choosing attributes in CICFlowMeter

VI. FuzzyNetClass: Architecture Proposal

FuzzyNetClass[3] was conceived for classifying network traffic to identify video streaming flows. The *FuzzyNetClass* proposal considers a rule base acting in three stages: Fuzzification, Inference, and Defuzzification. It returns as output the classification level of the analyzed network flow, allowing (or not) its identification as a video.

The modeling of *FuzzyNetClass* system was performed using the Interval Type-2 Fuzzy Logic System Toolbox (IT2FLT) module [34], [35], and Juzzy [36]. See, in Figure 3, a block diagram modelling the *FuzzyNetClass* architecture.

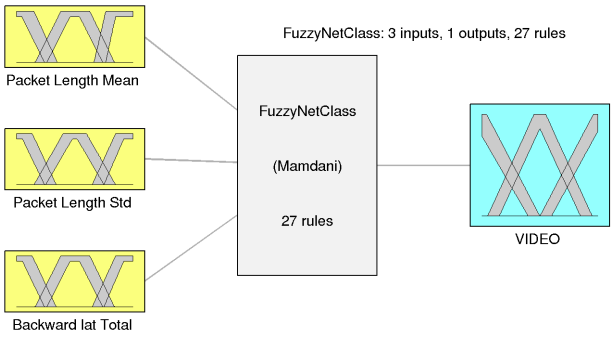


Fig. 3. Interval-valued Fuzzy System Block Diagram

A. FuzzyNetClass Data Base - Membership Functions

During the study of variables considering the experts' opinions, each LV was associated with four distinct FSs, using the trapezoidal graphical representation of the corresponding membership functions.

The selection of the attributes: Packet Length Mean, Fwd Packet Length Std, and Backward Iat Total to perform the NTC was based on the execution of the CfsSubsetEval algorithm using WEKA software[4], which is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.

Attribute values are applied to the default scale considering the range $[0, 10]$, establishing the value 10 as the limit for the values above it. So, for Packet Length Mean we use Eq. (3), for Fwd Packet Length Std we have Eq. (4), and Eq. (5) for Backward Iat Total, to obtain the membership degrees:

$$PLM = \frac{nf_i(PLM)}{\max PLM \cdot 10} \quad (3)$$

$$PLS = \frac{nf_i(PLS)}{\max PLS \cdot 10} \quad (4)$$

$$BIAT = \frac{nf_i(BIAT)}{\max BIAT \cdot 10} \quad (5)$$

according to the following parameters:

- nf_i represents a captured network flow;
- PLM is a network flow packet length mean attribute;
- PLS is a network flow forward packet length standard deviation attribute;
- $BIAT$ is a network flow backward packet total inter-arrival time attribute;
- $\max PLM$ is the total value of the highest packet length mean attribute identified in the dataset;
- $\max PLS$ is the total value of the highest forward packet length standard deviation attributed identified in the dataset;
- $\max BIAT$ is the total value of the highest backward packet total inter-arrival time attribute identified in the dataset.

The Linguistic Terms (LTs) defining the FSs of this Packet Length Mean (PLM) variable are stated as follows: "Low" (PLML), "Reasonable" (PLMR) and "High" (PLMH - best case). We denote $PLM = a$ and $a \in [0, 10]$. The Membership Functions are shown in Figure 4(a).

The Fwd Packet Length Std (PLS) attribute is used as input and obtained by reading the analyzed network stream. The LTs to the FSs defined for this variable are: "Low" (PLSL), "Reasonable" (PLSR - best case) and "High" (PLSH). We denote $PLS = b$ and $b \in [0, 10]$. These membership functions are presented in Figure 4(b).

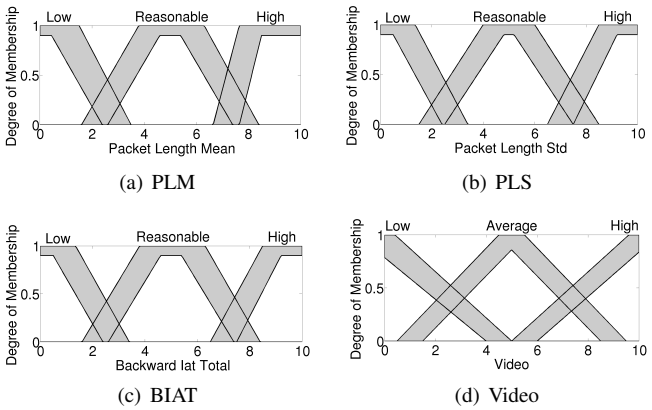
In the design of the FSs for Backward Iat Total (BIAT), the following LTs were created: "Low " (BIATL - best case), " Reasonable" (BIATR) and " High" (BIATH). We denote $BIAT = c$ and $c \in [0, 10]$. These membership functions are seen in Figure 4(c).

