FEDERAL UNIVERSITY OF RIO GRANDE - FURG
CENTER FOR COMPUTATIONAL SCIENCES
POSTGRADUATE PROGRAM IN COMPUTER SCIENCE
MASTER'S DEGREE IN COMPUTER ENGINEERING

Master's Dissertation

A Feature Engineering Approach for Anomaly Detection in Network Traffic Using the Generalized Choquet Integral

Abreu Esttebam Tavares Quevedo

Master's thesis presented to Postgraduate Program in Computer Science of the Federal University of Rio Grande - FURG, as a partial requirement for obtaining a Master's degree in Computer Engineering

Advisor: Prof. Dr. Bruno Lopes Dalmazo
Co-advisor: Prof. Dr. Giancarlo Lucca

Rio Grande, 2026

Q5u Quevedo, Abreu Esttebam Tavares

Uma Abordagem de Engenharia de Atributos para detecção de anomalias em tráfego de rede utilizando a integral de Choquet Generalizada/ Abreu Esttebam Tavares Quevedo - 2025.

55 f.

Dissertação (Mestrado) – Universidade Federal do Rio Grande – Programa de Pós-Graduação em Computação, 2025.

Orientador: Dr. Bruno Lopes Dalmazo
Coorientador: Dr. Giancarlo Lucca

1. Computação. 2 Detecção de anomalias 3. Integral de Choquet Generalizada. 4. Engenharia de Atributos 5. Detecção de DDoS. 6. Lógica Fuzzy. I. Dalmazo, Bruno Lopes. II. Lucca, Giancarlo. III. Título

CDU 004

Catalogação na fonte: Bibliotecária Brenda Sequeira CRB10/1979

Master's Dissertation

A Feature Engineering Approach for Anomaly Detection in Network Traffic Using the Generalized Choquet Integral

Abreu Esttebam Tavares Quevedo

Examining board:

Prof. Dr. Eduardo Borges
Federal University of Rio Grande (FURG)

Prof. Dr. Rodrigo Mansilha
Federal University of Pampa (Unipampa)

ACKNOWLEDGMENTS

I sincerely thank my entire family for their constant support, encouragement, and unconditional love. Their presence and strength have been fundamental throughout every step of this journey.

I am also deeply grateful to my advisors, Bruno Dalmazo and Giancarlo Lucca, for their guidance, dedication, and unwavering support throughout these two years. I also extend my sincere thanks to all the professors of PPGComp for their valuable insights, encouragement, and for helping shape my academic journey.

This study was financed in part by National Council for Scientific and Technological Development (CNPq), process 23/2551-0000773-8. Thank you.

ABSTRACT

QUEVEDO, Abreu Esttebam Tavares. **A Feature Engineering Approach for Anomaly Detection in Network Traffic Using the Generalized Choquet Integral**. 2026. 55 f. Dissertação (Mestrado) – Postgraduate Program in Computer Science. Federal University of Rio Grande - FURG, Rio Grande.

Network traffic constitutes one of the primary means of communication today and is essential for the proper functioning of various everyday activities. In the globalized context of the internet, numerous malicious actors seek to cause harm or extort victims, with Distributed Denial of Service (DDoS) attacks representing a critical threat to network stability. Although several models have been proposed, they remain far from achieving optimal performance in modern infrastructures. This study aims to evaluate the impact of a Feature Engineering approach on enhancing the performance of DDoS prediction algorithms Random Forest and XGBoost. **Specifically, the work proposes the optimization of predictive models by generating new features through an aggregation method based on the Generalized Choquet Integral with an adaptive α parameter**. By applying this method to the most relevant features identified by the SelectKBest algorithm, the study aims to effectively model complex dependencies among network variables that conventional methods typically ignore. Experimental results show that incorporating these new fuzzy-based features enhances predictive models, allowing Random Forest and XGBoost algorithms to achieve higher accuracy and stability even with a reduced feature set.

Keywords: Anomaly detection, Choquet Integral generalized, Feature Engineering, DDoS Detection, Fuzzy Logic.

RESUMO

QUEVEDO, Abreu Esttebam Tavares. **A Feature Engineering Approach for Anomaly Detection in Network Traffic Using the Generalized Choquet Integral**. 2026. 55 f. Dissertação (Mestrado) – Postgraduate Program in Computer Science. Federal University of Rio Grande - FURG, Rio Grande.

O tráfego de rede é um dos principais meios de comunicação da atualidade, sendo essencial para o funcionamento adequado de diversas atividades do nosso cotidiano. No mundo globalizado da internet, há inúmeras pessoas mal-intencionadas que visam prejudicar ou extorquir suas vítimas, tornando os ataques de DDoS uma ameaça constante. Diante desse problema, observa-se a existência de diversos modelos e algoritmos, embora ainda estejam longe de alcançar seu desempenho ideal. O objetivo deste trabalho é comparar e avaliar o impacto de uma abordagem de Engenharia de Atributos na melhoria do desempenho de algoritmos de predição de ataques DDoS Random Forest e XGBoost. A abordagem proposta consiste na otimização de modelos preditivos por meio da geração de novos atributos transformados, utilizando uma técnica de agregação baseada na Integral de Choquet Generalizada com a variação do parâmetro α adaptativo. Após a seleção das características mais relevantes via algoritmo SelectKBest, cada atributo original é individualmente processado pela Integral de Choquet para capturar interdependências e comportamentos não lineares que métodos convencionais podem ignorar. Os resultados demonstram que esta integração de características fuzzy permite que modelos como Random Forest e XGBoost alcancem maior estabilidade e precisão, resultando em uma redução significativa de alarmes falsos em tráfego legítimo.

Palavras-chave: Detecção de Anomalias, Integral de Choquet Generalizada, Engenharia de Atributos, Detecção de DDoS, Lógica Fuzzy.

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	artificial intelligence
ANOVA	analysis of variance
CIC	Canadian Institute for Cybersecurity
DDOS	distributed denial-of-service
MAE	mean absolute error
RF	Random Forest
ML	machine learning
PSO	particle swarm optimization
QoS	quality of service
SDN	software-defined networking
SMOTE	synthetic minority over-sampling technique
SVM	support vector machine
XGBoost	eXtreme gradient boosting

TABLE OF CONTENTS

1 INTRODUCTION

Computer networks are an integral part of daily life worldwide, enabling global communication and supporting essential services such as banking, healthcare, education, and business operations. In today's interconnected society, the increasing reliance on network infrastructure highlights not only its importance but also its vulnerability [18].

Data network traffic has played numerous roles in the field of security. However, issues such as data leakage, user information exposure, and authentication remain significant challenges in the current network environment. Furthermore, the exponential growth of data volume intensifies the complexity of managing and analyzing network traffic, requiring efficient techniques to process and extract relevant information [9]. Recent methods such as fuzzy time series graph mining [40] and programmable network strategies for anomaly mitigation [17] demonstrate the growing complexity and diversity of anomaly detection approaches in modern infrastructures.

Among the most critical threats to network stability are Distributed Denial of Service (DDoS) attacks. Given the increasing dependency on uninterrupted data services, malicious users may attempt to harm these services to disrupt access for individuals or organizations. In such attacks, multiple computers are hijacked and used to flood the target with excessive traffic, overwhelming the infrastructure and rendering it inaccessible.

Some examples of the magnitude of these attacks were highlighted in September 2025, when Cloudflare mitigated a record breaking distributed denial-of-service (DDoS) attack that peaked at 22.2 Tbps and lasted about 40 seconds. According to TecMundo, this attack was powerful enough to generate nearly ten billion packets per second, equivalent to streaming one million 4K videos simultaneously [19]. Such events illustrate the unprecedented scale of modern DDoS threats and reinforce the urgency of improving anomaly detection methods. In 2024, Cloudflare's autonomous DDoS defense systems blocked approximately 21.3 million DDoS attacks, marking a 53% increase compared to 2023, on average, the company mitigated 4870 DDoS attacks per hour throughout the year and in the same year the company reported an expressive 1885% increase in the number of attacks above 1 Tbps in the last quarter of 2024 [15], highlighting the severity and timeliness of this threat. A notable case in this context was the massive cyberattack against Elon

Musk's platform X, attributed to the hacker group Dark Storm, which caused a large-scale outage in March 2025 [31].

In the global scenario, Brazil has also become a major target of DDoS activity as in the first quarter of 2025, the country ranked as the 6th most attacked nation worldwide, with over 20.5 million DDoS attempts, representing a 358% of an over year increase. The main targets included telecommunications infrastructure, service providers, internet platforms, and the financial sector. Furthermore, Brazil also ranked among the top 10 countries from which DDoS attacks originated, underscoring its dual role as both a target and a source of malicious traffic in the network, another report from Cloudflare showed gambling and casino industry was the most attacked globally, followed by telecommunications, IT services, internet platforms, gaming, banking and financial services.

These malicious activities often generate identifiable patterns or irregularities, known as anomalies. Detecting such anomalies is essential for mitigating attacks effectively. However, due to the high dimensionality and complexity of modern network data, it becomes necessary to reduce the dimensionality of the data extracted from the network and focus on the most relevant features [24].

Concerned with current network security challenges, this study aims to detect anomalies by implementing a feature engineering approach. The proposed method leverages techniques such as Feature Selection [33] and the Generalized Choquet Integral [27] to enhance the detection and mitigation of DDoS attacks.

In light of these challenges, the Choquet integral serves as a powerful tool for data aggregation, as the fuzzy measure allows for effectively modeling relationships within the given data [3, 10]. Although numerous methodologies in the literature have focused on improving the performance of traditional machine learning algorithms such as Random Forest and XGBoost, they typically rely on conventional feature engineering techniques. Meanwhile, some recent works, such as [22] and [23], have applied the Choquet integral in domains like sustainable transportation and medical monitoring, showing its potential for modeling complex dependencies. However, this methodology remains unexplored in the field of cybersecurity, highlighting that no prior work has applied the generalized Choquet integral as a feature engineering mechanism for enhancing anomaly detection in network traffic, particularly in the context of network traffic analysis.

Therefore this work introduces a novel feature engineering approach for network traffic analysis, leveraging the fuzzy Choquet integral framework. A key element of the method applied in this study is the parameter α, which has a substantial impact on both execution time and the accuracy of anomaly detection. The α parameter serves as a tuning factor in fuzzy aggregation, in other words, different values regulate the degree of interaction among the variables. Depending on its value, the integral may become more conservative, requiring consensus across multiple attributes or more permissive, allowing a single attribute to exert greater influence. In this way, α operates as a calibration me-

chanism that directly shapes the model's sensitivity. The effectiveness of this approach is validated using real-world data from a reliable source, demonstrating its practical applicability in network management.

1.1 The research question

How can the generalized Choquet integral improve network anomaly detection?

Currently, distributed denial-of-service (DDoS) attacks are among the most critical threats to network stability. Such attacks can severely disrupt essential services, including healthcare, financial systems, government platforms, and public safety infrastructures, leading to significant social and economic consequences. For this reason, researchers and practitioners worldwide have been continuously developing methods to strengthen detection and mitigation strategies against these large-scale threats.

Despite the availability of various anomaly detection models based on machine learning, many still struggle to identify rare patterns, such as those generated by actual attacks hidden within legitimate traffic. In this context, the generalized Choquet integral, especially when fine-tuned through the α parameter, can enhance DDoS detection by serving as an effective feature engineering technique for machine learning models.

To address this research question, it is necessary to investigate whether the generalized Choquet integral can effectively generate more expressive features that improve the predictive performance of machine learning models in detecting anomalies. By doing so, this dissertation aims not only to contribute to the academic discussion on the generalized Choquet Integral as a tool for anomaly detection but also to provide practical insights for strengthening the resilience of network infrastructures against large-scale attacks. The next section presents the general and specific objectives that guide this research.

1.2 Objective

This section presents the objectives to be pursued throughout the development of this dissertation. Specifically, the general objective will be introduced first, followed by the specific objectives.

1.2.1 General objective

The objective of this dissertation is to explore methods that employ the Choquet integral with adaptive α for data aggregation. After selecting the most relevant features using the SelectKBest algorithm, these aggregations will be used to generate a new set of features aiming at improving the performance of traditional DDoS attack detection algorithms, specifically Random Forest and XGBoost.

1.2.2 Specific objectives

- To develop and optimize a computational framework for Generalized Choquet Integral Copulas, employing search algorithms to determine the optimal α parameter for network traffic aggregation

- Identify the most relevant features for anomaly detection in network traffic by running algorithms, with the goal of applying the generalized Choquet integral with an adaptive α parameter to the top-ranked features selected by the algorithm.

- Integrate the new set of features into the test, evaluate its impact on predictive models, analyze and interpret the resulting outcomes.

