UNIVERSITY OF AEGEAN

FACULTY OF SCIENCE

DEPARTMENT OF STATISTICS AND

ACTUARIAL - FINANCIAL MATHEMATICS

«A Deep Learning Classifier Selection Method for Credit Scoring using Neuro-Fuzzy Approaches»

Master Thesis for the Postgraduate

Programme of Studies

This thesis was prepared in partial fulfillment of the

requirements for the corresponding degree in

Statistics and Actuarial - Financial Mathematics

KOVOUSOGLOU K. GEORGIOS

JUNE 2023

SAMOS

KOVOUSOGLOU K. GEORGIOS

«A Deep Learning Classifier Selection Method for Credit Scoring using Neuro-Fuzzy Approaches»

JUNE 2023

Master Thesis for the Postgraduate Programme of Studies

Department Of Statistics And Actuarial - Financial Mathematics

Author: KOVOUSOGLOU K. Georgios

Supervisor: LAPPAS Pantelis, Assistant Professor

Committee Member: XANTHOPOULOS Stylianos, Associate Professor

Committee Member: TACHTSIS Eleftherios, Professor

SAMOS

DECLARATION OF NON-PLAGIARISM

This thesis is submitted by the postgraduate student Georgios K. Kovousoglou as partial fulfillment of the requirements of the Master of Science in Statistics and Actuarial Finance of the University of Aegean. It is duly declared that this thesis has been written by the undersigned and has not been submitted or evaluated in the framework of any other postgraduate programme or undergraduate degree, in Greece or abroad.

Georgios K. Kovousoglou

Signature:

Acknowledgements

To my family and beloved ones.

THANKS

With the completion of this thesis, I would like to thank my supervisor, Assistant Professor Mr. Lappa Panteli for his help in the preparation of this thesis. His accurate guidance and immediate response to my questions were valuable contributors to my writing of this thesis.

Abstract

Credit risk is a prominent facet of financial risk that holds paramount significance in determining the prosperity of financial institutions. As such, extensive research efforts have been dedicated to the development of various credit scoring theories and models. This thesis proposes a novel Neuro-Fuzzy model for effectively classifying the creditworthiness of clients. The research methodology encompasses two independent phases. In the initial phase, an Artificial Neural Network (ANN) is employed to generate evaluation metric scores for three distinct ANN topologies, namely those with 1, 2, and 3 hidden layers. The ANN's performance is assessed using four different tuning hyperparameter techniques. Concurrently, in the second phase, a comprehensive evaluation metric hierarchy is constructed through the integration of experts' opinions, utilizing the Fuzzy Analytic Hierarchy Process (AHP) model. By amalgamating the outcomes of both phases, a Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) model is employed to determine the optimal neural network topology for the creditworthiness classification task. The distinctiveness of this proposed method lies in its incorporation of financial experts' insights and alignment with the institution's policy, within the framework of a data-driven model that avoids reliance on output reasoning. The overarching objective is to harness the remarkable potential exhibited by Neural Networks in credit risk assessment. To evaluate the efficacy of the proposed model, it is applied to three distinct datasets, and the resulting outcomes are meticulously analyzed and discussed, thereby unveiling its practical utility. The conclusions drawn from this research underscore the indispensability of human expertise, as facilitated by the Fuzzy AHP approach, particularly in situations where the ANN models yield comparable performance metric outputs. Consequently, this approach emerges as a valuable validation mechanism for decisions grounded in ANN outputs.

keywords: Neural Networks, Fuzzy AHP, TOPSIS, Credit Scoring

Contents

List of Figures

List of Tables

Introduction

Due to significant costs associated with inaccurate judgments, credit risk assessment decisions are a critical factor in determining the performance of financial institutions in the lending business. The US subprime mortgage crisis serves as a demonstration of how credit risk judgments could have a profound impact on both local and global economies. Owing to consumer payment failures, several financial institutions have experienced considerable losses. The development of credit risk decision support tools and models plays a crucial role in enhancing evaluation judgments by enabling faster and more accurate conclusions, ultimately leading to improved decision-making processes.

The most popular method for determining whether credit applicants are creditworthy based on factors including their age, income, and marital status is credit scoring (Chen and Huang, 2003). Its purpose is to divide credit applicants into two groups based on their propensity to make payments: trustworthy applicants are the ones that are more likely to make their payments on time and subsequently acquire credit, whereas non-trustworthy applicants are the ones rejected due to a high risk of missing payments. Financial institutions have exploited a variety of techniques, with statistical techniques being the most often used, to create precise credit scoring models.

In the present day, with the continuous growth and complexity of the financial market, there is an increasing demand for advanced modeling techniques that could effectively simulate complex, non-linear Real-world applications. This necessity arises from the need to accurately capture and analyze the intricate dynamics and interdependencies within financial systems, enabling better understanding and management of risks in a rapidly evolving market landscape. In this situation, soft computing techniques have been effectively used to address non-linear issues in business, engineering, and medicine. These methods could be used as alternatives to statistical methods to find approximations of solutions to Real-world problems that involve a variety of errors and uncertainties.

Neuro-Fuzzy (NF) algorithms are one of the most innovative solutions in credit scoring for the reason that they combine the tremendous learning force of Neural Networks (NN) with expert knowledge via fuzzy reasoning. Recent studies have focused on employing Neuro-Fuzzy (NF) approaches to enhance credit scoring models. In Akkoç's study (2019), an Adaptive Neuro-fuzzy Inference System (ANFIS) had better accuracy than other statistical methods in credit scoring. In Pabuçcu and Ayan (2016) the ANFIS model made consistent, reliable, and successful credit rating forecasts compared with others statistical methods. Pasila (2019) described how to effectively predict the credit score of Indonesian Micro, Small, and Medium Enterprises (MSMEs) using a Takagi-Sugeno (TS) type multi-input single-output (MISO) neuro-fuzzy network. The application of a Neuro-Fuzzy with an extra accelerated Levenberg-Marquardt algorithm was also

mentioned. According to Tezerjan et al. (2021), a hybrid model based on adaptive neuro-fuzzy inference systems (ANFIS) and Recurrent Neural Networks (RNN), using historical data and indicators, identifies and forecasts the shocks of various stock market segments. By inputting the data together with other client criteria into a fuzzy rule base (FRB), the customer score was finally determined.

This study investigates three research questions:

Q1: How to solve credit scoring problems using efficient machine learning algorithms?

Q2: How to address computational challenges when single or multiple machine learning algorithms are considered?

Q3: How to select the most appropriate classifier for solving credit scoring problems?

The first question focuses on searching various solving credit scoring problems using efficient machine learning algorithms. In other words, traditional statistical approaches, such as linear or logistic regression, could be characterized as high bias and low variance approaches. As a result, they are unable to capture relevant relations between dependent and independent variables (underfitting), but they provide robust predictions regardless of the sample. On the other hand, computational intelligence approaches, such as neural networks, present low bias and high variance. As a result, they might capture complicated patterns, but their results might be difficult to be generalized, due to their sensitivity to training data (overfitting). Therefore, neural networks have been selected to be applied in credit scoring in this study, considering the importance of capturing detailed patterns as well as striking a balance between variance and bias (Neal, 2019).

The second query concerns the computational difficulties of using a single or multiple machine learning algorithm. The main drawback of neural networks is their computational expensiveness. To address this issue, various tuning methods are employed, including optimization algorithms such as Genetic Algorithms, Monte Carlo simulation, and Bayesian Inference.

The third question pertains to techniques for selecting the most appropriate classifier to solve the credit scoring problem. Initially, the hyperparameters of the selected neural network topologies are optimized. In continuance, multi-criteria decision analysis with fuzzy logic is utilized to incorporate expert knowledge. Finally, the classifier selection process is completed by weighing multiple performance measures.

Innovation of this thesis is associated with the usage of optimization techniques and Deep Learning in hybrid synthesis, as well as Multi-Criteria Decision Analysis concepts for classifier selection. More specifically, a novel solution of Neuro-Fuzzy in credit scoring was proposed by using optimization methods on three NN different topologies, as well as Fuzzy Analytic Hierarchy Process (FAHP) and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) methods for optimal model selection. The novel model contribution in the scientific community focused on the fact that evaluation metrics results of NN and decision maker view about metric importance would be able to be embedded, producing an optimal suggestion for model selection as

output. In this manner, employees of financial institutes would acquire a hybrid model selection method, and as a consequence institute's profitability and sustainability would be improved.