B. Fuzzification

At this stage, the input values are set by standard system scale as described in Section VI-A and then they are mapped to the fuzzy domain, as shown in Figure 5.

[3]https://github.com/brunomourapaz/FuzzyNetClass

[4]https://www.cs.waikato.ac.nz/ml/weka/



(a) PLM (b) PLS

(c) BIAT (d) Video

Fig. 4. PLM, PLS, BIAT, and Video in the default scale

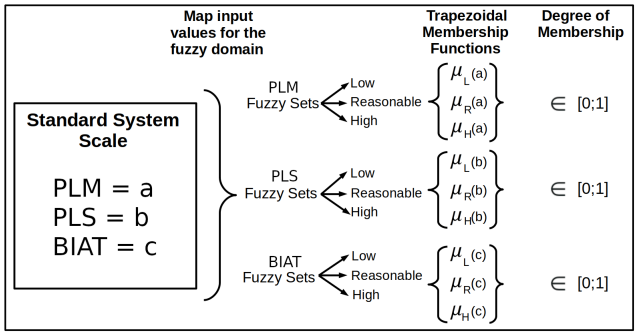


Fig. 5. Fuzzification Process

C. Rule-Based System

The Rule Base (RB) of *FuzzyNetClass* model is developed to be easily understandable, editable, and extensive. Table II summarizes the RB description. RB system can be extended, adding new rules if other input variables are desired to be manipulated. The RB considers the performance of the interval-valued system, describing consistently the control strategy [37] by considering three factors for its construction:

- the name of LV, turning the modeling closer to real world systems;
- the use of conjunctive (AND) operators aggregating the input variables;
- the implication function applied in the inference scheme using *Generalized Modus Ponens* (affirmative modus), i.e.,"if X is A, then Y is B".

D. Inference

In the inference process, the composition operators are used over FS, relating the rule antecedents to the implication functions that use the generalized modus ponens operator.

(i) The application of fuzzy operators on the input data, being three values obtained from the fuzzification. The "AND" fuzzy operator aggregates the main rules, and after the MIN (minimum) operator is considered.

(ii) Applying the fuzzy implication method: a combination of the value obtained by the fuzzy operators and the output FS values of the rule, using the MIN (minimum) operator on these combinations;

(iii) Fuzzy aggregation method results on the fuzzy output composition of each rule by using the MAX (maximum) operator. Thus, it creates a single fuzzy region to be analyzed by the next fuzzy process module.

TABLE II
FUZZYNETCLASS RULEBASE

	PLM		PLS		BIAT		VIDEO
	PLML		PLSL		BIATL		Average Video
	PLML		PLSL		BIATR		Low Video
	PLML		PLSL		BIATH		Low Video
	PLML		PLSR		BIATL		Average Video
	PLML		PLSR		BIATR		Low Video
	PLML		PLSR		BIATH		Low Video
	PLML		PLSH		BIATL		Low Video
	PLML		PLSH		BIATR		Low Video
	PLML		PLSH		BIATH		Low Video
	PLMR		PLSL		BIATL		Average Video
	PLMR		PLSL		BIATR		Low Video
	PLMR		PLSL		BIATH		Low Video
IF	PLMR	AND	PLSR	AND	BIATL	THEN	Average Video
	PLMR		PLSR		BIATR		Low Video
	PLMR		PLSR		BIATH		Low Video
	PLMR		PLSH		BIATL		Low Video
	PLMR		PLSH		BIATR		Low Video
	PLMR		PLSH		BIATH		Low Video
	PLMH		PLSL		BIATL		Average Video
	PLMH		PLSL		BIATR		Average Video
	PLMH		PLSL		BIATH		Low Video
	PLMH		PLSR		BIATR		Average Video
	PLMH		PLSR		BIATH		Low Video
	PLMH		PLSR		BIATL		High Video
	PLMH		PLSH		BIATR		Low Video
	PLMH		PLSH		BIATH		Low Video
	PLMH		PLSH		BIATL		Average Video

E. Defuzzification

In this module, the region transformation happens to result from the inference which is a discrete value (the utilization). The center of the area was the method used for modeling the defuzzification in *FuzzyNetClass*.