1.3 Contributions

The outcome of the design, experiments, and assessments of several mechanisms throughout this work resulted in the following publications, which are cited throughout the text with their respective contributions:

1.3.1 Conference papers

Published

- QUEVEDO, A. E. T.; AYRES, DENNER ; DIMURO, GRAÇALIZ ; RIKER, A. ; LUCCA, GIANCARLO ; DALMAZO, B. L. . **Optimizing Big Data Traffic Prediction Using Generalizations of Choquet Integral with Adaptive Weighting**. IEEE International Conference on Communications (ICC), 2025.**(h5-index: 76 (2025)) (Article [35])**

- QUEVEDO, A. E. T.; AYRES, DENNER ; Teixeira, Gabriel ; LUCCA, GIANCARLO ; DIMURO, GRAÇALIZ; DALMAZO, B. L. . **Improving Anomaly Detection in Network Traffic Using Choquet-Based Feature Engineering for Random Forest and XGBoost Models**. In: International Conference on Computational Science and Its Applications, 2025, Istanbul. Lecture Notes in Computer Science, 2025. v. 15650. p. 3-16. **(h5-index: 31 (2025)) (Article [36])**

- QUEVEDO, ABREU; AYRES, DENNER ; DIMURO, GRAÇALIZ ; LUCCA, GIANCARLO ; RIKER, ANDRÉ ; DALMAZO, BRUNO L. . **O Parâmetro α na generalização da Integral de Choquet para Previsão de Tráfego de Rede**. In: Escola Regional de Redes de Computadores, 2024, Brasil. Anais da XXI Escola Regional de Redes de Computadores (ERRC 2024). p. 30. **(h5-index: 5 (2025)) (Article [34])**

1.3.2 Cooperation papers

Published

- AYRES, DENNER ; QUEVEDO, ABREU ; LUCCA, GIANCARLO ; DIMURO, GRAÇALIZ ; DALMAZO, BRUNO L. . **Comparando Médias Móveis com Integral de Choquet para Detectar Anomalias no Tráfego de Redes**. In: Extended Proceedings of the Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 2024 p. 353. **(h5-index: 6 (2025)) (Article [5])**

- AYRES, DENNER ; QUEVEDO, ABREU ; DIMURO, GRAÇALIZ ; LUCCA, GIANCARLO ; DALMAZO, BRUNO L. . **Detecção de Anomalias de Rede utilizando Integrais de Choquet através de Medidas de Potência**. In: Escola Regional de Redes de Computadores, 2024, Brasil. Proceedings of the XXI Escola Regional de Redes de Computadores (ERRC 2024). p. 129. **(h5-index: 5 (2025)) (Article [4])**

1.3.3 Submitted

- QUEVEDO, ABREU ; GABRIEL TEIXEIRA; DIMURO, GRAÇALIZ ; LUCCA, GIANCARLO ; DALMAZO, BRUNO L. . **Reducing Dimensional Complexity in Big Data Network Traffic via Adaptive Choquet Aggregation**. *IEEE International Conference on Communications (ICC), 2026.*

1.4 Outline of the work

This dissertation is organized into the following chapters:

- **Chapter 2 – Theoretical foundation and concepts**: Presents the theoretical background on fuzzy logic, the Choquet integral, copulas, and supervised learning models that underpin the techniques used in this research.

- **Chapter 3 – Related Work**: Reviews existing studies on DDoS detection, fuzzy models, and feature selection methods, identifying key gaps and motivating the proposed approach.

- **Chapter 4 – Choquet Integral and adaptive α parameter**: Describes the implementation of the Generalized Choquet Integral, the construction of fuzzy copulas, and the adaptive optimization of the α parameter through a binary search strategy to minimize error in aggregation.

- **Chapter 5 – Fuzzy feature engineering for anomaly detection**: Details the proposed methodology, including the use of SelectKBest for feature selection, the generation of new features using the Choquet Integral and the application of Random Forest and XGBoost for performance evaluation.

- **Chapter 6 – Final considerations**: Summarizes the contributions and main findings of the work, highlighting the improvements achieved in anomaly detection through the proposed fuzzy feature engineering approach, and suggests directions for future research.

2 THEORETICAL FOUNDATION AND CONCEPTS

This chapter presents the fundamental concepts necessary for understanding the remainder of this work. Grasping these concepts will provide a solid foundation for the analysis and discussion of the results, enabling a clearer comprehension of the context and the contributions of this dissertation.

2.1 Search Method

When there is a need to locate a specific value within an array, we are faced with a set of elements and a series of challenges. Choosing the right search strategy becomes essential, especially when efficiency and scalability are at stake. In the context of this dissertation, we focus solely on the binary search method, given its relevance and adequacy to the dataset and problem addressed.

Binary search is a classic algorithm designed for efficiently locating a target value within a sorted array. As detailed in [25], the algorithm works by repeatedly dividing the search interval in half. It begins by comparing the target value to the middle element of the array. If a match is found, the corresponding index is returned immediately. Otherwise, the search continues in the half of the array that may contain the target, discarding the other half entirely.

To illustrate this procedure, consider a sorted array A with n elements and a target value T. The binary search algorithm attempts to find the index of T within A, assuming the array is indexed from 0 to $n - 1$.

The steps are as follows:

- Initialize L as 0 (the index of the first element) and R as $n - 1$ (the index of the last element).
- While $L \leq R$:
  - Compute m as the integer average of L and R: $m = \lfloor (L + R)/2 \rfloor$.
  - If $A[m] == T$, return m; the target has been found.
  - If $A[m] < T$, update L to $m + 1$ to continue searching the upper half.

  - If $A[m] > T$, update R to $m - 1$ to continue searching the lower half.

- If the loop ends without finding the target, return an indicator that the value was not found.

This method drastically reduces the number of comparisons required, making it highly efficient for large datasets, provided that the data is already sorted. Its simplicity and performance make binary search a valuable tool in algorithmic problem-solving and data retrieval contexts.

2.2 Random Forest and XGBoost

In this work, two supervised learning algorithms are employed to evaluate the proposed feature engineering method, **Random Forest** and **XGBoost**. Both models are based on decision trees but are different fundamentally in their ensemble learning strategies, which directly affects their performance characteristics, computational requirements, and suitability for different types of data.

2.2.1 Random Forest

Random Forest (RF) was introduced by Breiman in [8] as an ensemble learning method that combines multiple decision trees through bootstrap aggregating (bagging) and random feature selection at each node split. In this approach, each tree is trained on a bootstrapped sample of the original dataset, and at every split, a random subset of features is considered. The final prediction is obtained by aggregating the predictions of all trees via majority voting in classification or averaging in regression.

Through this randomization, it's possible to reduce the correlation between trees, improving generalization and reducing the risk of overfitting, being robust to noise, capable of handling high-dimensional datasets.

2.2.2 XGBoost

Extreme Gradient Boosting (XGBoost) is a scalable and optimized implementation of the gradient boosting framework proposed by Chen and Guestrin in [13]. Unlike RF, which builds trees in parallel, XGBoost constructs trees sequentially, with each new tree trained to correct the errors of the ensemble built so far. XGBoost incorporates several innovations to improve efficiency and accuracy, some of them are listed below:

- **Regularization**: L1 (Lasso) and L2 (Ridge) penalties are applied to tree weights to prevent overfitting.

- **Sparsity awareness**: the algorithm handles missing values and sparse data efficiently.

Table 1: Example table of t-norms

Name	Definition
Minimum	$T_M(x, y) = \min\{x, y\}$
Algebraic Product	$T_P(x, y) = xy$
Lukasiewicz	$T_L(x, y) = \max\{0, x + y - 1\}$

- **Parallelization**: optimized use of hardware resources during tree construction.
- **Shrinkage**: learning rate control to slow down the boosting process and improve generalization.

Comparison

While both RF and XGBoost can handle high dimensional data and capture nonlinear feature interactions, their strengths are different. While RF is generally more robust in noisy datasets with limited hyperparameter tuning, XGBoost can provide superior predictive performance when tuned but is more sensitive to parameter choices. In the context of anomaly detection in network traffic, RF offers stability and interpretability, while XGBoost can exploit subtle patterns.

2.3 Aggregation functions and the Choquet Integral

An important class of fuzzy operators is the class of aggregation operators [7]; [28].

Definition 1 A function $A : [0, 1]^n \to$ [0, 1] is called an n-ary aggregation operator if the following conditions are satisfied:

(A1) A is increasing in each argument: for each $i \in \{1, \ldots, n\}$, if $x_i \leq y$, then $A(x_1, \ldots, x_n) \leq A(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$;

(A2) A satisfies the boundary conditions: $A(0, \ldots, 0) = 0$ and $A(1, \ldots, 1) = 1$.

Let $r = (r1, \ldots, r_n)$ be a real n-dimensional vector, $r \neq 0$. A function $F : [0, 1]^n \to [0, 1]$ is said to be directionally increasing with respect to r (or r-increasing, for short) if for all $(x1, \ldots, x_n) \in [0, 1]^n$ and $c > 0$ such that $(x1 + cr1, \ldots, x_n + cr_n) \in [0, 1]^n$, it holds that $F(x1 + cr1, \ldots, x_n + cr_n) \geq F(x1, \ldots, x_n)$. Similarly, an r-decreasing function is defined.

Definition 2 [29, 37] An aggregation function $T : [0, 1]^n \to [0, 1]$ is a t-norm if, for all $x, y, z \in [0, 1]$, it satisfies the following properties:
(T1) Commutativity: $T(x, y) = T(y, x)$;
(T2) Associativity: $T(x, T(y, z)) = T(T(x, y), z)$;
(T3) Boundary condition: $T(x, 1) = x$.

If T satisfies only the property (T3) of t-norms (as well as its symmetric $T(1, x) = x$), then it is called a semi-copula. Some examples also used in [26] of t-norms are presented in Tabela 1.

Definition 3 [26] A bivariate function $C : [0,1]^2 \to [0,1]$ is a copula if it satisfies the following conditions, for all $x, x', y, y' \in [0,1]$ with $x \leq x'$ and $y \leq y'$:
(C1) $C(x,y) + C(x',y') \geq C(x,y') + C(x',y)$;
(C2) $C(x,0) = C(0,x) = 0$;
(C3) $C(x,1) = C(1,x) = x$.

In [26], four different commutative, non-associative copulas were considered, as defined in [2]. These four functions are shown in Tabela 2, where the following notation is used:

- **Max = max** $\{x, y\}$
- **Min = min** $\{x, y\}$
- **W = max**$\{0, x + y - 1\}$
- **P =** xy

For the definitions below, let $N = \{0, ..., n\}$.

Definition 4 [14, 30] A function $m : 2^N \to [0,1]$ is a discrete fuzzy measure if, for all $X, Y \subseteq N$, it satisfies the properties:
(m1) Monotonicity: If $X \subseteq Y$, then $m(X) \leq m(Y)$;
(m2) Boundary conditions: $m(\emptyset) = 0$ and $m(N) = 1$.

Definition 5 [14] Let $m : 2^N \to [0,1]$ be a discrete fuzzy measure. The discrete Choquet integral for m is defined as a function $\mathfrak{C}_m : [0,1]^n \to [0,1]$, given by

$$\mathfrak{C}_m(X) = \sum_{i=1}^{n} (x_{(i)} - x_{(i-1)}).m(A_{(i)}), \tag{1}$$

where $(x_{(i)}, \ldots, x_{(n)})$ is a non-decreasing permutation of the input x, that is, $0 \leq x_{(1)} \leq \ldots \leq x_{(n)}$, with the convention that $x_{(0)} = 0$, and $A_{(i)} = \{(i), \ldots, (n)\}$ is the subset of the $n - i + 1$ largest components of x.

Note that Equation 1 can also be written as:

$$\mathfrak{C}_m(X) = \sum_{i=1}^{n} (x_{(i)}.m(A_{(i)}) - x_{(i-1)}.m(A_{(i)})), \tag{2}$$

which is called the Choquet integral in its expanded form.

We now define the class of CαC-integrals.