This thesis consists of seven chapters. Chapter 2 provides a comprehensive definition of risk management sources and emphasizes the significance of credit scoring. Chapters 3, 4, and 5 focus on elaborating the key terminology and concepts related to Artificial Neural Networks, Fuzzy Logic with Multi-criteria Decision Analysis, and the proposed Neuro-Fuzzy approach, respectively. In Chapter 6, the conducted computational experiments and the obtained -from implementing the Neuro-Fuzzy model in three datasets- results are presented. Lastly, Chapter 7 summarizes the conclusions drawn from the experiments, discusses the limitations of this research, and suggests potential perspectives for future extensions.

Chapter 2:
Risk Management

2.1. Introduction

From antiquity to the present day, there have been many different definitions of risk depending on the century and cultural changes. Some definitions emphasize the likelihood of an event, which contains the uncertainty of positive or negative consequences. Still, certain definitions distinguish risk as a subset of uncertainty that could be quantified or measured.

The definition of risk in the field of finance is the unpredictability of investment returns, encompassing both positive and negative impacts. According to this perspective, a higher expected return is related to a higher level of result volatility. In an effort to provide a comprehensive understanding of risk, it is formed a definition by the ISO 31000 Risk Management framework, an international standard from the International Organization for Standardization that offers businesses concepts and recommendations for risk management. Hence, risk management is referred to as, 'Coordinated activities to direct and control an organization concerning risk…' (Te, 2016).

In the financial system, the uncertainty that has a negative impact on wealth or profits, or the uncertainty that is solely related to adverse outcomes, is a risk. For a company its survival and growth are vital and the only way to achieve this is to take risks. Understanding the portfolio of risks that the corporation is presently taking and the expected risks is the primary responsibility of the risk management function. In certain, it must assess if the risks are tolerable and the steps should be followed in the case that they are not (Hull, 2018).

Risk managers and regulators share this perspective. Regulations are an essential component for a risk manager since they are designed to increase the financial system's and financial firm's resilience under pressure. Risk managers regard themselves as responsible for determining, evaluating, and controlling the possibility and effects of unfavorable occurrences for the company. According to this perspective, risk is defined as the likelihood of suffering a loss as a result of an interaction with uncertainty. Due to the financial institutions' vulnerability to such unpredictability, there is interaction. Exposure is the degree to which a firm could be harmed by specific events that could negatively impact earnings. For instance, the magnitude of foreign currency income is provided as a measure of exposure to foreign exchange rates, and the amount of debt

that is priced according to market rates operates as a measure of exposure to interest rates.

While it is possible to mitigate the impact of uncertainties, the inherent nature of uncertainty itself cannot be eliminated. For example, with an aim to reduce the impact of variations in foreign exchange rates on earnings, a company with foreign currency revenues receives a loan using the same foreign currency. A company that lends at a floating rate could use borrowing at a floating rate to lessen volatility in net interest income, interest revenue excluding interest costs, and other related factors.

Even if it is acceptable to hedge risks with cash instruments, derivatives are most frequently used. Derivative instruments acquire their value from other underlying assets. A contract that specifies the potential use of exchange rate in converting external income into local currency might be signed by the firm indicated above that is willing to minimize its lengthy exposure to foreign exchange. This approach is generally more straightforward compared to the endeavor of borrowing funds in a different currency. Derivatives offer a wide range of applications due to their versatility and adaptability (Bessis, 2015).

This chapter aims to provide an overview of the primary sources of financial risk, with particular emphasis on credit risk. Credit scoring is identified as the most widely used method for mitigating credit risk, and its utility is discussed in detail.

2.2. Sources of financial risks

Financial risks arise from different sources of uncertainty and could be categorized into several main categories: credit risk, market risk, liquidity risk, interest rate risk, foreign exchange risk, solvency risk, and operational risk. Each of these categories represents a distinct type of risk that financial institutions face. The following paragraphs provide an overview of each category.

Credit risk

In this assignment, a soft computing model is developed with the upper aim to eliminate credit risk. The hazard of losses resulting from borrowers' defaults or a decline in credit standing is known as credit risk. The danger that debtors would not execute their financial commitments is known as default risk. Losses of the amount lent to the counterparty in the event of default might be whole or partial.

Credit risk also refers to a borrower's credit standing being worse, which increases the possibility of default even if it does not always indicate default. The asset value of a mortgage does not change as the creditworthiness of the borrower drops, but its economic worth is lower because the chance of default increases. An unfavorable migration causes the stated price of a traded debt to decrease (Bessis, 2015).

Market risk

Market risk is the potential for financial loss due to unfavorable market movements that lower the value of market participants' positions. Any market variable that fluctuates unpredictably qualifies as a risk factor; examples include interest rates, stock indexes, and exchange rates. Market risk is influenced by how long it takes to sell the assets since bigger market movements often happen over longer time periods. The liquidation period is longer for goods that are difficult to trade in busy markets and shorter for exotic instruments that are traded bilaterally (over the counter). Market risk is the name given to price risk for traded instruments. Instruments that are not sold on regulated markets are marked to market because profits and losses, regardless of whether a transaction occurs or not, are reported as changes in value (Bessis, 2015).

Liquidity risk

The risk of being unable to raise money when needed is known as liquidity risk. Banking companies could raise money by taking on debt or by peddling financial assets on the open market. Borrowing to raise money is referred to as funding liquidity. When borrowers are not able to borrow money or are unable to do so under ordinary circumstances, funding liquidity risk manifests. Asset liquidity is the term used to describe the money obtained from the selling of assets on the market as a backup source of funding, for instance during market interruptions. When the market is unable to absorb the transactions at the present price, asset liquidity also describes the possibility that prices might shift against the seller or buyer as a consequence of its trades. When there are too many participants making similar deals, asset liquidity risk also develops. For instance, under the challenging circumstances of the 2008 crisis, banks seeking to raise capital through the sale of assets suffered significant losses as a result of the steep discounts in their deals (Bessis, 2015).

Interest rate risk

Interest rate risk is the threat of decreased profitability, or interest earnings minus interest expenses, in reaction to fluctuations in interest rates. The majority of loans, receivables, and term or saving deposits on a bank's balance sheet generate revenue and expenditures that are impacted by interest rates. The danger of interest rates affects all lenders and borrowers. Variable rate interest costs and income for lenders and borrowers are based on short-term market rates. Loans and obligations with fixed rates are similarly subject to interest rate risk. When interest rates fall, fixed-rate borrowers might benefit from lower interest rates, but fixed-rate lenders might still be able to lend at a higher rate if rates rise. Both are subject to interest rate variations due to their opportunity costs brought on by changes in the market (Bessis, 2015).

Foreign exchange risk

The threat of suffering losses as a result of exchange rate fluctuations is known as foreign exchange risk. Changes in the values of assets and foreign currency-denominated liabilities or the indexation of revenues and fees to exchange rates are the two factors that cause fluctuations in earnings (Bessis, 2015).

Solvency risk

Solvency risk is the chance that there won't be enough money on hand to pay for losses. A minimal capital base is required, according to the "capital adequacy" concept, which is backed by regulators, to resist unforeseeable losses that might be caused by the firm's current risks. When unforeseen losses exceed the capital level, as they did for many firms during the 2008 financial crisis, solvency issues arise. The likelihood that the bank would incur a default and that potential losses would exceed its capital base is determined by this capital buffer (Bessis, 2015).

Operational risk

Operational risks include those posed by the breakdown of the information system, reporting systems, internal risk monitoring policies, and procedures designed to swiftly take corrective actions. "The risk of direct or indirect loss due to inefficient or insufficient methods, people, and systems, or due to external occurrences," is how the regulators define operational risk. Operational risk became more important as regulators required that operational risks be charged against capital (Bessis, 2015).