This method calculates the centroid (u) of the area consisting of the fuzzy inference system output (connecting all contribution rules stated in Sections VI-C and VI-D), as seen in Figure 6. The centroid is calculated by:

$$u = \frac{\sum_{i=1}^{N} u_i \mu_{OUT}(u_i)}{\sum_{i=1}^{N} \mu_{OUT}(u_i)} \tag{6}$$

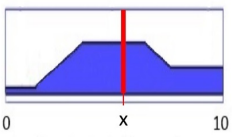


Fig. 6. Defuzzification Process using the Center of Area method

Figure 7 presents a flowchart of the proposed *FuzzyNetClass* architecture, where each step performed in the video streaming traffic classification is illustrated. The Wireshark/Tcpdump module is responsible for using the tools to capture network traffic and generate files in pcap format. Next, CicFlowMeter tool step aims to extract the network flows by processing the captured files and producing CSV files containing 77

attributes extracted from each network flow. In the Selected, Normalization Attributes step, a Python script is executed using Panda[5] library, which has the purpose of extracting and normalizing the selected attributes, and then making the PLS, PLM, and BIAT values ready to be used as input. In the fuzzifier module occurs fuzzification through the trapezoidal membership functions. Then, the inference process considers the rule base, and after the type-1 reducer transforms the interval-valued fuzzy set of the inference output into a fuzzy set. Finally, the Defuzzification is achieved, returning a single (crisp) value as the output.

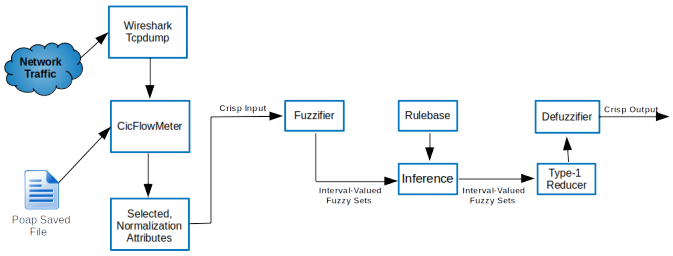


Fig. 7. FuzzyNetClass Architecture Proposal

VII. Evaluation

In order to validate the proposal, 339 videos were captured from the Youtube platform in VoD (Video on Demand) format. The capture time of each video was between 2 to 4 minutes to permit recording of possible quality changes due to intermittence and variations in network conditions. The tool Tcpdump[6] was used to save captured files. The Firefox browser was used to access the videos in its default configuration. The videos were captured over three months with capture sessions at different times of the day, on weekdays and weekends.

The captured files were submitted to the CicFlowMeter tool with the following parameters: 1200s of flow timeout and inactivity time at 5s. These parameters have the function of limiting the time of the network flow. In the case of this work, the capture time was limited to the maximum time of the original video. The QUIC protocol predominated in the transmission of the analyzed videos, but the HTTPS protocol was used in some cases. The QUIC and HTTPS protocols are encrypted by default. The streams were captured in a home Internet access environment with a 240 Mbit/s link with Gigabit Passive Optical Network (GPON) technology.

Additional files with captures were collected to generate other types of network traffic flows, captured from an academic network of a university in southern Brazil and from public datasets [38], [39]. Network flows of the following protocols were captured: DNS, NTP, FTP, SSH, HTTP, HTTPS, and QUIC. To validate HTTPS and QUIC streams, which are commonly used for video streaming, they were analyzed in the Wireshark[7] tool by a network expert to discard streams of video. We collected 3100 flows from dataset 20211017 and 11250 from dataset 20211024 serving as "noisy data" and evaluated the proposed classification process. In dataset 20211017, 247 video streaming flows were used, and 92 in dataset 2021124.