Definition 6: Let $m : N^2 \to [0,1]$ be a fuzzy measure and $C\alpha : [0,1]^2 \to [0,1]$ be a family of copulas indexed by α. The family of discrete CαC-integrals with respect to m is defined as the function $\mathfrak{C}_m^{C_\alpha} : [0,1]^n \to [0,1]$, given, for all $x \in [0,1]^n$, by

$$\mathfrak{C}_m^{C_\alpha}(x) = \sum_{i=1}^{n} C\alpha(x_{(i)}, m(A_{(i)})) - C_\alpha(x_{(i-1)}, m(A_{(i)})), \tag{3}$$

Table 2: Example table of copulas

Copula ID	Functions	Property
(a)	$C_\alpha(x,y) = xy[1+\alpha(1-x)(1-y)]$	$-1 \leq \alpha \leq 1(\alpha \neq 0)$
(b)	$C_\alpha(x,y) = \frac{1}{1+\alpha}\max[x+y-1+\alpha-\alpha\lvert x-y\rvert, 0]$	$0 < \alpha < 1$
(c)	$C_\alpha = (1-\alpha)W + \alpha \min$	$0 < \alpha < 1$
(d)	$C_\alpha = \frac{\alpha^2(1-\alpha)}{2}W + (1-\alpha^2)P + \frac{\alpha^2(1+\alpha)}{2}\min$	$-1 < \alpha < 1(\alpha \neq 0)$

where $(x_{(i)}, \ldots, x_{(n)})$ is a non-decreasing permutation of the input x and $A_{(i)} = \{(i), \ldots, (n)\}$ is the subset of the indices of the $n-i+1$ largest components of x, with α assuming different ranges according to the adopted function.

Theorem 1 For any $\alpha \in [-1,1]$, bivariate copula $C : [0,1]^2 \to [0,1]$ and fuzzy measure $m : 2^N \to [0,1]$, $\mathfrak{C}_m^{C_\alpha}$ is a mean aggregation function.

The Choquet integral combines inputs by considering not only the importance of individual inputs or their magnitudes, but also the importance of the groups to which they belong (or coalitions they form). This allows assigning importance to all possible groups of criteria.

3 RELATED WORK

In this chapter, research efforts related to the scope of this dissertation are discussed. Therefore, studies were gathered that made significant contributions in the area of DDoS anomaly detection, fuzzy models, and network traffic management.

Chen *et al.* [12] has the objective of solving a low attack detection accuracy, through the unequal distribution of the network traffic. To deal with this problem a new approach called “the fuzzy entropy weighted natural nearest neighbor (FEW-NNN) method” is proposed to increase the accuracy and efficiency of flow-based network traffic attack detection. To conduct these tests, several datasets were utilized as samples for the study, specifically KDD99 and CIC-IDS-2017. Consequently, it was observed that the “FEW-NNN” method significantly enhances the accuracy and efficiency of flow-based network traffic attack detection, presenting a promising prospect in the field of network intrusion detection.

In Dalmazo *et al.* [16] a performance analysis of network traffic predictors in the cloud is presented, aiming at selected models of suitable predictions for cloud environments. The study presents a standardized analysis engine designed to evaluate candidate forecasting models based on accuracy, historical dependency, execution time, and computational efficiency. The conclusion of this work is drawn from the observed study results of the Dropbox case, it can be seen that all predictions based on local analysis show a considerable improvement using the Dynamic Window Size Algorithm (DyWiSA), also facilitating online traffic prediction due to its short dependence on historical data.

The Poisson Moving Average model, while yielding favorable results, maintained the same computational complexity as its counterparts, according to local analysis. Advancing to ARIMA, employed on the “Data Center"dataset, it demonstrated a considerable benefit over other predictors. However, this advantage was achieved through great computational complexity and time spent. The Poisson Moving Average which is more attractive due to its low cost of computational complexity, proved to be more appropriate for dynamic cloud environments compared to alternative models.

In ldhyani *et al.* [1], a hybrid model is proposed for predicting network traffic and enhancing Quality of Service (QoS). This model employs advanced time series predic-

tion techniques combined with fuzzy c-means clustering to analyze network data. Such integration improves the existing time series models, enabling the generation of “Clustering granules". The methodology adopted in the study involves analyzing various datasets and comparing the proposed model with alternative models presented during its introduction. The article concludes that the proposed model yields more satisfactory predictions, significantly improving the accuracy of network traffic forecasts with the aid of artificial intelligence.

In Novaes *et al.*[32], a modular system for anomaly detection and mitigation is presented, applied within SDN network environments. The proposed approach, based on Long Short-Term Memory with fuzzy logic (LSTM-FUZZY), is divided into three phases: characterization, anomaly detection, and mitigation. The system was tested in two scenarios: the first used IP flows collected from SDN Floodlight controllers through emulation in Mininet, and the second employed the CICDDoS 2019 dataset. The results show that the modules comprising the proposed system were effective in fulfilling their respective roles. Moreover, the system operates autonomously, improving execution by eliminating the need for human intervention. It was also concluded that applying an autonomous system supports the tasks assigned to the network administrator, enabling them to maintain and ensure operational efficiency, thereby facilitating management procedures. One of the system’s key strengths lies in its modular architecture, which allows for the integration and adaptation of different traffic characterization, anomaly detection, and mitigation techniques in SDN environments. This characteristic is essential to support the system’s adaptation as network dynamics evolve and new security demands arise.

Wani *et al.* [41] present three well-known machine learning algorithms used in the field of network security, Random Forest, Naïve Bayes, and Support Vector Machine, to detect Distributed Denial of Service (DDoS) attacks. The study aims to contribute to the mitigation of DDoS attacks, which are classified as critical threats due to their potential to compromise network availability. In this study, the authors used a few tools to set up the testing environment and simulate the attacks. They used OwnCloud, an open-source cloud platform, as the target for the attacks, while the DDoS attacks were carried out using the Tor Hammer tool. Everything was run on the Kali Linux 2018.2 system (Kernel 4.15.0, GNOME 3.28.0). The paper also introduced a dataset with 9 features and 4 class labels, used to evaluate their algorithms. To measure performance, they used metrics like precision, recall, specificity, and F-measure. Overall, the SVM algorithm showed the best results across all metrics, followed by Random Forest.

In addition, Jiang *et al.* [21] proposed a model for detecting network intrusions using Particle Swarm Optimization (PSO) combined with eXtreme Gradient Boosting (XGBoost). Their goal was to improve accuracy, fine-tune parameters, and select the most relevant features to create a more effective anomaly detection model. Compared to other techniques, the PSO-XGBoost model showed a clear improvement in detecting anoma-

lies. Even with the good results, the authors point out that achieving high precision in Network Intrusion Detection Systems (NIDS) remains a significant challenge, especially when dealing with anomaly detection.

Experimental results demonstrate that the PSO-XGBoost model usually outperforms traditional approaches such as Random Forest, Bagging, and AdaBoost, particularly in detecting minority-class attacks like U2R and R2L. However, the literature lacks studies that explore models for anomaly detection while maintaining low complexity, as fuzzy models do.

Aziz *et al.* [6] proposed a hybrid model to identify the core traffic of the network. First, a payload-based approach identifies most of the traffic. Any remaining unidentified traffic is then analyzed using a statistical unsupervised machine learning method, ensuring that no traffic is left unidentified. The main contributions include, capturing local IP traffic from a university network, identifying traffic using a payload-based method, labeling the unidentified traffic, and applying an unsupervised machine learning approach to classify the rest. In the proposed framework, DPI (Deep Packet Inspection) is employed as the first stage to classify application-layer traffic using predefined protocol signatures. Traffic flows that remain unidentified by DPI are then subjected to machine learning supervised or unsupervised methods. The authors developed and evaluated the framework using real traffic data, and the results demonstrate that the hybrid model effectively addresses the limitations of standalone DPI and machine learning methods. Although the study presents a hybrid system offering a solution for real-time traffic categorization, the proposal lacks a specialized solution for detecting anomalies in network traffic.

Following the networking traffic management field, Shetty *et al.* [38] presented a comprehensive study on the application of Artificial Intelligence (AI) and Machine Learning (ML) for intelligent network traffic control in modern communication systems. The primary objective is to address the limitations of traditional traffic management methods such as static routing, traffic shaping, and load balancing, which struggle to adapt to the dynamic and high-volume traffic of the today's networks, This perspective aligns closely with our work, as both studies seek to overcome the limitations of conventional network traffic management and employing artificial intelligent and learning mechanisms. The authors propose the integration of AI/ML techniques to enhance traffic prediction, congestion control, bandwidth optimization, latency reduction, and Quality of Service (QoS) improvements. Supervised, unsupervised, reinforcement, and deep learning models are analyzed for their effectiveness in managing traffic flows. Real-world case studies from companies like Vodafone, Akamai, and Cloudflare demonstrate the practical value of AI/ML in telecom and content delivery networks. While this study provides valuable insights into intelligent traffic control, it does not primarily focus on classification tasks or on reducing false positives, highlighting a different emphasis compared to our work.

Yao Hu and Bibo Tu [20] investigated an attack detection method based on the Fuzzy

C-Means (FCM) clustering algorithm. Their approach focuses on identifying abnormal network behavior by enabling the system to autonomously learn both typical and atypical service patterns while effectively managing complex network interactions. Once malicious traffic is detected through FCM clustering, the system assesses the severity of the intrusion and initiates appropriate countermeasures. To evaluate the performance of the FCM algorithm, the author used two metrics: miss rate and bit error rate. These results were then compared to those from K-Means and DTSOM. The findings showed that FCM performed better, with lower error rates than the other methods.

3.1 Summary of the related work

To provide a concise overview of the existing literature discussed in this chapter, Table 3 summarizes the main research topics addressed by each studie. The studies cover a wide range of contributions in areas such as anomaly detection, fuzzy logic, traffic prediction, and the use of machine learning models.

The references included in this summary are:

- **[1]** – Intelligent hybrid model to enhance time series models for predicting network traffic
- **[6]** – Towards accurate categorization of network ip traffic using deep packet inspection and machine learning
- **[12]** - Few-nnn: A fuzzy entropy weighted natural nearest neighbor method for flow-based network traffic attack detection
- **[16]** -Performance analysis of network traffic predictors in the cloud
- **[20]** – Security situation assessment model of ddos attack based on progressive fuzzy c clustering algorithm
- **[21]** – Network intrusion detection based on pso-xgboost model
- **[32]** – Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment
- **[38]** – Intelligent network traffic control with ai and machine learning
- **[41]** – Analysis and detection of ddos attacks on cloud computing environment using machine learning techniques

These works were selected due to their relevance in the fields of DDoS mitigation, intelligent traffic control, fuzzy-based models, and advanced detection mechanisms in modern network environments.

Table 3: Topics Covered by Each Article

Topic	[12]	[16]	[1]	[32]	[41]	[21]	[6]	[38]	[20]	**This Work**
Fuzzy Logic	X		X	X					X	X
Anomaly Detection	X			X	X	X			X	X
Traffic Prediction		X	X				X	X		X
Feature Selection	X					X				X
SDN				X						
Hybrid Model			X	X			X			
ML/AI Algorithms					X	X	X	X		X

3.2 Discussion of the related work

As mentioned earlier, these popular machine learning algorithms tend to achieve high accuracy, but they often face challenges when adapting to network behavior, which can lead to false positives and limited generalization. Additionally, although hybrid models can improve accuracy in classifying anomalies, their main focus is usually on categorizing traffic rather than directly detecting anomalies. As a result, some vulnerabilities may remain in the network without being properly addressed. Moreover, the studies presented in the literature typically rely on the use of raw attributes or, at most, apply feature selection techniques. However, they do not propose the generation of new transformed features that could enhance the expressiveness of the models. They also fail to address the problem of class imbalance, which is particularly critical for the detection of less frequent attacks.

To address these gaps in the literature, this master thesis proposes an anomaly detection method that leverages the Choquet integral to create new features and enhance detection performance. By incorporating fuzzy measures, the approach improves decision-making. Unlike traditional machine learning models [41], which rely heavily on accurate classifications, this method adaptively weighs traffic features, capturing complex dependencies and improving anomaly detection. In particular, this proposal aims to improve recall and F1-score for the minority class, which represents a significant advantage in the context of imbalanced datasets. As noted in the literature, two models commonly used in similar studies are XGBoost and Random Forest. Therefore, both models will be employed in this work to evaluate and compare the effectiveness of the proposed approach. This method offers a more flexible detection mechanism, capable of recognizing legitimate variations in network behavior and using them to improve detection accuracy.

4 CHOQUET INTEGRAL AND ADAPTIVE α PARAMETER

In the context of network anomaly detection, choosing the proper method to aggregate features is essential. This chapter explores the application of the generalized Choquet integral with an adaptive α parameter to improve prediction accuracy.

The Choquet integral has proven to be a powerful aggregation operator in fuzzy environments [39], particularly due to its ability to model interactions among input criteria through fuzzy measures. In this work, we build upon the theoretical foundation of the Choquet integral, introduced in theoretical section and apply its generalized form with parameterized copula functions to adapt the α and observe its behavior when predicting and modeling network traffic data.

4.1 Conceptual model

Figure 1 presents the basis of this work, highlighting its stages in detail showing each stage of the process, along with its principal components and interactions.

1. **Treated dataset**: Reflects the process of collecting, organizing, and refining raw data to make it ready for setup of the sliding windows (step 1 in Figure 1).