The effective understanding and management of different categories of financial risks play a crucial role in enhancing the resilience of institutions and enabling them to make informed decisions that mitigate potential adverse effects. Within this context, this study specifically focuses on the development of methodologies aimed at reducing the impacts of credit risk within financial institutions. By addressing this specific area of risk, the study aims to contribute to the overall risk management framework of financial institutions, ultimately promoting stability and sustainable growth in the financial sector.

2.3. Credit risk

The term "credit risk" could be defined as the possibility of an unexpected alteration in the creditworthiness of a counterparty resulting in an unforeseen modification in the market value of the associated credit exposure (Resti & Sironi, 2007). This definition encompasses three key concepts that are integral to understanding credit risk.

Default and Migration risk

The first indicates the risk of loss due to the borrower's real bankruptcy (when payments are terminated), whereas the second indicates the risk of loss because of a just deterioration of its credit rating.

Generally, the value of a credit exposure would be reduced more dramatically as a result of a decline in the borrower's credit rating the more spread variation there is and the longer the remaining life of the loan. For instance, given a fixed-interest loan, the market value of the loan, which is based on the present value of the related cash flows, would undoubtedly decrease as the borrower's creditworthiness declines. The rationale seems to be that the present value of future flows should be calculated using a discount rate that includes not only the risk-free rate for the corresponding maturity

but also a spread (risk premium) that accounts for the possibility of the borrower defaulting. A decline in creditworthiness raises that chance, which inevitably causes the spread to widen and the present value to decrease. This could also apply to a variable-rate loan when the borrower's spread above the market rate is predetermined and immutable (Resti & Sironi, 2007).

Risk of Unexpected Event

The actual risk is reflected by the potential for such assessments to turn out to be inaccurate, i.e., by a counterparty decline that the lender was not prepared for. In this sense, proper risk solely refers to unforeseen events that were predicted. The variation in the counterparty's credit rating should be unexpected. In other words, if a bank has provided a loan in anticipation of the counterparty's future deterioration in quality, such deterioration would have been appropriately assessed and taken into account throughout the loan decision-making process and the definition of the interest rate. In actuality, fair attention has been always given to anticipated changes in the borrower's economic and financial situation when determining the likelihood of default and the related interest rate (Resti & Sironi, 2007).

Credit Exposure

The concept of credit exposure indicates that credit risk includes off-balance-sheet activities such as guarantees, derivative contracts traded over the counter, and transactions in securities, foreign currencies, or derivatives pending final settlement, and is not at all restricted to the "classic" forms of credit awarded by a bank (Te, 2016).

Finally, the market value of credit risks mentioned in the above definition raises two issues. The first one is concerned with a number of credit exposures that are represented at historical value rather than market value in the books of financial institutions. The price that an independent buyer would assign to the exposure when it was acquired by the bank would serve as the basis for appraisals in order to accurately quantify credit risk and its implications. The second issue mentions that the majority of a financial institution's credit exposures are made up of illiquid assets for which there is no established secondary market; as a result, the market value could only be approximated using an internal asset-pricing model.

All in all, credit risk assessment could be effectively carried out through the utilization of credit scoring techniques. These techniques enable the evaluation of borrower's creditworthiness and the prediction of potential defaults or credit losses. By employing robust credit scoring models, financial institutions could make informed decisions regarding lending and credit management, mitigating the risks associated with loan default and credit-related losses. Credit scoring serves as a valuable tool in the assessment of credit risk, providing a systematic and objective approach to evaluating the creditworthiness of borrowers and supporting sound credit decision-making processes.

2.4. Credit Scoring

Credit risk and default probability models have evolved over the years, from the earliest statistical models that linked ratings or default frequencies to the financial features of businesses to more complex econometric methods and neural network models.

The concept behind scoring is to adopt a measure to categorize "excellent" and "poor" credit scores into separate groups based on the borrowers' common characteristics. The method uses historical and present values of observable variables for corporations, including financial ratios, profitability, leverage, and size. Income, age, and professional activity are a few factors that affect a person's credit status.

Such models must have their defaults properly stated in order to be used. This is not a simple task in retail banking. For instance, when members of the same family are involved, the default of one member of the group might result in the default of the entire group. Additionally, a single person frequently uses a variety of bank products, including loans and savings accounts. One option for paying a debt would be to use an additional overdraft from the bank account. Although it appears that the loan does not default, the account does. In these circumstances, the default on one product affects the borrower's other items as well. Contagion rules across groups of people and between goods are necessary for accurately characterizing a default in retail banking. These rules have a direct impact on the statistics of default events (Bessis, 2015).

All phases of the credit life cycle involve the use of credit scores. Following are some examples:

- When deciding whether to accept or refuse a loan request and how much it would cost, the credit score takes into consideration application scores based on a borrower's application data.
- Collection ratings that, depending on a variety of factors, including the borrower's previous performance, indicate the possibility of the loan or borrower slipping deeper into arrears.
- Behavioral ratings are based on data that is already known about a borrower's prior actions in various stages of the credit life cycle.
- Early warning ratings that alert the credit score to a situation that might have an impact on the borrower's credit risk.
- By verifying data and behavior, fraud detection ratings let credit scores be alerted to potentially fraudulent activity (Crouhy et al., 2005).

Predictive modeling techniques are employed to assess the credit risk associated with prospective borrowers. These models could be categorized into two main types: parametric and non-parametric, while the learning process could be either supervised or unsupervised.

Parametric models, such as Generalized Linear Models (GLMs), including additive scorecards, fall under the parametric approach. These models utilize predetermined mathematical functions and assume specific distributions for the variables involved.

They are widely used in credit scoring and could provide interpretable and transparent results.

On the other hand, non-parametric predictive models employ more complex black-box techniques. These models are capable of handling non-linear correlations, predictor interactions, unstable or poorly understood data, and pattern recognition challenges. They are particularly useful when dealing with intricate relationships within the data. Non-parametric models offer greater flexibility but might be less interpretable compared to parametric models.

The learning process for these models could be supervised or unsupervised. In supervised learning, the models are trained using labeled data, where the outcome variable (e.g., default or non-default) is known. Unsupervised learning, on the other hand, involves analyzing unlabeled data to identify patterns or groupings without prior knowledge of the outcome. Table 2.1 in the referenced source (Finlay, 2012) presents commonly used predictive models for credit scoring, providing an overview of the different techniques employed in this field. It serves as a reference to understand the variety of models available and their suitability for credit risk assessment.

Traditional statistical approaches	Computational Intelligence approaches
Linear Regression	Ensemble model
Linear Discriminant Analysis	Neural Networks
Logistic Regression	Genetic Algorithms

Table 2.1. Predictive models for credit scoring.

In conclusion, the evaluation and management of credit risk play a pivotal role in the financial industry, allowing institutions to effectively assess and mitigate potential risks associated with lending activities. Throughout this chapter have been referred a range of financial risk resources, credit risk components, and models utilized in credit risk assessment, encompassing statistical approaches as well as the emerging field of neural networks.

However, in the transition to the subsequent chapter, the focus would shift toward the domain of neural networks and their applicability in credit risk analysis. Consequently, in the upcoming chapter, the concepts and principles underlying Neural Networks would be comprehensively described to elucidate their significant contribution to this thesis. By providing a clear understanding of Neural Networks, their potential applications and advantages in credit scoring would be highlighted, setting the stage for their subsequent implementation and evaluation in the research.

Neural networks offer the promise of enhancing credit risk models by capturing intricate patterns and nonlinear relationships within data, thereby facilitating more precise and robust risk assessments. By integrating neural networks into credit risk

frameworks, financial institutions could harness advanced computational capabilities to more effectively identify and mitigate credit risks. The exploration of neural networks within the context of credit risk presents novel avenues for advancing risk management practices and augmenting decision-making processes within the financial sector.

Chapter 3:
Neural Networks

3.1. Introduction

In the early years of humanity, there were plenty of machine inventors who had created machines with the aim to assist them with daily tasks. Before the creation of the first computer with software, there was a lot of discussion on the ability of such machines to become ever sentient (Lovelace, 1842). Artificial intelligence (AI) is a modern scientific discipline with a number of beneficial applications and ongoing research topics. Intelligent software is employed to help basic scientific research, automate monotonous processes, understand speech or visual input, and make medical diagnoses.