The captured files were submitted in the CicFlowMeter tool, and the output in CSV format was applied to a Python script with Panda library. This script output generates a CSV file with just the selected attributes and normalized attribute values.

In the sequence of the process, the generated datasets are processed by *FuzzyNetClass*, which performs all the steps of the interval-valued fuzzy inference system. The output provides the level of the analyzed stream, which will be used for the classification related to the type of video stream.

In Table III the *FuzzyNetClass* execution results are presented, where the percentages and quantity of classified flows in each group are highlighted concerning the membership degree for each output set.

In the range from 0.0 to 5.6, the flows that fall into the Low set group are classified. From 5.61 to 8.0, it groups the flows of the Average set. Finally, from 8.01 to 10, the High set flows are considered. In dataset 20211017, there was 78.13% accuracy, and in dataset 20211024, there was 77.17% for classifying video streaming flows in the Average or High ranges. The results obtained are promising and point to the continuity of the research.

TABLE III
Video Streaming Classification - Summary Results

Dataset	From 0.0 to 5.6		From: 5.61 to 8.0		From: 8.01 to 10.0	
	RFx1	PFx1	RFx2	PFx2	RFx3	PFx3
20211017	3044	94.00%	86	2.66%	107	3.3%
20211024	11296	99.38%	38	0.33%	33	0.29%

(RFx) Total flows (PFx) Percentage of flows in the range

VIII. Conclusions

In this work, we present *FuzzyNetClass*, a new approach for classifying video streaming network traffic using T2FL. Preliminary results showed a reasonable accuracy rate using datasets of network flows known in the literature and datasets of current captured flows. The main advantage of *FuzzyNetClass* is the modeling of uncertainties and inaccuracies obtained in the Internet network environment. Likewise, our approach provides flexibility and low computational cost to perform the processing in the classification of network traffic.

Further work considers an extension supporting admissible orders [40]–[42] to compare different sorting methods and analyze correlation and entropy of output data in different datasets, adding to *FuzzyNetClass* a dynamic step for generating rules. In addition, it enables adjustments allowing the classification of video streaming network traffic in live and VoD formats.

References

[1] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, 2015.

[5]https://pandas.pydata.org/
[6]https://www.tcpdump.org/
[7]https://www.wireshark.org/

[2] S. A. Abdullah and A. S. Al-Hashmi, "TiSEFE: Time series evolving fuzzy engine for network traffic classification," *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 116–124, 2018.

[3] F. Al-Obeidat and E.-S. El-Alfy, "Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols," *Personal and Ubiquitous Computing*, vol. 23, no. 5, pp. 777–791, 2019.

[4] A. Shifa, M. N. Asghar, A. Ahmed, and M. Fleury, "Fuzzy-logic threat classification for multi-level selective encryption over real-time video streams," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 5369–5397, 2020.

[5] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[6] T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning." *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 56–76, 2008.

[7] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2018.

[8] Sandvine, "The global internet phenomena report covid-19 spotlight," 2020.

[9] Q. Liang and J. M. Mendel, "MPEG VBR video traffic modeling and classification using fuzzy technique," *IEEE Transactions on Fuzzy systems*, vol. 9, no. 1, pp. 183–193, 2001.

[10] E. A. Jammeh, M. Fleury, C. Wagner, H. Hagras, and M. Ghanbari, "Interval type-2 fuzzy logic congestion control for video streaming across ip networks," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1123–1142, 2009.

[11] A. Rizzi, A. Iacovazzi, A. Baiocchi, and S. Colabrese, "A low complexity real-time internet traffic flows neuro-fuzzy classifier," *Computer Networks*, vol. 91, pp. 752–771, 2015.

[12] J. Asmuss and G. Lauks, "Network traffic classification for anomaly detection fuzzy clustering based approach," in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. IEEE, 2015, pp. 313–318.

[13] A. Shalaginov and K. Franke, "Automated generation of fuzzy rules from large-scale network traffic analysis in digital forensics investigations," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, 2015, pp. 31–36.

[14] K. Qader, M. Adda, and M. Al-Kasassbeh, "Comparative analysis of clustering techniques in network traffic faults classification," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 6551–6563, 2017.