2. **Choquet averaging functions**: This step consists of implementing customized routines in the Python environment to apply the Choquet integral techniques. Once this process is completed, the data becomes ready to be fitted into the Choquet aggregation functions (Step 2 in Figure 1).

3. **Choquet functions**: The implementation of functions (step 3 in Figure 1) based on the Choquet integral for aggregation and prediction of network data requires a comprehensive understanding of both mathematical theory and computational techniques, this process was supported by the literature, more specifically in [27].

4. **The α variation**: The creation of search methods for varying the α parameter within the framework of the Choquet integral involves studying how it influences

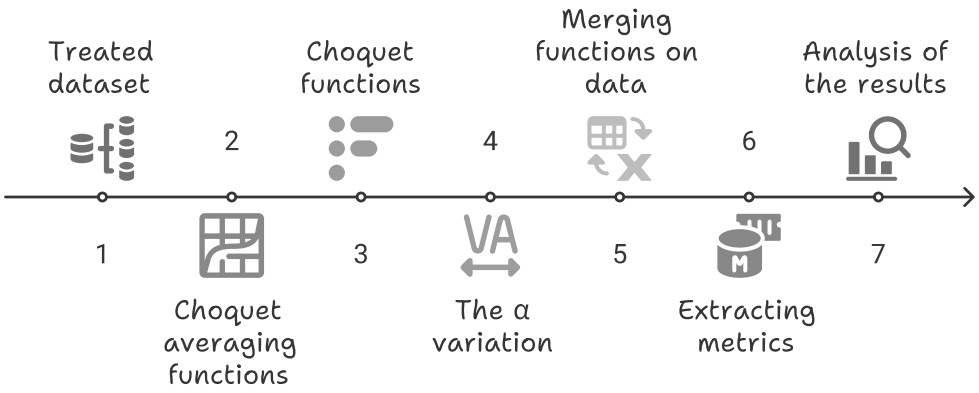

Figure 1: Conceptual model Choquet Integral and adaptive α parameter

the aggregation process and its implications for different attributes in the dataset. Designing search methods entails devising algorithms that systematically explore a range of α values, considering their impact on the resulting aggregation outcomes, i.e., minimizing the prediction error (step 4 in Figure 1).

5. **Merging functions on data**: The application of Choquet functions and the search for the α parameter in the context of network traffic involves integrating mathematical modeling with practical exploration of the α parameter. Initially, finding the most suitable α parameter entails developing algorithms to systematically explore the parameter space and identify values that yield the most accurate representations of network dynamics.

6. **Extracting metrics**: This process involves quantifying the prediction error associated with different α values. By comparing the predicted outcomes obtained with different α values to the actual observed outcomes, it is possible to assess the predictive performance of the Choquet functions across a range of α values. This analysis provides valuable insights into how variations in α affect prediction accuracy (step 5 in Figure 1).

7. **Analysis of the results**: Following the computation of evaluation metrics for each α parameter, the results are synthesized into a comprehensive report featuring tables

and graphs (step 6 in Figure 1). This report facilitates an analysis and interpretation of the performance of each α value and the efficiency of the methodology employed for predicting network traffic. By visually presenting the findings, stakeholders can gain insights into the impact of different α parameters on predictive accuracy and assess the overall effectiveness of the search methodology in optimizing prediction performance.

4.2 Dataset

We use the **CIC-IDS2017** dataset, a well known dataset used by Liangchen et al. [12] and in many other works, developed by the Canadian Institute for Cybersecurity (CIC). CIC-IDS2017 contains benign traffic and contemporary attacks captured over five consecutive days (July 3–7, 2017). Background traffic was generated with the B-Profile system to emulate realistic human behavior (25 users; HTTP, HTTPS, FTP, SSH, email), and flows were extracted and labeled with CICFlowMeter, yielding CSV files with 78 features plus the class label.

Subset used in this work. We adopt the Wednesday (working-hours) split (July 5, 2017), which combines normal activity with DoS/DDoS attacks. Starting from the public MachineLearningCSV files, we apply light preprocessing (type fixing, removal of non-informative IDs, and NaN/Inf handling) and keep the supervised label `Label`. The working table used throughout this chapter has **692,703 flow instances** and **79 attributes**. A short sample of the processed table is shown in Figure 2.

To ensure reproducibility and interpretability, we report: (i) row/column counts, (ii) the target and class distribution for Wednesday working-hours in Table 4 and a sample of the dataset in Figure 2

Table 4: Dataset metadata (CIC-IDS2017, Wednesday working-hours)

Source	Canadian Institute for Cybersecurity (CIC)
Window	Wed, July 5, 2017 (working hours)
Granularity	Bidirectional flow records (CSV)
Attacks (Wed)	slowloris, slowhttptest, HULK, GoldenEye
Original features	78 (CICFlowMeter) + `Label`
Working table	692,703 instances $\times$ 78 attributes
Label distribution (binary)	Non-attack: 440,031 (63.52%); Attack: 252,672 (36.48%)
Target	`Label` (benign vs. attack subclasses)

	Bwd Packet Length Mean	Avg Bwd Segment Size	Bwd Packet Length Max	Packet Length Std	Max Packet Length	Fwd IAT Max	Flow IAT Max	Packet Length Mean	Packet Length Variance	Average Packet Size	Label
0	6.000000	6.000000	6	0.000000	6	0	38308	6.000000	0.000000e+00	9.000000	BENIGN
1	65.200000	65.200000	163	56.529599	163	109	73	29.294118	3.195596e+03	31125.000000	BENIGN
2	525.000000	525.000000	1575	671.751541	1575	915	810	370.588235	4.512501e+05	393.750000	BENIGN
3	705.000000	705.000000	2077	747.760984	2077	2615	5002306	386.600000	5.591465e+05	429.555556	DoS GoldenEye
4	2326.400000	2326.400000	8736	2478.420729	8736	3050	5001318	926.846154	6.142569e+09	1004.083333	DoS GoldenEye
5	1938.666667	1938.666667	5792	1855.776448	5792	2847	5000674	858.071429	3.443906e+09	924.076923	DoS GoldenEye
6	1932.500000	1932.500000	4355	1886.332364	4355	675	577	1197.700000	3.558250e+09	1330.777778	DoS Hulk
7	1932.500000	1932.500000	5792	2119.419944	5792	619	658	1196.700000	4.491941e+06	1329.666667	DoS Hulk
8	0.000000	0.000000	0	0.000000	0	304	304	0.000000	0.000000e+00	0.000000	DoS Hulk
9	654.666667	654.666667	1964	650.575472	1964	5415	5415	257.777778	4.232484e+05	290.000000	DoS Slowhttptest
10	0.000000	0.000000	0	183.603562	520	110000000	110000000	65625.000000	3.371027e+04	75.000000	DoS Slowhttptest
11	0.000000	0.000000	0	183.332592	520	110000000	110000000	66375.000000	3.361084e+04	75.857143	DoS Slowhttptest
12	703.666667	703.666667	3525	895.115146	3525	78311	4951173	354.866667	8.012311e+05	380.214286	DoS slowloris
13	0.000000	0.000000	0	0.000000	6	229	229	6.000000	0.000000e+00	9.000000	DoS slowloris
14	0.000000	0.000000	0	3.286335	6	5000925	5000925	2.400000	1.080000e+01	3.000000	DoS slowloris
15	4370.686524	4370.686524	17376	2669.389319	17376	5025702	5024984	1713.525708	7.125639e+09	1713.913910	Heartbleed
16	3733.713270	3733.713270	14480	2414.090913	14480	996350	995350	1610.776871	5.827835e+09	1611.105467	Heartbleed
17	3699.312676	3699.312676	13032	2381.909586	13032	996402	995259	1603.277970	5.673493e+09	1603.603574	Heartbleed

Figure 2: Sample of the network traffic cic 2017 dataset (692,703 rows × 11 columns).

4.3 The α influence and search algorithms

To explore the influence of different α in the Choquet integral generalizations (Table 2) [27], we introduce an adaptive α [35], which influences aggregation and prediction by becoming more or less sensitive to particular value combinations in the data, enabling finer control over how input features contribute to the output. To evaluate the impact of this adaptation on prediction quality, we use the Mean Absolute Error (MAE) as the performance metric [35].

To identify the most suitable value of α for each copula, we developed three distinct search strategies:

1. **Brute force search**, which evaluates all possible α values within a defined interval;
2. **Binary search**, which iteratively narrows the search space to converge on the optimal α;
3. **Randomized binary search**, which introduces stochasticity in the midpoint selection to avoid local minima.

4.3.1 Brute force method

The brute force search algorithm, referenced as Algorithm 1, aims to optimize a parameter called α by minimizing the error associated with the estimated value and the true value. This algorithm conducts search operations for each α, utilizing two decimals, and compares all the values of the prediction errors. It then returns the variables "bestAlpha", "bestError", and "numInteractions" as a response. These variables are utilized to determine the best α value for each equation (refer to Table 2) and store the error obtained for each α value.

4.3.2 Binary search method

The binary search algorithm presented aims to optimize a parameter called "alpha", by minimizing an associated error. Below is presented a detailed summary of the procedure

and an example of the Algorithm 2.

4.3.3 Random binary search method

The random binary search method is identical to the binary search, with just one line of code changed as it shows in Algorithm 3. To achieve the random binary algorithm it has to change only line 4 of the Algorithm 2. The difference in this approach is that instead of taking the middle value between aMin and aMax, a random number is chosen within these two ranges. This results in varying values for testing α.

Algorithm 1 Brute Force Search

Input: alpha_min, alpha_max, choquet
Output: bestAlpha, bestError, numIterations

1: $bestError \leftarrow \infty$; $bestAlpha \leftarrow 0$
2: $errorGraphTable \leftarrow null$; $numIterations \leftarrow 0$
3: $alphaValues \leftarrow genApList(alpha\_min, alpha\_max)$
4: $indexMin \leftarrow 0$
5: $totalError, errorGraphTable \leftarrow Choquet$
6: $smallestError \leftarrow totalError[0]$
7: **for** $i \leftarrow 0$ **to** $len(totalError) - 1$ **do**
8: **if** $totalError[i] < smallestError$ **then**
9: $smallestError \leftarrow totalError[i]$
10: $indexMin \leftarrow i$
11: **end if**
12: **end for**
13: $refValues \leftarrow [\,]$
14: **if** $alphaValues[indexMin] \geq 0$ **then**
15: **for** $i \leftarrow 0$ **to** 9 **do**
16: $refValues.add(alphaValues[indexMin] + i \times 0.01)$
17: **end for**
18: **else**
19: **for** $i \leftarrow 0$ **to** 9 **do**
20: $refValues.add(alphaValues[indexMin] - i \times 0.01)$
21: **end for**
22: **end if**
23: $alphaValues \leftarrow refValues$
24: $totalError, errorGraphTable \leftarrow Choquet$
25: $smallestError \leftarrow totalError[0]$
26: $indexMin \leftarrow 0$
27: **for** $i \leftarrow 0$ **to** $len(totalError) - 1$ **do**
28: **if** $totalError[i] < smallestError$ **then**
29: $smallestError \leftarrow totalError[i]$
30: $indexMin \leftarrow i$
31: **end if**
32: **end for**
33: **return** $alphaValues[indexMin]$, $smallestError$

Table 5: Search results for α using Brute Force Search

Method	Best α	Error AVG	Time
Choquet a	-1	1068.49	49s
Choquet b	0.6	1120.41	49s
Choquet c	0.23	1332.44	132s
Choquet d	-0.9	1249.12	207s

Table 6: Search results for α using Binary and Random Binary Search

	Binary Search			**Random Binary Search**		
Method	Best α	Error AVG	Time	Best α	Error AVG	Time
Choquet a	-0.98	1069.14	19s	-0.99	1068.55	17s
Choquet b	0.49	1121.34	23s	0.49	1180.04	25s
Choquet c	0.25	1332.66	63s	0.15	1343.75	55s
Choquet d	-0.87	1242.47	84s	-0.64	1429.20	80s

The comparison of methods revealed clear differences in efficiency, precision, and execution time. The brute force method was the slowest, taking 7 minutes and 19 seconds. In contrast, random binary search and binary search reduced the execution time to 2 minutes and 57 seconds and 3 minutes and 9 seconds, respectively. Using the speedup metric, random binary search achieved an improvement of **147.58%**, while binary search achieved **132.28%** over brute force. Binary search algorithm 2 demonstrated the best balance between precision and execution time. The brute force search, while somewhat accurate, is impractical for large input files due to its high execution time. Random binary search is faster but relies on randomness. Additionally, data dimensionality significantly impacts performance, increasing execution time even with the best binary search method.