It turns out that the true challenge for artificial intelligence is to solve problems that people instinctively and apparently easily solve, such as recognizing spoken words or faces in images, but which humans find difficult to define in formal terms. In this situation, allowing computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each thought defined in relation to simpler notions, would be a solution to this issue. By using experience-based learning, this approach lessens the need for human operators to explicitly specify every piece of information that the computer needs. Computers could learn more complicated concepts by using fundamental ideas as a foundation (Heaton, 2018).

The limitations encountered by hard-coded knowledge-based systems have highlighted the necessity for artificial intelligence (AI) systems to possess the ability to learn autonomously by identifying patterns in unstructured data. This capability is known as machine learning, and it has revolutionized the way computers tackle Real-world problems and arrive at seemingly subjective conclusions. For instance, logistic regression, a straightforward machine learning technique, could determine whether a cesarean birth is recommended (Mor-Yosef et al., 1990), while financial institutions could employ the naive Bayes algorithm, another simple machine learning technique, to assess the likelihood of clients becoming defaulters before granting loans (Heaton, 2018).

In recent decades, extensive research has been conducted by scientists in applied sciences on neural networks, owing to their ability to mimic the distributed communication and information processing nodes found in biological systems. Artificial neural networks (ANNs) were developed as a result of this exploration. It is important to note that while ANNs are inspired by biological brains, they differ in several

fundamental aspects, with ANNs often being static and symbolic in contrast to the dynamic and analog nature of organic brains (Marblestone et al., 2016; Bengio et al., 2015).

In the context of credit scoring, neural networks have proven to be a valuable algorithm and have been employed by various scoring models documented in the literature. In the subsequent sections, this chapter would delve into the components and functioning of neural networks. Specifically, the concept of Perceptron and Multi-Layer Perceptron would be presented, along with their training processes, regularization techniques, pruning methods to mitigate overfitting, and the use of optimization algorithms for hyperparameter tuning. By examining these aspects, a comprehensive understanding of neural networks and their application in credit scoring would be attained.

3.2. Perceptron and Multi-Layer Perceptron

All neural networks are constructed upon the foundation of artificial neurons. An artificial neuron consists of two essential components: an adder that calculates the weighted sum of all inputs received by the neuron, and a processing unit that utilizes this weighted sum to generate an output, employing a predefined function called the activation function. It is worth noting that each artificial neuron possesses a distinct set of weights and thresholds (biases), which are acquired through the utilization of various learning algorithms. These learning algorithms play a crucial role in adjusting the weights and biases to enable the neural network to learn and make accurate predictions. Some commonly employed learning algorithms include:

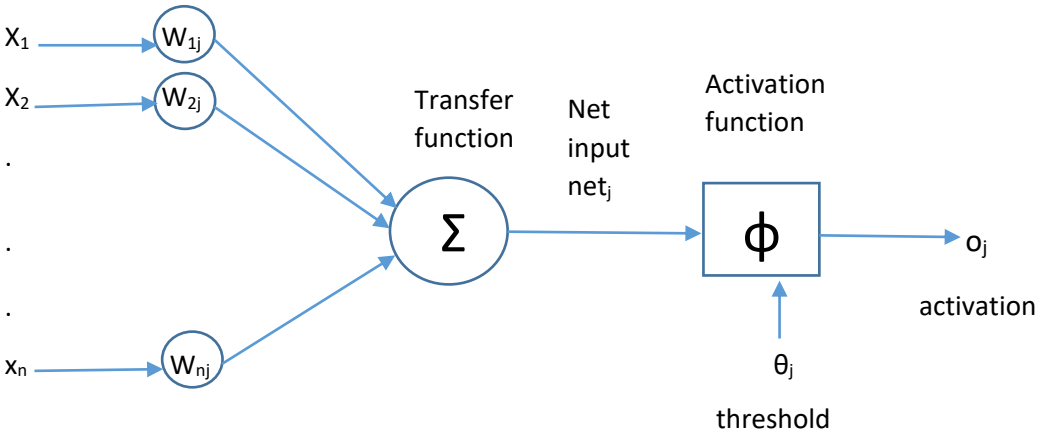


Figure 3.1. An artificial neuron is graphically illustrated (Patterson & Gibson, 2017).

It is referred to as a perceptron if there is only one layer of these neurons. Since it only buffers the input, the input layer is referred to as the zeroth layer. The output layer is the single layer with neurons. Each one of the output layer's neurons has unique weights and thresholds. The network is referred to as Multi-Layered Perceptron (MLP)

when there are presented several layers. One or more layers are concealed in an MLP. There are various numbers of hidden neurons in these layers. Each buried layer's neurons perform the same activation function:

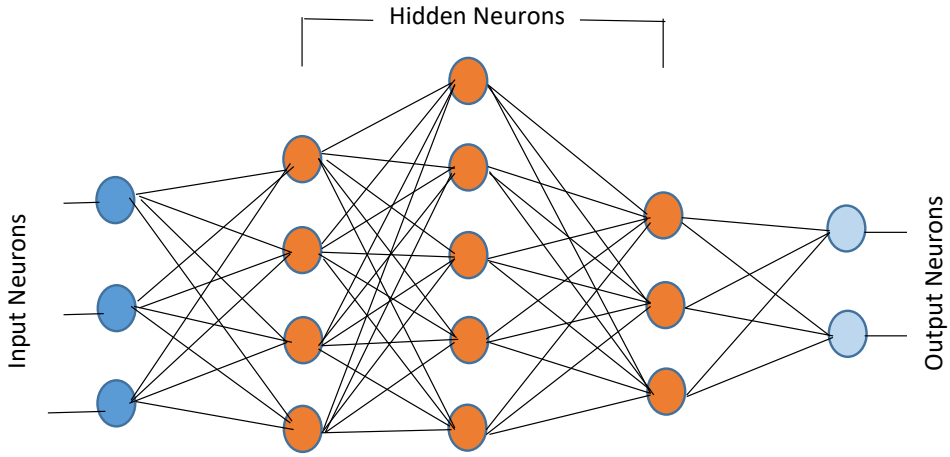


Figure 3.2. The architecture of Multi-Layer Perceptron (Patterson & Gibson, 2017).

An MLP with three inputs, and three hidden layers, each having four, five, three, and two neurons in the output layer are shown in the above Figure. In MLP, every neuron in the layer above is coupled to every other neuron in the layer below. MLPs are also known as fully connected layers since there is no feedback in the information flow in MLPs, and for this reason, these networks were often referred to as feedforward networks.

Gradient descent techniques are used to train perceptron and learn using supervised learning algorithms, in which the network is given the intended output for each input contained in the training dataset. For the objective function to be minimal, once the network has entirely learned all of the training data, an error or objective or loss function J(W) at the output was defined. A few loss functions are displayed in Table 3.1. As the gradient of the objectives function reduces, the weights of the output layer and the hidden layers are updated.

The purpose is to define the objective function's global minima. Gradient descent appears to have a vulnerability when approaching a local minimum, since the gradient would grow if it is attempted to move the network to a different location on either side, compelling it to remain there. There are several gradient descent variations designed to increase convergence and prevent becoming trapped at local minima that might be used to address this issue (adding momentum, and variable learning rate).

One of the primary methods used in perceptron for gradient descent is defining the objective function for the output layer, which also determines how the weights of neurons in the hidden layers are adjusted. The backpropagation (BPN) method is employed for this purpose, wherein the weight adjustments are computed by propagating the output error backward through the network to the hidden layers.

To sum up, a perceptron consists of:

- Inputs, X_1, X_2, …, X_n
- Weights, W_{1J}, W_{2J}, …, W_{nJ}
- The sum of product between inputs and weights, $\sum_{i=1}^{n} X_i W_{iJ}$
- The activation function, the more representative is illustrated in Table 3.2.
- Threshold or Bias, θ_J
- Output, Y_J

A feedforward technique known as a multilayer perceptron combines inputs and initial weights to determine a weighted total, and both are affected by the activation function. The main distinction from a straightforward perceptron is the passing of each linear combination to the layer underneath. The method would be impossible to determine the weights that minimize the loss function if it merely computed the weighted sums in each neuron, sent the results to the output layer, and stopped there. Similarly, to this, no real learning would occur if the algorithm simply ran one computation iteration.