[15] P. Ducange, G. Mannarà, F. Marcelloni, R. Pecori, and M. Vecchio, "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers," in *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[16] F. Iglesias, J. Milosevic, and T. Zseby, "Fuzzy classification boundaries against adversarial network attacks," *Fuzzy Sets and Systems*, vol. 368, pp. 20–35, 2019.

[17] D. Parfenov, L. Zabrodina, A. Zhigalov, and I. Bolodurina, "Research of multiclass fuzzy classification of traffic for attacks identification in the networks," in *Journal of Physics: Conference Series*, vol. 1679, no. 4. IOP Publishing, 2020, p. 042023.

[18] L. Zadeh", "The concept of a linguistic variable and its application to approximate reasoning—i," *Information Sciences*, vol. 8, no. 3, pp. 199 – 249, 1975.

[19] J. M. Mendel, "Fuzzy sets for words: a new beginning," in *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, vol. 1, 2003, pp. 37–42.

[20] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

[21] N. N. Karnik and J. M. Mendel, "Introduction to type-2 fuzzy logic systems," in *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence*, vol. 2, 1998, pp. 915–920 vol.2.

[22] K. Mittal, A. Jain, K. S. Vaisla, O. Castillo, and J. Kacprzyk, "A comprehensive review on type 2 fuzzy logic applications: Past, present and future," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103916, 2020.

[23] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[24] M. Gehrke, C. Walker, and E. Walker, "Some comments on interval valued fuzzy sets," *International Journal of Intelligent Systems*, vol. 11, no. 10, pp. 751–759, 1996.

[25] E. Klement, R. Mesiar, and E. Pap, "Triangular norms. position paper I: basic analytical and algebraic properties," *Fuzzy Sets and Systems*, vol. 143, no. 1, pp. 5–26, 2004.

[26] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *FUZZ-IEEE*. IEEE, 2011, pp. 2131–2138. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5976945

[27] J. M. Mendel, "On km algorithms for solving type-2 fuzzy set problems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 426–446, 2013.

[28] B. Claise, G. Sadasivan, V. Valluri, and M. Djernaes, "Rfc 3954: Cisco systems netflow services export version 9," *IETF http://www. ietf. org/rfc/rfc3954. txt*, 2004.

[29] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *2015 Tenth international conference on digital information management (ICDIM)*. IEEE, 2015, pp. 43–48.

[30] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, pp. 407–414.

[31] Y. Sani, A. Mauthe, and C. Edwards, "Adaptive bitrate selection: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2985–3014, 2017.

[32] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over http," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 562–585, 2018.

[33] M. A. Hall, "Correlation-based feature selection for machine learning," 1999.

[34] *"User's Guide, Interval Type-2 Fuzzy Logic Toolbox For Use with MATLAB"*, Tijuana Campus/Mexico, Institute of Technology and Baja California Autonomous University, 2005-2008.

[35] J. R. Castro, O. Castillo, and L. G. Martínez, "Interval type-2 fuzzy logic toolbox," *Engineering Letters*, vol. 15, no. 1, pp. 89–98, 2007.

[36] C. Wagner, "Juzzy - a java based toolkit for type-2 fuzzy logic," in *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, April 2013, pp. 45–52.

[37] G. J. Klir, *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley-Interscience, 2005.

[38] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[39] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *Proceedings of the Freenix Track: 2000 USENIX Annual Technical Conference, June 18-23, 2000, San Diego, CA, USA*. USENIX, 2000, pp. 263–270. [Online]. Available: http://www.usenix.org/publications/library/proceedings/usenix2000/freenix/cho.html

[40] H. Bustince, J. Fernández, A. Kolesárová, and R. Mesiar, "Generation of linear orders for intervals by means of aggregation functions," *Fuzzy Sets and Systems*, vol. 220, pp. 69–77, 2013.

[41] H. Zapata, H. Bustince, S. Montes, B. Bedregal, G. P. Dimuro, Z. Takáč, M. Baczyński, and J. Fernández, "Interval-valued implications and interval-valued strong equality index with admissible orders," *International Journal of Approximate Reasoning*, vol. 88, pp. 91–109, 2017.

[42] B. M. P. Moura, G. B. Schneider, A. C. Yamin, M. L. Pilla, and R. H. S. Reiser, "Allocating virtual machines exploring type-2 fuzzy logic and admissible orders," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.