4.3.4 Setup of the environment

All experiments were conducted on a Google Colab virtual machine provisioned with an Intel(R) Xeon(R) CPU @ 2.20 GHz (1 core, 2 threads) and 13 GB of RAM, using Colab version `1.0.0 (2023-12-18)`. The entire pipeline was implemented in Python 3 (`3.10.12`). To ensure reproducibility, we kept the same hardware profile and interpreter version across all runs.

4.4 Discussion of results

To explore the influence of different α in the Choquet integral generalizations (Table 2) [27], we introduce an adaptive α [35], which influences aggregation and prediction by becoming more or less sensitive to particular value combinations in the data, enabling finer control over how input features contribute to the output. To evaluate the impact of this adaptation on prediction quality, we use the Mean Absolute Error (MAE) as the

Algorithm 2 Binary Search

Input: $alpha_{min}$, $alpha_{max}$, $tolerance$
Output: $bestAlpha$, $bestError$, $numIterations$

$bestError \leftarrow \infty$; $bestAlpha \leftarrow 0$; $pError \leftarrow \infty$
$stagnantTries \leftarrow 0$; $numIterations \leftarrow 0$
while $stagnantTries < decimals$ **do**
    $curAlpha \leftarrow (alpha_{min} + alpha_{max})/2$
    $curError, errorBDF \leftarrow calcError(curAlpha)$
    if $curError < bestError$ **then**
        if $abs(curError - bestError) < tolerance$ **then**
            $stagnantTries \leftarrow stagnantTries + 1$
        else
            $stagnantTries \leftarrow 0$
        end if
        $bestError \leftarrow curError$
        $bestAlpha \leftarrow curAlpha$
        $errorDisplay \leftarrow errorBDF$
    else
        if $numIterations \geq decimals$ **then**
            $stagnantTries \leftarrow stagnantTries + 1$
        end if
        if $curError < pError$ **then**
            $alpha_{max} \leftarrow curAlpha$
        else
            $alpha_{min} \leftarrow curAlpha$
            $pError \leftarrow curError$
        end if
        $numIterations \leftarrow numIterations + 1$
    end if
end while
return $bestAlpha$, $bestError$, $numIterations$

Algorithm 3 Random binary search different term

...
...
...
$currentAlpha \leftarrow random.uniform(aMin, aMax)$

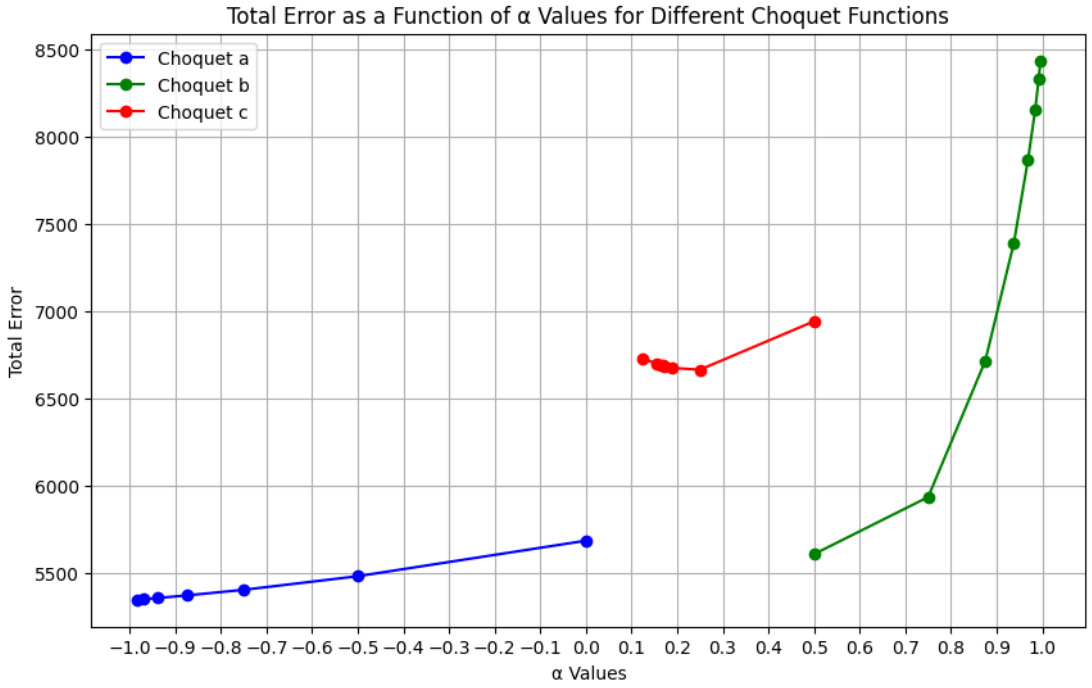

Figure 3: Total error depending on the values of α for Choquet a, b, and c

performance metric [35].

This study analyzed the generalizations of Choquet integrals utilizing various α-based functions applied to network traffic prediction [35]. Through the evaluation of results, it was observed that the binary search method demonstrated efficiency by reducing execution time and minimizing errors in network traffic prediction. Compared to the brute force method, which represents the worst-case scenario, the binary search and the random binary search achieved speedups of **132.28%** and **147.58%**, respectively. The values of α obtained varied between the methods, indicating the sensitivity of the parameter in different approaches [35]. By correlating the parameter with the improvement of the mean absolute errors, it was possible to observe that it varies for each equation in (Table 2). More detailed insights can be obtained by examining the graphs of α as a function of Total Error in Figure 3 and 4.

In summary, all equations exhibit different error behavior based on the input of the parameter α. The only equation that differs significantly in exponential levels is Choquet d (Figure 4), with error values reaching over 70,000 when approaching its maximum possible α value.

Regarding the limitation of these methods, the brute force search [34], although somewhat accurate, is time-consuming and impractical for large input files, while random binary search, despite being faster, relies on randomness. Another factor that demands considerable effort is the data dimensionality, as time increases significantly for tests, even when utilizing the best binary search method.

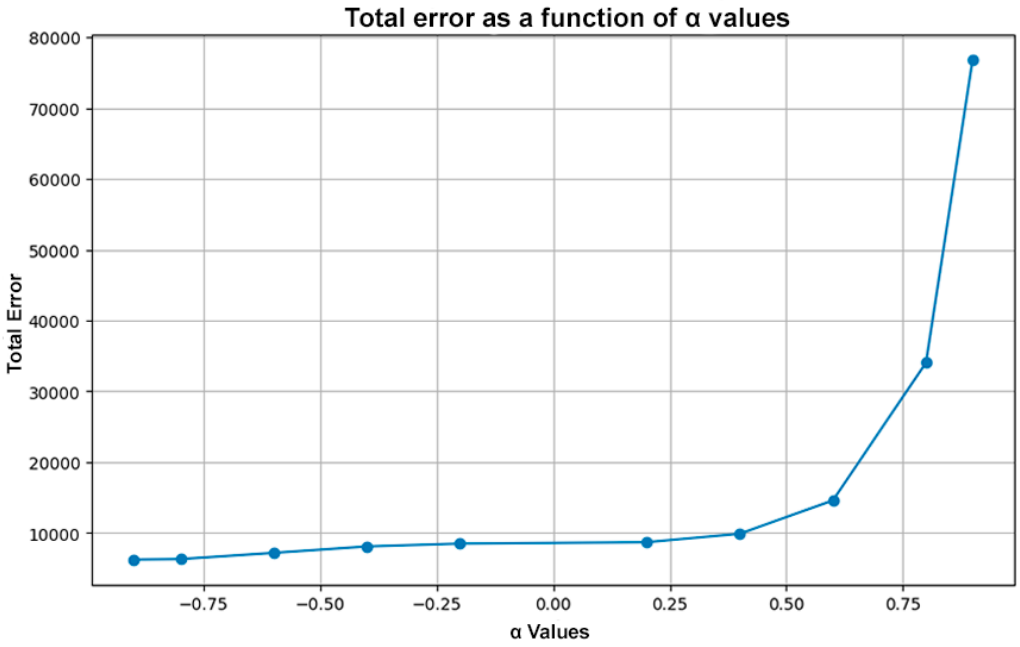

Figure 4: Total error depending on the values of α for Choquet d

5 FUZZY FEATURE ENGINEERING FOR ANOMALY DETECTION

Despite various methodologies in the literature aimed at enhancing existing algorithms such as Random Forest and XGBoost, no prior study has applied the generalized Choquet integral to generate features for improving predictive performance [36]. Addressing this gap, this dissertation proposes a novel approach to feature engineering, aiming to enhance anomaly detection regarding DDoS attacks. To achieve this, we first employ KBest [33] as a feature selection algorithm to identify the most relevant attributes for classification using Random Forest and XGBoost. Once the optimal set of features is determined, we then process each of them using a Generalized Choquet Integral with an adaptive α. After selecting the top-k features with KBest, every selected feature is individually transformed through the Choquet integral, generating new Choquet-features. The experiments are conducted incrementally, using only the original features (without Choquet), then comparing one original feature with its corresponding Choquet transformed version, then two original features with their two Choquet-features, and so on, until all of them are evaluated. This procedure makes it possible to assess how the progressive, pairwise incorporation of Choquet-feature representations influences model accuracy, stability, and efficiency in anomaly detection.

5.1 Conceptual model

Figure 5 provides the foundation of this work, detailing its stages and illustrating each step of the process along with its main components and interactions.

1. **Data**: A networking DDoS attack dataset is selected as the initial input for the experiments, containing labeled traffic for classification. (Step 1 in Figure 5)

2. **Data preprocessing**: The dataset is processed through cleaning and formatting steps to improve efficiency and ensure compatibility with the machine learning algorithms. (Step 2 in Figure 5)

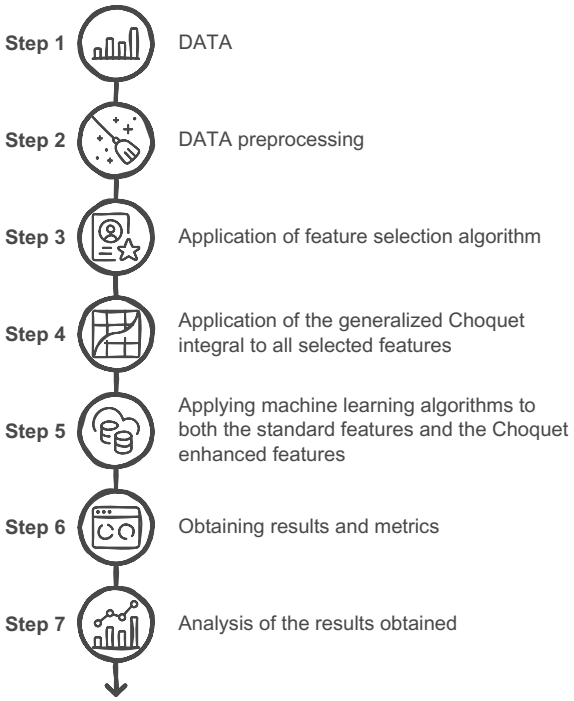

Figure 5: Conceptual model fuzzy feature engineering for anomaly detection

3. **Application of feature selection algorithm**: After preprocessing, a feature selection algorithm is employed to identify and retain the most informative features for the prediction process. (Step 3 in Figure 5)

4. **Application of the generalized Choquet integral to all selected features**: The top ranked features is chosen, and the generalized Choquet integral with adaptive α is applied to generate the new set of features by aggregating values and predicting behavioral trends. (Step 4 in Figure 5)

5. **Applying machine learning algorithms to both the standard features and the Choquet enhanced features**: As a baseline, machine learning algorithms are applied first using only the original selected features and then in the Choquet-features. (Step 5 in Figure 5)

6. **Obtaining results and metrics**: The evaluation is performed incrementally: first, one original feature is compared with its single Choquet-based counterpart, then two original features are compared with their two Choquet-based versions, then three, and so on. At each step, performance metrics such as accuracy, recall, and F1-score are computed and recorded. (Step 6 in Figure 5)

7. **Analysis of the results obtained**: A comparative analysis of the collected metrics is conducted to understand the impact and added value of the new features on anomaly detection. (Step 7 in Figure 5)

5.2 Dataset

This section reuses the **same dataset** described earlier in Section 4.2: the **CIC-IDS2017** (Wednesday, working-hours) split captured on July 5, 2017. As detailed in Table 4, flows were extracted and labeled with CICFlowMeter, and our working table comprises **692,703** instances and **78** predictive features plus the `Label` column.

To avoid duplication, we only recall the key elements relevant to this chapter, the binary aggregation of the target into Non-attack vs Attack (**440,031** vs. **252,672** instances, 63.52%/36.48%), all preprocessing steps (type fixing, removal of non-informative IDs, NaN/Inf handling) are identical to those reported in Section 4.2.