Backpropagation, a learning method used by the multilayer perceptron, enables gradual adjustment of the network's weights to minimize the loss function. As multilayer perceptron often employs gradient descent as the optimization function, these functions must have a differentiable derivative. The most commonly used optimizers are listed in Table 3.3. In each iteration, the weighted sums are propagated through all layers, and the gradient of the mean squared error is calculated over all input and output pairs. The gradient value is then used to update the weights starting from the first hidden layer, propagating the changes backward through the network. This process is repeated until the gradient for each input-output pair converges, which occurs when the recently computed gradient does not differ from the previous iteration by more than a predefined convergence threshold. The value of gradient descent in each iteration is calculated using the equation provided (Patterson & Gibson, 2017).

$$\Delta_w(t) = -\varepsilon \frac{dE}{dw_{(t)}} + \alpha \Delta_{w(t-1)}$$

where,

$\Delta_w(t)$: Gradient current iteration

ε: Bias

dE: error

$dw_{(t)}$: weight vector

α: learning rate, that increases the amount of the updates (steps) a neural network makes to its parameter vector x as it moves across the loss function space.

However, to enhance the overall performance of a Multi-Layer Perceptron (MLP), it is necessary to apply a training process. Training involves adjusting the weights and biases of the MLP based on a set of training data, allowing it to learn patterns and relationships within the data. This iterative process helps the MLP improve its predictive capabilities and increase its accuracy in credit risk assessment. Through training, the MLP could optimize its parameters and adapt to different credit risk scenarios, leading to more reliable and robust predictions. By investing time and resources into the training process, financial institutions could maximize the performance of MLP models and improve their ability to effectively assess and manage credit risk.

3.3. Training Neural Networks

Transforming a dataset into a suitable input for a neural network is often a challenging aspect of network training. Selecting the appropriate neural network architecture could also be a complex decision, given the numerous approaches found in the literature for each problem category. In practice, the benefits of a new architecture over an older one could be diminished by data preprocessing, cleansing, normalization, and the construction of a larger training set. Modern neural network designs emerge from extensive scientific research and collaborations among various entities, making them difficult to create individually. Furthermore, training such models often requires substantial processing capacity. Instead of attempting to replicate the findings of recent publications, it is often more worthwhile to invest time in gathering more training data and developing a solution around a reliable but less advanced model.

The architecture of a neural network encompasses the number, type, and size of layers, and making decisions in this regard could be highly intricate. It is recommended to begin with one or two layers, train a model, and assess whether it adequately fits the training data, indicating low bias. If the model does not fit well, an alternative is to progressively increase the number of layers and the size of each layer until the model precisely matches the training set. Once this is achieved, regularization techniques could be applied if the model exhibits poor performance on the validation data, indicating high variance. If, even after regularization, the model fails to fit the training data, it might be necessary to significantly increase the network's size. The iterative process continues until the model adequately fits the chosen metrics for both the training and validation datasets (Burkov, 2019).

3.4. Regularization

During the training process, neural networks are susceptible to overfitting, where the model becomes overly complex and starts to memorize the training data instead of learning general patterns. Overfitting could lead to poor performance on unseen data and reduced generalization ability. To address this issue, the usage of regularization techniques is an effective approach.

In addition to the well-known L1 and L2 regularization techniques, neural networks offer a range of regularization methods, including dropout, early stopping, and batch normalization. While batch normalization does not strictly fall under the category of regularization, it often results in a more regularized model due to its normalization effect on the layer outputs. These regularization techniques play a vital role in mitigating overfitting and enhancing the generalization capability of neural networks.

Dropout, a conceptually simple regularization technique, involves randomly excluding a subset of units from the computation during each iteration of the training process in the network. The regularization effect becomes more pronounced as the proportion of omitted units increases, necessitating the definition of the dropout parameter for each layer. Empirical determination of the dropout parameter, which typically ranges from 0 to 1, is performed by tuning it using validation data (Chollet, 2021). In practice, regularization techniques such as L2 regularization and dropout are often combined to further enhance model performance. However, it is generally advised to avoid applying dropout to the first layer of the network to prevent the loss of crucial information from the input dataset. These regularization techniques contribute to the overall stability and robustness of neural network models by mitigating the risk of overfitting and improving their generalization performance. By carefully selecting and combining appropriate regularization methods, neural networks could achieve superior performance on unseen data and effectively address complex Real-world problems (Chollet, 2021).

Early stopping is a widely employed technique in neural network training, which involves storing the intermediate model after each epoch and evaluating its performance on a separate validation set. As the number of epochs increases, the cost function typically decreases due to the iterative nature of the gradient descent algorithm. The decreasing cost signifies that the model is effectively fitting the training data. However, there comes a point where the model might start to overfit: the cost continues to decrease, but its performance on the validation data deteriorates. At this stage, it is crucial to identify the decline in performance on the validation set, which serves as an indicator of overfitting, and terminate the training process accordingly.

Alternatively, a predetermined number of epochs could be allowed to pass before selecting the best model, rather than terminating the training immediately upon observing a decline in validation performance. These intermediate models stored at different epochs are known as checkpoints. While early stopping provides a means to prevent overfitting and select the optimal model, it should be noted that its effectiveness depends on appropriately regularizing the model. Some machine learning experts commonly employ early stopping as a regularization technique, while others face challenges in effectively applying regularization to avoid undesired behaviors (Chollet, 2021).

The technique known as *batch normalization*, which could be more accurately termed batch standardization, plays a crucial role in neural network architectures by normalizing the output of each layer before it is passed as input to the subsequent layer. In addition to its primary purpose of normalization, batch normalization exhibits

regularization effects and contributes to faster and more stable training of neural networks. Therefore, the inclusion of batch normalization is considered advantageous, and it is recommended to incorporate a batch normalization layer between two consecutive layers in neural network libraries (Géron, 2022).

In practical terms, *regularization methods such as L1 and L2* (also known as Lasso and Ridge regression, respectively) typically result in a slight increase in bias while effectively reducing variance. These regularization techniques aim to modify the objective function by introducing a penalty component that increases with the complexity of the model. Specifically, L1 regularization applies an L1 penalty equal to the absolute value of the coefficient's magnitude, whereas L2 regularization imposes an L2 penalty equal to the square of the coefficient's magnitude. Given a loss function L(W), the L1 regularization term is weighted by the scalar alpha divided by two and added to the regular loss function selected for the task at hand (Chollet, 2021). The formulation could be expressed as follows:

$$\hat{L}(W) = L(W) + \alpha \|W\|_1$$

while, regularization L2 is represented as:

$$\hat{L}(W) = L(W) + \frac{\alpha}{2} \|W\|_2^2$$

The α is called the regularization rate and is an additional hyperparameter, which determines the model's regularization degree. The higher value of α, the higher risk of underfitting, whereas the lower one, the higher risk of overfitting to data. The summarized differences between L1 and L2 regularization are presented in Table 3.4 (Heaton, 2018).

3.5. Pruning

In addition to regularization techniques that mitigate overfitting, there are other methods that could enhance the performance of neural network models, including pruning techniques. The method of neural network pruning is based on the logical idea of eliminating redundant components from a well-performing network that consumes excessive resources. Although large neural networks have demonstrated their ability to learn, it has been observed that not all parts of the network remain useful after training. The objective of pruning is to remove these unnecessary elements while maintaining the network's performance.

There are several techniques available for pruning neural networks, with weight pruning being one of the primary approaches. In weight pruning, the network is made sparse by setting certain parameters to zero, while keeping the overall architecture intact. This results in a reduced parameter count without altering the network structure. Another approach involves removing individual nodes from the network, leading to a smaller network design while striving to preserve the accuracy achieved by the larger original network.