5.3 Implementation

For the implementation of this work, we used four algorithms: Random Forest, SelectKBest, XGBoost, and the Choquet Integral, the first three being well-known and widely used in the academic community.

It is worth noticing that SelectKBest [33] is a machine learning technique used for feature selection, and it was applied in this work to identify and choose the most relevant variables from the dataset. This approach enhances model performance while reducing the risk of overfitting. SelectKBest is one of the most widely used methods within this technique and falls under the category of filter-based feature selection. It applies statistical tests such as Chi-Squared, ANOVA F-test, and Mutual Information Score to rank features based on their relationship with the target variable, selecting the top K features with the highest scores.

To predict DoS attacks, we employed two machine learning algorithms: Random Forest and XGBoost. Although both are based on decision trees, they utilize distinct techniques to enhance accuracy and robustness. Random Forest is an ensemble learning method that constructs multiple decision trees from different subsets of the training data. By selecting random features at each split, it reduces overfitting and improves generalization. The final prediction is determined by aggregating the outputs of all trees, typically through majority voting in classification tasks. This approach not only enhances robustness to noise but also provides valuable internal estimates for error monitoring. On the other hand, XGBoost (eXtreme Gradient Boosting) is a highly efficient and scalable gradient-boosting algorithm. Unlike Random Forest, it builds decision trees sequentially, with each new tree correcting the errors of the previous ones using gradient descent optimization. XGBoost is well known for its speed and ability to handle large datasets. We can have a look at how it behaves in the Algorithm 4.

The Choquet-based algorithm employed in this work [35] was developed using copulas [27], as presented in Table 2, it is used for data aggregation and trend prediction. The algorithm takes a selected feature as input and processes it using specific techniques to

compute the mean absolute error (MAE) associated with each of the four copulas and its α related to them. The copula and the α with the lowest MAE are then selected to produce the final output, which will be used in the classification algorithms.

5.4 Evaluation

To prove the technical feasibility of the work, this section evaluates the influence of using the Generalization of the Choquet integral as a feature engineer and its results when used in algorithms to detect anomalies.

5.4.1 Generalization of the Choquet Integral

This section addresses the implementation of a prediction model using the Choquet Integral applied to the features extracted from the dataset, where the mean absolute error (MAE) analysis is performed with different copulas (table 2) and α values associated. Its possible to observe an algorithm example in (algorithm 2) of how the best α of each copula was identified, the parameter such as upper and lower bounds to get the most useful α was selected after an analysis of the metrics from the MAE combined with number of interactions, was concluded that after one hundred interactions the continuation of the search algorithm becomes irrelevant, returning a minimally acceptable improvement values.

First, when running the Choquet algorithm, we computed the tendencies of each previously imputed feature value for each copula. Next, we applied a binary search to identify the optimal α value for each copula. The lowest error was found after 8 iterations of the binary search algorithm. The best result for our dataset was achieved with $\alpha = 0.5342$, using the first copula (Table 2) in the generalized Choquet integral.

5.4.2 Data preprocessing and feature selection algorithm

Before selecting and generating new features, several preprocessing steps were performed as part of this study to ensure data quality and improve model performance.

Since we started with the raw CIC-IDS 2017 dataset, data preprocessing was necessary. This process was carried out using the Pandas and NumPy libraries for data manipulation and cleaning. The preprocessing steps included converting categorical columns into numerical values using techniques such as one-hot encoding and label encoding, handling missing data through mean/mode imputation, and balancing the dataset with undersampling and SMOTE (Synthetic Minority Over-sampling Technique) [11] to address class imbalances.

Once the data was preprocessed, the SelectKBest algorithm from the Scikit-learn library was applied with a parameter value of 10, using the ANOVA F test, a scoring function to rank and select the ten most relevant features for prediction. This step helped

reduce dimensionality while retaining the most informative attributes, as there were 78 features previously and then 10 features were selected with the highest scores, which we can see listed below along with their respective scores (Table 7):

The choice of selecting only the top 10 features was intentional. Preliminary experiments showed that, for both Random Forest and XGBoost, the performance tends to stabilize once more than 6–7 features are used as shown in the performance curves in Figures 6 and 7, therefore, limiting the feature space to 10 attributes prevents the model from incorporating unnecessary information, while still allowing the evaluation of how performance evolves as the feature space gradually expands. This makes the analysis more controlled and avoids feeding excessive or redundant features to the classifiers.

Table 7: Top-10 selected features based on SelectKBest

Rank	Feature	Score
1st	Bwd Packet Length Mean	112023.06
2nd	Avg Bwd Segment Size	112023.00
3rd	Bwd Packet Length Std	108310.46
4th	Bwd Packet Length Max	106125.98
5th	Packet Length Std	101429.54
6th	Max Packet Length	94126.52
7th	Fwd IAT Max	93922.66
8th	Flow IAT Max	93525.75
9th	Packet Length Mean	84086.98
10th	Packet Length Variance	82271.07

5.4.3 Application of Random Forest and XGBoost algorithms on data

After preprocessing the data, selecting new features, and generating the additional feature, the dataset was ready for the prediction phase. To carry out the predictions, two algorithms explained in Subsection 2.2 were used: Random Forest and XGBoost. For each algorithm, two tests were conducted: the first without the new generated features and the second with the new features inclusion. This approach provided insights into the model's improvement comparing both results.

It is important to highlight the parameters used for both algorithms, Random Forest and XGBoost, you can see it in Table 8 and Table 9 respectively.

Table 8: Hyperparameters used for Random Forest

Parameter	Value	Description
`n_estimators`	100	Number of trees in the forest
`max_depth`	10	Maximum depth of each tree
`random_state`	42	Ensures reproducibility
`stratify`	y	Maintains class distribution in train/test split

Table 9: Hyperparameters used for XGBoost

Parameter	Value	Description
`n_estimators`	100	Number of boosting rounds
`max_depth`	6	Maximum depth of each tree
`learning_rate`	0.1	Step size for each boosting iteration
`subsample`	0.8	Percentage of data used per boosting round
`colsample_bytree`	0.8	Percentage of features used per tree
`random_state`	42	Ensures reproducibility
`use_label_encoder`	False	Avoids deprecated label encoding warnings
`eval_metric`	logloss	Loss function for evaluation

5.4.4 Accuracy Comparison: With vs. Without Choquet Feature

Figure 6, 7 and Tables 10, 11 present the accuracy evolution as the feature space grows (from 1 to 10 features), comparing the traditional feature set against the configuration that incorporates the Choquet-based feature. As can be observed, when only a small number of features is available, the configuration with the Choquet–based feature yields consistently higher accuracy compared to the baseline without aggregation. As the number of features increases, the performance gap gradually narrows, and both approaches converge to similar accuracy levels once additional raw features are incorporated. The best k for Random Forest and XGBoost, the largest accuracy gain occurs at $k = 4$ ($\Delta Y = 0.07$), indicating an effective positve operating window at $k \in [1, 5]$ before both curves saturate and converge for $k \geq 6$.

Table 10: Accuracy difference of Y (ΔY) for the First Ten Features (Random Forest).

Feature	Y_1	Y_2	$\Delta Y = Y_1 - Y_2$
1	0.931	0.870	0.061
2	0.934	0.870	0.064
3	0.941	0.870	0.071
4	**0.953**	**0.880**	**0.073**
5	0.948	0.909	0.039
6	0.962	0.990	-0.028
7	0.964	0.984	-0.020
8	0.965	0.985	-0.020
9	0.964	0.984	-0.020
10	0.963	0.983	-0.020

Table 11: Accuracy difference of Y (ΔY) for the First Ten Features (XGBoost).

Feature	Y_1	Y_2	$\Delta Y = Y_1 - Y_2$
1	0.924	0.860	0.064
2	0.923	0.857	0.066
3	0.934	0.864	0.070
4	**0.951**	**0.880**	**0.071**
5	0.947	0.903	0.044
6	0.962	0.986	-0.024
7	0.964	0.986	-0.022
8	0.965	0.986	-0.021
9	0.966	0.987	-0.021
10	0.967	0.988	-0.021

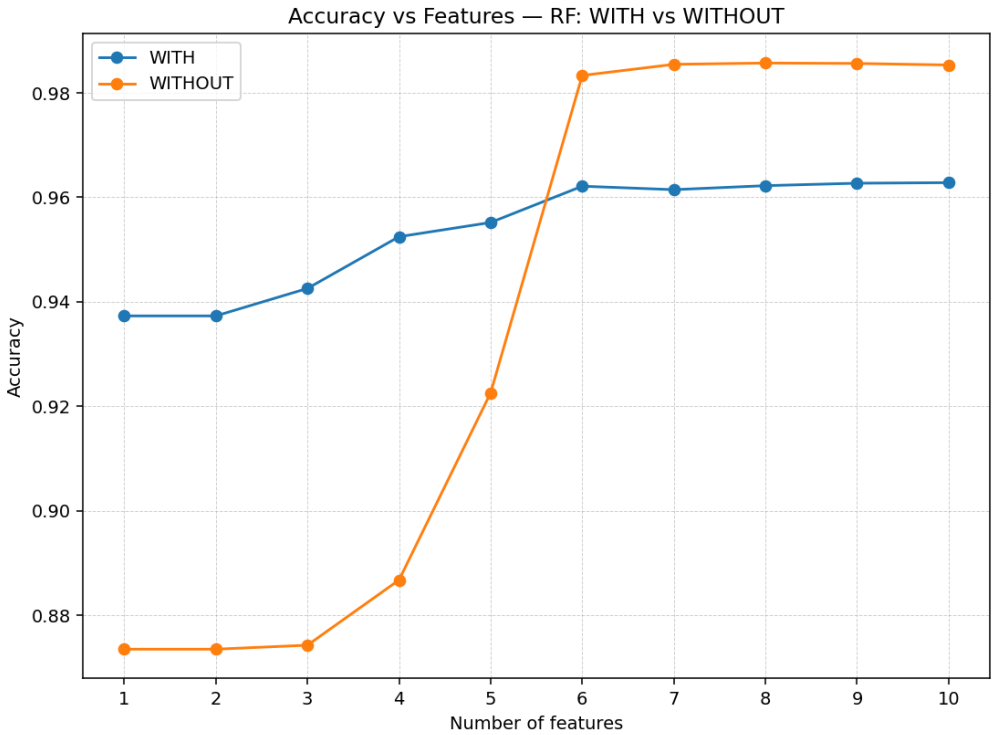

Figure 6: The evolution of the accuracy according to the number of features (Random Forest)(With Choquet and Without Choquet).

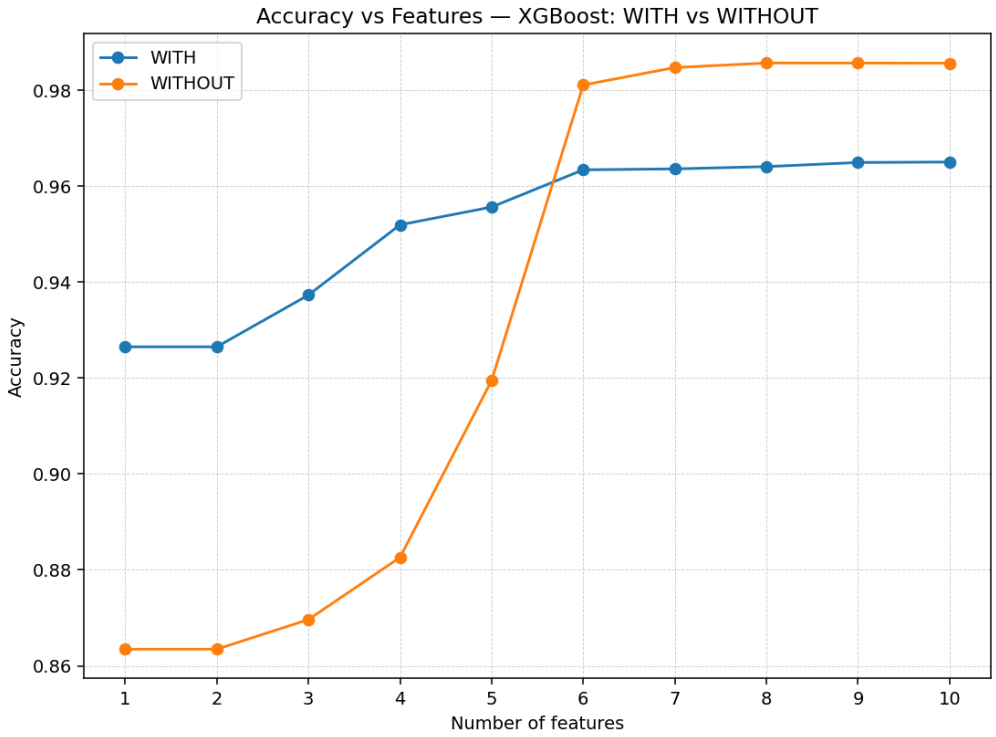

Figure 7: The evolution of the accuracy according to the number of features (XGBoost) (With Choquet and Without Choquet).