The process of pruning is often guided by three main factors. Weight magnitude criteria involves pruning weights with the least absolute value. Gradient magnitude pruning takes into account the gradients computed over a small sample of training data, and pruning decisions are made based on the product of the gradient and the associated weight for each parameter. The choice of whether the pruning criterion is applied globally to all network parameters or filters, or if it is computed separately for each layer, depends on whether global or local pruning is employed.

By employing pruning techniques, neural networks could be made more efficient by reducing their size and computational requirements, while aiming to retain their performance and accuracy. Pruning provides a valuable approach for optimizing and streamlining neural network architectures in practical applications. To conclude, in Table 3.6 referred the most noticeable benefits and drawbacks of pruning a neural network (Blalock et al., 2020).

In conclusion, pruning and regularization techniques play vital roles in optimizing neural network models. Pruning aids in reducing model complexity by eliminating unnecessary connections and weights, thereby improving the network's efficiency and interpretability without significant loss of performance. Regularization, on the other hand, addresses the issue of overfitting by adding a penalty term to the loss function, promoting generalization, and preventing the model from memorizing the training data.

3.6. Hyperparameter tuning

In the pursuit of enhancing the performance of neural network models, considerations extend beyond regularization and pruning techniques. An integral aspect of model optimization involves determining the appropriate values for hyperparameters.

First of all, the term *Hyperparameter* is referred to every configuration option that allows users to choose and that might have an effect on performance. Neural networks have several categories of hyperparameters, such as:

- Layer size, the number of neurons into each layer,
- Magnitude, that is the momentum and learning rate,
- Regularization, like dropout, drop connect, L1 and L2,
- Activation functions, as they are represented in Table 3.2,
- Weight initialization strategy, as Geron (2022) refers.
- Loss functions, as they are represented in Table 3.1,
- Settings for epochs during training, mini-batch size, that is the number of vectors that passed into the learning algorithm simultaneously,
- Normalization scheme for input data.

The selection of hyperparameters in neural networks is typically influenced by the type of problem being addressed, including choices such as activation functions, loss functions, and layer sizes. Table 3.5 describes the most commonly used combinations of these hyperparameters. It is important to note that excessively large numbers of

neurons in a layer should be avoided as it could lead to overfitting. Regularization techniques are employed to mitigate this risk.

The initialization of weights is a crucial aspect of the learning process, and the choice of activation function determines the appropriate strategy. For activation functions belonging to the ReLU family, the He initialization strategy is recommended, whereas other activation functions like tanh or identity typically use the Xavier strategy. The selection of weight initialization plays a significant role in ensuring stability and effectiveness during training.

The learning rate hyperparameter is one of the most critical factors affecting the training process in neural networks. A high learning rate could lead to unstable training, while a low learning rate might result in ineffective training. Ideally, a higher learning rate is initially set and gradually decreased as convergence is approached. Empirically, learning rate values in the range of [0.1, 0.01, 0.001] are often reliable and recommended for starting, with 0.001 being a commonly used initial value.

The choice of batch size affects the training speed of the neural network. Increasing the batch size initially leads to a decrease in training time. It is common practice to use batch sizes that are powers of two. For small networks, batch sizes ranging from 32 to 256 are commonly used for CPU training, while 32 to 1024 are typical for GPU training. When the mini-batch size is doubled, the number of epochs must be doubled as well to maintain the same number of parameter updates (Patterson & Gibson, 2017).

To conclude, the recommendations discussed above provide a framework for discovering the optimal values of these hyperparameters. This tuning process, also known as hyperparameter optimization, has garnered significant attention in the literature, leading to the development of numerous optimization algorithms.

3.7. Optimization algorithms for tuning

As mentioned in the above section, there are numerous optimization algorithms for hyperparameter tuning. These algorithms provide efficient and automated ways to explore the hyperparameter space and find the optimal configurations. The choice of the specific optimization algorithm depends on factors such as the size of the search space, computational resources, and time constraints. The proposed model in this thesis employed a limited set of hyperparameter tuning techniques, including Bayesian optimization, random search, quasi-Monte Carlo, and genetic algorithms.

Bayesian optimization

The Bayes theorem and its approach to stochastic processes for measuring variables counting their event probability and event of uncertainty are the foundations of the Bayesian optimization strategy. This method might be thought of as a substitute for the Gaussian process. A complicated prediction model with several hyper-

parameters might also be quickly searched for using Bayesian optimization while retaining the model's accuracy.

The Bayes theorem, in a few words, states that the posterior probability of a model M gave evidence E is proportional to the likelihood of E given M multiplied by the prior probability of M:

$$P(M|E) \propto P(E|M)P(M)$$

Bayesian optimization considers the f(x) function as a "black box" when dealing with an optimization problem in machine learning. The phrase "black box" refers to an item that could only be understood in terms of its inputs and outputs, without any understanding of its underlying workings, according to the definition of computing science. The only way to evaluate the f(x) function is to sample at a point x and obtain a potential response because it lacks an analytical formulation (Theodoridis, 2015).

The Tree-structured Parzen estimator (TPE), a Bayesian optimization variation that divides data into good and poor groups and uses the density ratio of those groups as an acquisition function, is a popular and adaptable HPO technique (AF). However, there are frequently certain limitations in Real-world applications, such as latency or memory needs (Watanabe & Hutter, 2022). The TPE technique is made to optimize quantization hyperparameters to obtain the highest potential latency improvement and an estimated accuracy objective. TPE is an iterative procedure that builds a probabilistic model from the history of evaluated hyperparameters and utilizes it to recommend the next set of hyperparameters to assess.

Sequential Model-Based Optimization (SMBO) is a method used by the Tree-structured Parzen Estimator (TPE). Based on prior measurements, SMBO approaches successively build models to approximate the performance of hyperparameters, and then pick new hyperparameters to test based on this model. The TPE technique models P(x|y) and P(y), where x is the related quality score and y, is the hyperparameters. By altering the generating process of hyperparameters and substituting non-parametric densities for the configuration prior's distributions, P(x|y) is modeled.

It is advised to run TPE for at least 200 iterations in order for it to get an ideal solution, which takes many iterations. This method might take anywhere from 24 hours to a few days to finish depending on the model because every iteration involves an assessment of a created model, which includes accuracy measurements on a dataset and latency measurements using a benchmark.

There are many benefits since TPE allows for the definition of the tuning time of quantization, the error tolerance, and the target latency improvement. It also enables a large variety of variables in the parameter search space. This method ensures that the greatest possible accuracy and latency are obtained from the combination of ideal parameters. However, the main drawback is that TPE does not model interactions between hyperparameters (Bergstra et al., 2013).

Grid Search

The most popular method for hyperparameter tuning is the grid search because of its easy-to-understand process. Being an ignorant search strategy, it does not take into account the results of earlier iterations. In order to find the hyperparameter combination that produces the highest performance, this approach involves evaluating each distinct combination of hyperparameters in the search space. Unfortunately, this method does not scale well; as the size of the hyperparameter search space grows, run time and computation would increase exponentially (Syarif et al., 2016). As a result, this particular hyperparameter tuning algorithm was not utilized in this study.

Random Search

Another ignorant search technique that treats iterations separately is the random search. But it just examines a predetermined amount of hyperparameter sets randomly, rather than looking for all of them in the search space. The user chooses this quantity. In comparison to grid search, the approach uses less computation and run time since it executes fewer trials for hyperparameter tuning. Unfortunately, the random search presents the risk of omitting the optimal collection of hyperparameters and skipping the peak model performance since it tries hyperparameter sets at random (Romeijn, 2008).

NSGA-II: Non-dominated Sorting Genetic Algorithm

A modified mating and survival selection is used, and the algorithm generally follows the structure of a genetic algorithm. First, individuals are chosen frontally in the NSGA-II. As a result, a front would need to be divided since not every person would be permitted to live. Based on crowding distance, solutions are chosen in this dividing front. The Manhattan Distance, or "crowding distance," exists in the objective space. The crowding distance of infinity is given to the extreme points since they are wanted to be preserved during each generation. Additionally, the NSGA-II employs a binary tournament mating selection to exert some additional selection pressure. Prior to crowding distance, each individual is first evaluated according to rank. In the original C code, there is also an alternative that uses the dominance criterion between two solutions in place of the rank (Deb et al., 2002).