5.4.5 Other analysis metrics

In this subsection, we examine the additional performance metrics—precision, recall, and F1-score—using the results presented in Tables 12, 14, and 13 for Random Forest, as well as Tables 15, 17, and 16 for XGBoost. These tables allow us to compare the baseline feature set against the Choquet-feature pipeline and to understand how fuzzy aggregation affects model behavior across different feature counts. The discussion below summarizes the main trends and highlights the feature configurations where the Choquet-based representation offers the most substantial improvements.

Overall, the additional metrics (precision, recall, and F1-score) in both Random Forest (Tables 12, 14, 13) and XGBoost (Tables 15, 17, 16) confirm the same pattern observed earlier with accuracy, the Choquet-feature produces its strongest impact when the feature set is small, particularly between one and five features. Among these configurations, four features consistently yields the largest and most stable gains, appearing as the best point in the most of the three metrics for both classifiers. Although three features occasionally achieves competitive improvements, the results at four features are more systematic across models.

The improvements in precision indicate fewer false positives, meaning the classifier is better at avoiding unnecessary alerts on benign traffic, higher recall shows that the model misses fewer attacks, which is essential for effective anomaly detection and the increase in

F1-score reflects a better balance between these two aspects, resulting in a more reliable classifier overall.

Taken together, these findings demonstrate that the Choquet aggregation helps the models capture meaningful interactions early in the feature ranking process, enabling strong performance even with a small set of attributes. However, when considering the full set of k attributes, the benefits diminish as both pipelines approach their performance ceiling, with results showing that beyond 6 features, the inclusion of additional attributes leads to a slight decline in model effectiveness.

Table 12: Precision for the First Ten Features (Random Forest).

Feature	Precision W/ Choquet-feature $_1$	Precision dataset features $_2$	$\Delta = P_1 - P_2$
1	0.928	0.891	0.037
2	0.928	0.891	0.037
3	0.935	0.893	0.042
4	**0.951**	**0.902**	**0.049**
5	0.955	0.933	0.022
6	0.960	0.983	-0.023
7	0.959	0.985	-0.026
8	0.961	0.985	-0.024
9	0.961	0.985	-0.024
10	0.961	0.985	-0.024

Table 13: F1-score for the First Ten Features (Random Forest).

Feature	F1-score W/ Choquet-feature $_1$	F1-score dataset features $_2$	$\Delta = F_1 - F_2$
1	0.926	0.863	0.063
2	0.926	0.863	0.063
3	**0.931**	**0.864**	**0.067**
4	0.944	0.880	0.064
5	0.947	0.922	0.025
6	0.956	0.983	-0.027
7	0.955	0.985	-0.030
8	0.956	0.985	-0.029
9	0.957	0.985	-0.028
10	0.957	0.985	-0.028

Table 14: Recall for the First Ten Features (Random Forest).

Feature	Recall W/ Choquet-feature $_1$	Recall dataset features $_2$	$\Delta = R_1 - R_2$
1	0.937	0.873	0.064
2	0.937	0.873	0.064
3	**0.942**	**0.874**	**0.068**
4	0.952	0.886	0.066
5	0.955	0.922	0.033
6	0.962	0.983	-0.021
7	0.961	0.985	-0.024
8	0.962	0.985	-0.023
9	0.962	0.985	-0.023
10	0.962	0.985	-0.023

Table 15: Precision for the First Ten Features (XGBoost).

Feature	Precision W/ Choquet-feature $_1$	Precision dataset features $_2$	$\Delta = P_1 - P_2$
1	0.897	0.870	0.027
2	0.897	0.870	0.027
3	0.921	0.885	0.036
4	**0.937**	**0.896**	**0.041**
5	0.941	0.930	0.011
6	0.960	0.981	-0.021
7	0.960	0.984	-0.024
8	0.961	0.985	-0.024
9	0.963	0.985	-0.022
10	0.963	0.985	-0.022

Table 16: F1-score for the First Ten Features (XGBoost).

Feature	F1-score W/ Choquet-feature $_1$	F1-score dataset features $_2$	$\Delta = F_1 - F_2$
1	0.911	0.852	0.059
2	0.911	0.852	0.059
3	0.926	0.859	0.067
4	**0.944**	**0.876**	**0.068**
5	0.947	0.919	0.028
6	0.958	0.981	-0.023
7	0.958	0.984	-0.026
8	0.958	0.985	-0.027
9	0.959	0.985	-0.026
10	0.959	0.985	-0.026

Table 17: Recall for the First Ten Features (XGBoost).

Feature	Recall W/ Choquet-feature $_1$	Recall dataset features $_2$	$\Delta = R_1 - R_2$
1	0.926	0.863	0.063
2	0.926	0.863	0.063
3	0.937	0.869	0.068
4	**0.951**	**0.882**	**0.069**
5	0.955	0.919	0.036
6	0.963	0.981	-0.018
7	0.963	0.984	-0.021
8	0.964	0.985	-0.021
9	0.964	0.985	-0.021
10	0.965	0.985	-0.02

5.4.6 Detection performance

To better understand how the Choquet-based feature impacts the classifiers, confusion matrices were generated for both algorithms under two conditions: (i) using only the raw selected features, and (ii) using the same features with the addition treatment of the Choquet-aggregated feature. Remember that 4 features are used for Random Forest and XGBoost, since Tables 10 and 11 indicate that these configurations yield the highest accuracy for each model and so well the best (ΔY).

For Random Forest and XGBoost Without Choquet in Figure 8, the confusion matrix shows that the model correctly classifies 83% of benign traffic (true label 0) and 99% of attacks (true label 1), with 17% of benign samples flagged as attacks (false positives) and only 1% of attacks missed (false negatives). With the Choquet feature in Figure 9, benign detection improves substantially to 97% (false positives drop to 3%), while attack detection decreases to 93% (false negatives rise to 7%). This reflects a clear *trade-off*: the Choquet aggregation greatly reduces false alarms on benign traffic, at the cost of a higher miss rate on attacks; the baseline does the opposite, identifying attacks well but misclassifying more benign samples.

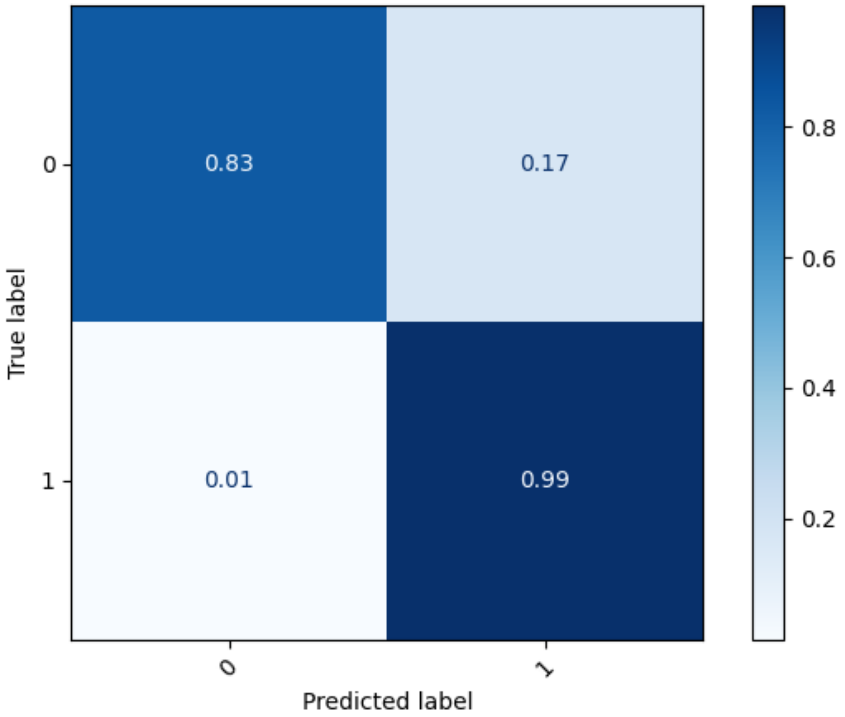

Figure 8: Confusion matrix of the classifier trained using only the original selected features (baseline) for 4 features (best case).

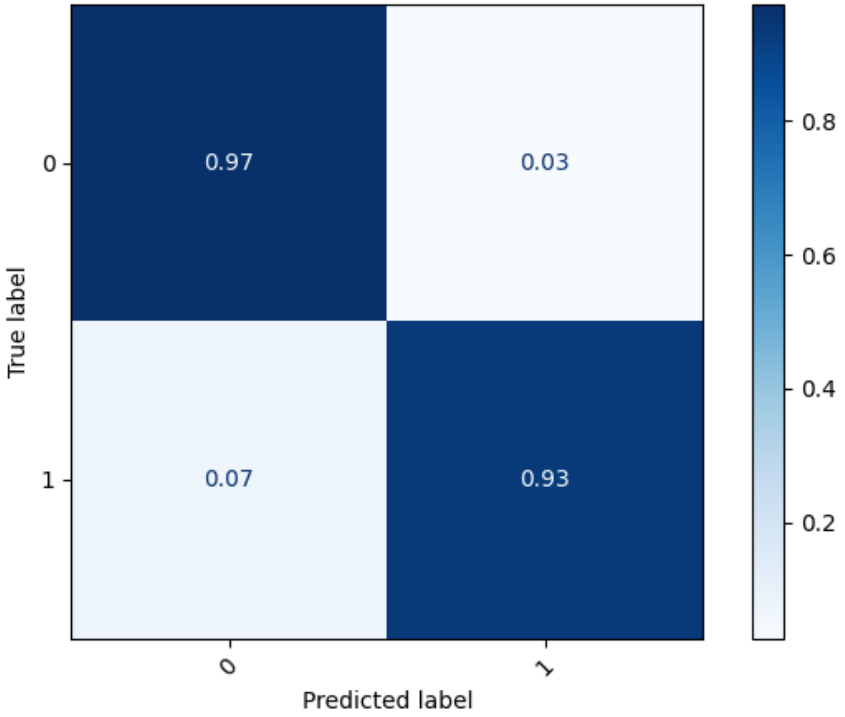

Figure 9: Confusion matrix of the classifier trained with the addition of the generalized Choquet-based feature for 4 features (best case).

5.5 Discussion of the results

Through the data and tables presented, we will discuss the results obtained in this work.

Overall, the results consistently indicate that fuzzy aggregation via the generalized Choquet integral is most beneficial when the feature space is small and interactions among variables are not yet well represented by raw attributes. From a data perspective, the pipeline begins with a compact and reliable basis: the CIC-IDS2017 dataset (692,703 rows, 79 columns) is cleaned and reduced through SelectKBest to a controlled search space of up to $k = 10$ features, an upper bound chosen because the accuracy curves stabilize after 6–7 features for both models (Figures 6 and 7). This design avoids injecting superfluous information while still exposing the performance trajectory as k increases.

Methodologically, the comparison is fair: both pipelines (with and without the Choquet-based aggregated feature) share identical splits, training settings, and evaluation protocol.

Empirically, Figures 6 and 7 and Tables 10 and 11 reveal a consistent pattern: with few features ($k! \in![1, 5]$), the Choquet-augmented pipeline outperforms the baseline, as k increases, the gap narrows, and both models converge for $k! \geq!$6–7. The largest improvement occurs at $k = 4$, with $\Delta Y \approx 0.07$, indicating that fuzzy aggregation enables the classifiers to achieve high accuracy using fewer attributes. The confusion matrices (Figures 8 and 9, shown for the best-k configuration) reinforce this result by highlighting a practical trade-off: the baseline favors maximum attack detection but produces more false positives on benign traffic, whereas the Choquet-based model substantially reduces false alarms (higher true-negative rate) at a modest cost in missed attacks.

In short, the generalized Choquet aggregation delivers better accuracy in low-dimensional data, helping reducing big data. Once the feature set surpasses $k \approx$ 6–7, the benefits decreasing as both pipelines approach their ceiling. These findings support the use of Choquet-based fusion as a compact, interaction-aware representation that accelerates performance while controlling dimensionality.

Beyond the quantitative gains, the results highlight practical implications for large-scale network monitoring. The proposed Choquet-based aggregation offers a balanced trade-off between feature expressiveness and computational cost, making it suitable for deployment in data centers or backbone networks where massive traffic must be analyzed under strict latency constraints. Although the experiments were performed on offline batches of the CIC-IDS2017 dataset, the methodology remains compatible with streaming frameworks, as the aggregation can be computed incrementally as new features arrive. This property indicates that the approach can be integrated into big data analytics pipelines, enabling real-time anomaly detection without compromising interpretability or scalability.

Algorithm 4 Feature Selection, Choquet and Incremental Model Evaluation

Input: Dataset D

Output: Performance metrics for baseline and Choquet pipelines from $k = 1$ to N

$D_{clean} \leftarrow$ Preprocess(D) ▷ Data cleaning and preprocessing

$selectedFeatures \leftarrow$ KBest(D_{clean}, $k = N$) ▷ Select top N original features

$choquetFeatures \leftarrow [\,]$

for $j = 1$ to N **do**

    $c_j \leftarrow$ ChoquetIntegral($selectedFeatures[j]$) ▷ Compute Choquet-based feature for the j-th selected attribute

    Append c_j to $choquetFeatures$

end for

$Results \leftarrow [\,]$

for $k = 1$ to N **do**

    $F_k^{base} \leftarrow selectedFeatures[1:k]$ ▷ First k original features

    $F_k^{choq} \leftarrow choquetFeatures[1:k]$ ▷ First k Choquet-transformed features

    $D_k^{base} \leftarrow$ PrepareData(D_{clean}, F_k^{base}) ▷ Dataset with k raw features

    $D_k^{choq} \leftarrow$ PrepareData(D_{clean}, F_k^{choq}) ▷ Dataset with k Choquet features

    $RF_k^{base} \leftarrow$ Train(RandomForest, D_k^{base})

    $XGB_k^{base} \leftarrow$ Train(XGBoost, D_k^{base}) ▷ Train baseline models

    $RF_k^{choq} \leftarrow$ Train(RandomForest, D_k^{choq})

    $XGB_k^{choq} \leftarrow$ Train(XGBoost, D_k^{choq}) ▷ Train Choquet-augmented models

    $resRF_k^{base} \leftarrow$ Evaluate(RF_k^{base}, D_k^{base})

    $resXGB_k^{base} \leftarrow$ Evaluate(XGB_k^{base}, D_k^{base})

    $resRF_k^{choq} \leftarrow$ Evaluate(RF_k^{choq}, D_k^{choq})

    $resXGB_k^{choq} \leftarrow$ Evaluate(XGB_k^{choq}, D_k^{choq})

    Store $(k, resRF_k^{base}, resRF_k^{choq}, resXGB_k^{base}, resXGB_k^{choq})$ in $Results$ ▷ Keep metrics for later Δ analysis

end for

return $Results$ ▷ Metrics for baseline and Choquet pipelines for all k

6 FINAL CONSIDERATIONS

In this chapter, the main findings of the work were summarized and the potential directions for future research were outlined.

6.1 Summary of the work

Based on the proposal for adapting the α parameter from a previous study [35], and aligned with this work's main objective of enhancing a feature to improve network anomaly detection [36], the following considerations can be made.

Using generalized Choquet integral by aggregating feature relevance through fuzzy measures, the method effectively captures nonlinear dependencies among attributes and mitigates redundancy in high-dimensional traffic datasets. Experiments conducted on the CIC-IDS2017 dataset demonstrated that the Choquet aggregation can outperform baseline feature selection methods in scenarios with limited feature subsets, achieving up to 7% accuracy improvement at $k = 4$ and a 77.5% reduction in data volume.

Furthermore, the results suggest a clear trade-off: the Choquet model reduces false positives while slightly increasing false negatives, which can be balanced by threshold calibration. When evaluated over repeated stratified experiments, the performance p gains were statistically significant ($p < 0.05$) for both Random Forest and XGBoost classifiers.

6.2 Future work

Looking ahead, the team plans to apply the Choquet Integral to different datasets and assess how well it performs in detecting various types of DDoS attacks. Furthermore, other optimization strategies for the α variable could be tested, aiming to further reduce the mean absolute error (MAE) during feature generation.

Another line of research includes applying this technique to deep learning models and evaluating the impact of the new feature on more complex architectures, such as convolutional neural networks to detect more subtle and temporal patterns in attacks.

REFERENCES

[1] Aldhyani, T. H., Alrasheedi, M., Alqarni, A. A., Alzahrani, M. Y., and Bamhdi, A. M. (2020). Intelligent hybrid model to enhance time series models for predicting network traffic. *IEEE Access*, 8:130431–130451.

[2] Alsina, C., Schweizer, B., and Frank, M. J. (2006). *Associative functions: triangular norms and copulas*. World Scientific.

[3] Amorim, M., Dimuro, G., Borges, E., Dalmazo, B. L., Marco-Detchart, C., Lucca, G., and Bustince, H. (2023). Systematic review of aggregation functions applied to image edge detection. *Axioms*, 12(4).

[4] Ayres, D., Quevedo, A., Dimuro, G., Lucca, G., and Dalmazo, B. (2024a). Detecção de anomalias de rede utilizando integrais de choquet através de medidas de potência. In *Anais da XXI Escola Regional de Redes de Computadores*, pages 129–134, Porto Alegre, RS, Brasil. SBC.

[5] Ayres, D., Quevedo, A., Lucca, G., Dimuro, G., and Dalmazo, B. (2024b). Comparando médias móveis com integral de choquet para detectar anomalias no tráfego de redes. In *Anais Estendidos do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 353–357, Porto Alegre, RS, Brasil. SBC.

[6] Aziz, W. A., Qureshi, H. K., Iqbal, A., Al-Dulaimi, A., and Al–Rubaye, S. (2023). Towards accurate categorization of network ip traffic using deep packet inspection and machine learning. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 01–06.

[7] Beliakov, G., Pradera, A., Calvo, T., et al. (2007). *Aggregation functions: A guide for practitioners*, volume 221. Springer.

[8] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[9] Cardoso, F. C., Berri, R. A., Borges, E. N., Dalmazo, B. L., Lucca, G., and de Mattos, V. L. D. (2024). Echo state network and classical statistical techniques for time series forecasting: A review. *Knowledge-Based Systems*, 293:111639.

[10] Carpenter, J., Layne, J., Serra, E., Cuzzocrea, A., and Gallo, C. (2023). Structural node representation learning for detecting botnet nodes. In Gervasi, O., Murgante, B., Taniar, D., Apduhan, B. O., Braga, A. C., Garau, C., and Stratigea, A., editors, *Computational Science and Its Applications – ICCSA 2023*, pages 731–743. Springer Nature Switzerland.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

[12] Chen, L., Gao, S., Liu, B., Lu, Z., and Jiang, Z. (2020). Few-nnn: A fuzzy entropy weighted natural nearest neighbor method for flow-based network traffic attack detection. *China Communications*, 17(5):151–167.

[13] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

[14] Choquet, G. (1953). 1954. *Theory of Capacities." Annales de l'Institut Fourier5*, pages 131–295.

[15] Cloudflare (2024). Relatório de ameaças ddos: 4º trimestre de 2024. `https://blog.cloudflare.com/pt-br/ddos-threat-report-for-2024-q4/`. Acesso em: 2 set. 2025.

[16] Dalmazo, B., Vilela, J., and Curado, M. (2017). Performance analysis of network traffic predictors in the cloud. *Journal of Network and Systems Management*, 25.

[17] Dalmazo, B. L. et al. (2021). A systematic review on distributed denial of service attack defense mechanisms in programmable networks. *International Journal of Network Management*, 31(6):e2163.

[18] Dalmazo, B. L., Vilela, J. P., and Curado, M. (2018). Triple-Similarity Mechanism for Alarm Management in the Cloud. *Computers & Security - Elsevier*, 78:33–42.

[19] Fadilpašić, S. (2025). Cloudflare says it has once again blocked the largest-ever DDoS attack in history. Accessed: 22 December 2025.

[20] Hu, Y. and Tu, B. (2024). Security situation assessment model of ddos attack based on progressive fuzzy c clustering algorithm. In *2024 International Conference on Data Science and Network Security (ICDSNS)*, pages 1–4.

[21] Jiang, H., He, Z., Ye, G., and Zhang, H. (2020). Network intrusion detection based on pso-xgboost model. *IEEE Access*, 8:58392–58401.

[22] Karczmarek, P., Gałka, , Kiersztyn, A., Dolecki, M., Kiersztyn, K., and Pedrycz, W. (2023). Choquet integral-based aggregation for the analysis of anomalies occurrence in sustainable transportation systems. *IEEE Transactions on Fuzzy Systems*, 31(2):536–546.

[23] Li, J., Ma, J., Omisore, O. M., Liu, Y., Tang, H., Ao, P., Yan, Y., Wang, L., and Nie, Z. (2024). Noninvasive blood glucose monitoring using spatiotemporal ecg and ppg feature fusion and weight-based choquet integral multimodel approach. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):14491–14505.

[24] Li, W. and Moore, A. W. (2007). A machine learning approach for efficient traffic classification. In *2007 15th International symposium on modeling, analysis, and simulation of computer and telecommunication systems*, pages 310–317. IEEE.

[25] Lin, A. (2019). Binary search algorithm. *WikiJournal of Science*, 2(1):1–13.

[26] Lucca, G., Dimuro, G. P., Bedregal, B., Sanz, J. A., and Bustince, H. (2016). A proposal for tuning the alpha parameter in a copula function applied in fuzzy rule-based classification systems. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 367–372.

[27] Lucca, G., Sanz, J. A., Dimuro, G. P., Bedregal, B., and Bustince, H. (2020). A proposal for tuning the α parameter in $c_\alpha c$-integrals for application in fuzzy rule-based classification systems. *Natural Computing*, 19(3):533–546.

[28] Mayor, G. and Trillas, E. (1986). On the representation of some aggregation functions.

[29] Menger, K. (1942). Statistical metrics. *Proceedings of the National Academy of Sciences*, 28(12):535–537.

[30] Murofushi, T., Sugeno, M., and Machida, M. (1994). Non-monotonic fuzzy measures and the choquet integral. *Fuzzy sets and Systems*, 64(1):73–86.

[31] Nelson, J. (2025). X outage linked to dark storm hacker group as elon musk confirms 'massive cyberattack'. Accessed: 2025-09-10.

[32] Novaes, M. P., Carvalho, L. F., Lloret, J., and Proença, M. L. (2020). Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access*, 8:83765–83781.

[33] Otchere, D. A., Ganat, T. O. A., Ojero, J. O., Tackie-Otoo, B. N., and Taki, M. Y. (2022). Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions. *Journal of Petroleum Science and Engineering*, 208:109244.

[34] Quevedo, A., Ayres, D., Dimuro, G., Lucca, G., Riker, A., and Dalmazo, B. (2024). O parâmetro na generalização da integral de choquet para previsão de tráfego de rede. In *Anais da XXI Escola Regional de Redes de Computadores*, pages 30–34, Porto Alegre, RS, Brasil. SBC.

[35] Quevedo, A., Ayres, D., Dimuro, G., Riker, A., Lucca, G., and Dalmazo, B. L. (2025a). Optimizing big data traffic prediction using generalizations of choquet integral with adaptive weighting. In *ICC 2025 - IEEE International Conference on Communications*, pages 4872–4877.

[36] Quevedo, A., Ayres, D., Teixeira, G., Dimuro, G., Lucca, G., and Dalmazo, B. L. (2025b). Improving anomaly detection in network traffic using choquet-based feature engineering for random forest and xgboost models. In Gervasi, O., Murgante, B., Garau, C., Karaca, Y., Taniar, D., C. Rocha, A. M. A., and Apduhan, B. O., editors, *Computational Science and Its Applications – ICCSA 2025*, pages 3–16, Cham. Springer Nature Switzerland.

[37] Schweizer, B. and Sklar, A. (2011). *Probabilistic Metric Spaces*. Dover Publications.

[38] Shetty, S., S, T. M., M, V. H., and Shaikh, R. N. (2024). Intelligent network traffic control with ai and machine learning. In *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 353–357.

[39] Wang, J., Zhao, H., Xu, J., Li, H., Zhu, H., Chao, S., and Zheng, C. (2018). Using intuitionistic fuzzy set for anomaly detection of network traffic from flow interaction. *IEEE Access*, 6:64801–64816.

[40] Wang, Y.-N., Wang, J., Fan, X., and Song, Y. (2020). Network traffic anomaly detection algorithm based on intuitionistic fuzzy time series graph mining. *IEEE Access*, 8:63381–63389.

[41] Wani, A. R., Rana, Q. P., Saxena, U., and Pandey, N. (2019). Analysis and detection of ddos attacks on cloud computing environment using machine learning techniques. In *Proc. 2019 Amity Int. Conf. on Artificial Intelligence (AICAI)*, pages 870–875.