Quasi Monte Carlo Sampler

In comparison to conventional random sequences, Quasi-Monte Carlo (QMC) sequences are designed to exhibit fewer discrepancies. They have been found to outperform random sequences in hyperparameter optimization tasks. Monte Carlo (MC) sampling is a technique used to estimate expectations by taking sample averages of random values. The law of large numbers provides the theoretical basis for this approach. Quasi-Monte Carlo (QMC) sampling, on the other hand, employs deterministic sequences instead of random ones, allowing us to leverage the law of large numbers. Similar principles apply to both Monte Carlo and quasi-Monte Carlo techniques. The main challenge lies in computing the average value of a function, denoted as f, when it is evaluated at a series of locations, x_1, ..., x_N.

$$\int_{[0,1]} f(u)du \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Each x_i is a vector with s elements since the unit cube with s dimensions is integrated. The method used to choose the x_i distinguishes Quasi-Monte Carlo from Monte Carlo. MC adopts a pseudorandom sequence, whereas QMC uses a low-discrepancy sequence and as a result converges quicker than MC (Owen, 2003).

In conclusion, this chapter has provided a comprehensive overview of various aspects related to neural network training, architecture, regularization, and hyperparameter optimization. Looking ahead, the next chapter would shift focus to multi-criteria decision analysis with fuzzy logic. This methodology introduces a different perspective by incorporating fuzzy sets and fuzzy logic to handle decision-making problems involving uncertainty and imprecision. Complex decision issues involving the examination of numerous competing criteria might be efficiently solved by merging fuzzy logic with multi-criteria decision analysis.

The transition to multi-criteria decision analysis with fuzzy logic reflects the broader scope of this thesis, which aims to explore various computational intelligence techniques and their applications in decision support systems. By combining the knowledge and insights gained from the neural network chapter with the fuzzy logic approach in the upcoming chapter, could be developed a comprehensive framework for tackling Real-world decision problems with enhanced accuracy and robustness.

In summary, the neural network chapter has laid the foundation for understanding the principles, methodologies, and practical considerations in neural network training and optimization. The integration of multi-criteria decision analysis with fuzzy logic in the subsequent chapter would further broaden our understanding and provide valuable insights into the application of computational intelligence techniques for decision support.

Chapter 4:
Multi-criteria Decision Analysis with Fuzzy Logic

4.1. Introduction

This chapter examines the fundamental principles of Fuzzy Logic and its integration with Multi-criteria Decision Analysis (MCDA) methodologies within the proposed Neuro-Fuzzy model presented in this thesis.

Fuzzy logic, introduced by Lot Zadeh in 1965, is grounded in the mathematical theory of fuzzy sets. It addresses the relationship between precision and uncertainty, recognizing that problems in the Real-world often possess degrees of truth or falsehood, goodness or badness, rather than being strictly binary. The fuzzy theory seeks to bridge this linguistic gap by providing a means to handle imprecise and uncertain information within mathematical models, ultimately enhancing the decision-making process. An additional advantage of fuzzy logic is its capacity to incorporate human reasoning through the use of fuzzy rules, which are expressed in natural language. For instance, in the context of a driver approaching a traffic light, a few illustrative If-Then rules could be formulated as follows:

- If the light is red, the driver's speed is high, and the distance between them is close, then the driver should brake hard.

- If the light is red, the driver's speed is low, and the distance between them is long, then the driver should maintain their speed.

- If the light is orange, the driver's speed is moderate, and the distance between them is long, then the driver should brake gently.

- If the light is green, the driver's speed is low, and the distance between them is close, then the driver should accelerate.

The aforementioned example highlights the approximate estimation of input variables by the human brain, which aligns with the notion of precision degrees in fuzzy logic.

This chapter aims to provide a comprehensive overview of the components comprising a Fuzzy Inference System, namely fuzzification, operators on fuzzy sets, the knowledge base, Fuzzy Inference Mechanism, and defuzzification. Additionally, the

chapter explores the integration of the Fuzzy Analytic Hierarchy Process (FAHP) and the Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS) within the proposed Neuro-Fuzzy model.

However, in order to foster a deep understanding of fuzzy logic, it is imperative to examine its historical development. By delving into the historical perspective, we could enhance our comprehension of the foundations of fuzzy set theory and fuzzy logic. Accordingly, this chapter references the contributions made by eminent scientists who have played significant roles in the advancement of fuzzy logic. Furthermore, it acknowledges the influence of Greek philosophers, particularly Aristotle, who holds a prominent position in the realm of fuzzy logic. By exploring the historical trajectory, the ability to grasp the fundamental principles and historical evolution of fuzzy set theory and fuzzy logic could be augmented.

4.2. Historical review

Fuzziness was mentioned many years ago, as the famous ancient Greek philosopher Aristotle supports that “A well-trained mind would be content with the level of precision that the subject's nature permits rather than pursuing accuracy when just a rough approximation of the truth could be achieved.” (Carter et al., 2021; Aristotle, 3rd BCE).

Eubulides of Miletus in the 4th century BCE studied the impression through the “Sorites” paradox, which did not have an obvious answer, since it was wondered “When does a collection of grains of sand become a heap?” (Moline, 1969). Specifically, “If 1.000.000 grains of sand are a heap of sand, then a heap of sand minus one grain is still a heap? What happened if they are reduced by another one or two or three grains?” (Sorensen & Roy, 2009). The whole problem was the difficulty of correctly assigning a threshold of “truth” to a proposition.

All classical logic "habitually assumes that precise symbols are being used," according to Bertrand Russell (1923). Therefore, it only applies to a hypothetical heavenly existence and not to our life on Earth. (Bertrand Russell, 1923; Carter et al., 2021). Every set theory with an unlimited comprehension principle results in contradictions, as demonstrated by Russell's dilemma (Irvine & Deutsch, 2021). Bertrand Russell (19th century), a British philosopher and mathematician who lived in the 19th century, and in his book with the title “Principia Mathematica” refers to the constructed mathematical logic, which is consisted of three aims. Firstly, there is a need to minimize the number of primitive notions, axioms, and inference rules. Secondly, mathematical propositions should be expressed precisely in symbolic logic by using the most convenient notation. Thirdly, it is necessary to resolve the paradoxes of symbolic logic and set theory (Whitehead & Russell, 1962).

Friedrich Waismann (19th century) was an Austrian mathematician, physicist, philosopher, and member of the Vienna Circle. He introduced the meaning of open

texture or porosity intending to describe the global possibility of vagueness in real situations (Audi & Robert, 1999). To be more specific, even if scientists try to precisely determine a statement, vagueness would remain, due to an indefinite number of possibilities (Freeman et al., 2013). Jan Łukasiewicz (1951) introduced philosophical logic, mathematical logic, and the history of logic and his scientific research expanded the methods of Aristotelian logic (Łukasiewicz, 1951).

Last but not least, Lotfi Zadeh (1975) was a professor of computer science at the University of California, Berkeley, as well as a mathematician, computer scientist, electrical engineer, and researcher in artificial intelligence. The contribution made by Zadeh to the field of fuzzy logic science is significant. Several well-known terminologies are introduced by him, including fuzzy sets, fuzzy logic, fuzzy algorithms, fuzzy semantics, fuzzy languages, fuzzy control, fuzzy systems, fuzzy probabilities, fuzzy events, and fuzzy information (Zadeh, 1975).

4.3. Fuzzy Inference System

Fuzzy Inference System (FIS) is the most essential part of fuzzy logic systems. It uses fuzzy set theory, fuzzy rules, and approximate reasoning intending to discover the output which corresponds to crisp inputs. This system tries to imitate the human's thinking process using reasoning and this is achieved by Fuzzy If-Then rules. Figure 4.1 represents the Fuzzy Inference System structure (Singh & Lone, 2020).

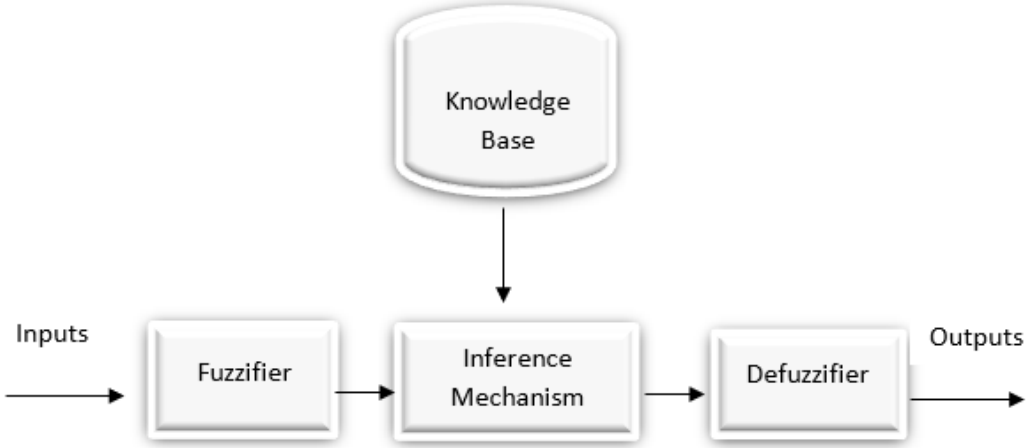


Figure 4.1. Fuzzy Inference System (Singh & Lone, 2020).

Figure 4.1 presents a diagram illustrating the process of a Fuzzy Inference System (FIS). This system encompasses various stages, beginning with the provision of crisp input values to the fuzzifier. The fuzzifier applies fuzzy membership functions, resulting in the generation of fuzzy sets. Subsequently, the Inference mechanism applies fuzzy rules sourced from the knowledge base, leading to the production of fuzzy outputs. Finally,

the defuzzifier transforms the fuzzy outputs back into the original domain (Singh & Lone, 2020).

In the forthcoming chapters, each essential component of the FIS would be comprehensively explored to provide a comprehensive understanding of the entire process. Consequently, the initial phase of fuzzification would be introduced.

4.4. Fuzzification

Decision-making processes that involve non-random uncertainty, characterized by imprecise natural language expressions, have demonstrated inherent limitations. Lotfi Zadeh (1965) proposed that the concept of set membership is crucial in addressing uncertainty and introduced a set theory that operates within the range of [0, 1]. Zadeh asserted that a fuzzy set consists of objects whose membership values are determined by a membership function, which serves as a characteristic function assigning a degree of membership to each object within the set. These membership values are typically represented by numerical values within the interval [0, 1] (Zadeh, 1965). During the fuzzification process, crisp inputs are transformed into fuzzy sets through the utilization of the membership function.

Fuzzy set

Definition 4.1 *Let* X *be a space of points (objects), with a generic element of* X *denoted by x. A fuzzy set A in* X is *characterized by a membership function* $\mu_A(X)$ *which associates with each point x a real number in the interval* [0,1] *representing the grade of membership of x in A (Rutkowska, 2001).*

$$A = \{(x, \mu_A(x)), \quad \forall x \in X\}$$

where

$$\mu_A(x): X \rightarrow [0,1]$$

The nearer the value of $\mu_A(X)$ *to unity, the higher the grade of membership of x in A. If* $\mu_A(X) = 1$, *then x fully belongs to A. If* $\mu_A(X) = 0$, *then x does not belong to A. Space X is called the universe of discourse.*

When the universe of discourse (X) is a finite set, a fuzzy set A could be represented as:

$$A = \sum_{\iota=1}^{n} \frac{\mu_A(x_\iota)}{x_\iota}$$

In case the universe of discourse is not a finite set, fuzzy set A could be represented as:

$$A = \int_{X} \frac{\mu_A(x)}{x}$$

Figure 4.2 shows examples of Gaussian and triangular membership functions which characterize fuzzy sets A and B, respectively, in the universe of discourse X = $\mathbb{R}$.

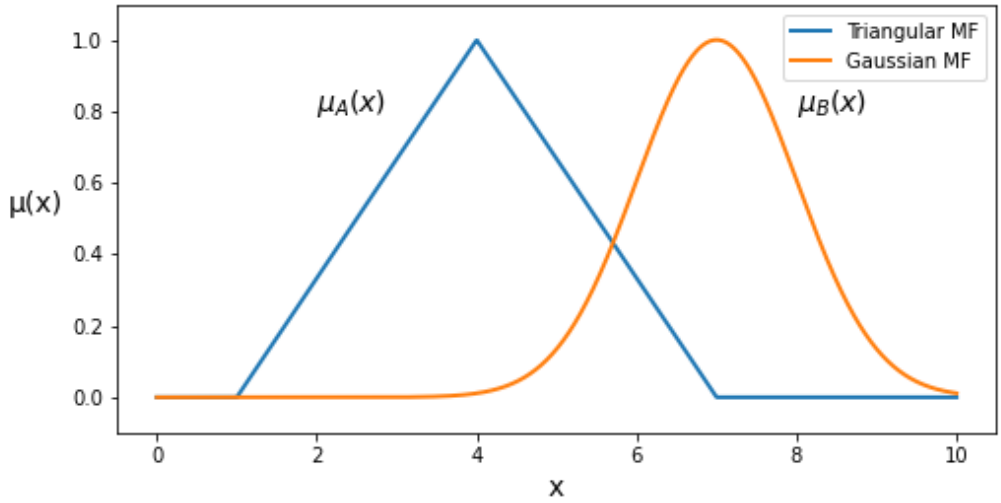


Figure 4.2. Graph of membership functions (Rutkowska, 2001).

It is worth mentioning that there are various shapes of membership functions besides the above-mentioned, Gaussian and triangular, as well as the trapezoidal membership functions are the most widely applied in fuzzy systems. Other definitions concerning fuzzy sets, such as fuzzy singleton, support, core, height, equality of a fuzzy set, normal, empty, convex, type two fuzzy set, fuzzy number, and interval, a-level set, are presented in Table 4.1.

Fuzzy membership functions

A membership function (MF) represents the degree of truth in fuzzy logic and a degree of membership function could be a continuous number over the interval [0,1]. For instance, in the case that we must brake our car before a traffic light there is not an assigned discrete value 0 or 1 to this behavior, but brake usually takes a low value for a gentle brake instead of a rough brake.

Definition 4.2 *A membership function on a set A is a characteristic function that transforms a classic value (crisp value) to a fuzzy one, thus fuzzy set is created. The membership function of a fuzzy set is donated* μ_A *(Rutkowska, 2001).*

This concept in a fuzzy set is represented by the membership functions and in this section, various types of membership functions, such as triangular, trapezoidal, Gaussian, Generalized Bell, and sigmoidal, are explained in Table 4.2.

4.5. Operators on fuzzy sets

The process of fuzzification plays a crucial role in the application of fuzzy logic. It involves converting crisp input values into fuzzy sets through the use of membership functions. Once the input values are fuzzified, various fuzzy operators could be applied to manipulate and reason with the fuzzy sets. These fuzzy operators provide the necessary tools to handle the linguistic gap and capture the complexity of Real-world problems, enabling more flexible and nuanced reasoning in the decision-making process.

In this section, the classical set-theoretic operations commonly employed in ordinary set theory were extended to accommodate fuzzy sets. To maintain consistency, the same symbols as those used in set theory are utilized when extending these operations to fuzzy sets. Table 4.3 provides an illustration of various operators, including complement, union, intersection, containment, T-norm, and S-norm.

Regarding the complement operation, the assigned value is interpreted as the membership grade of an element x belonging to the negation of the fuzzy set A. For example, if A represents a fuzzy set denoting high atmospheric humidity, the complement fuzzy set comprises weather conditions characterized by moderate, low or the absence of humidity in the atmosphere. In order to properly represent the complement operation, function c must satisfy at least the following two criteria, as outlined by Zadeh (1965).

- c (0) = 1 and c (1) = 0, which means that c behaves as the classical complement for crisp sets (boundary conditions).
- For all $a, b \in [0, 1]$, if a < b, then c(a) ≥ c(b), where a and b represent degrees of membership. This means that c is a monotonic non-increasing function.

Figure 4.3 portrays the operation of complement, supposing that A is a normally distributed fuzzy set.

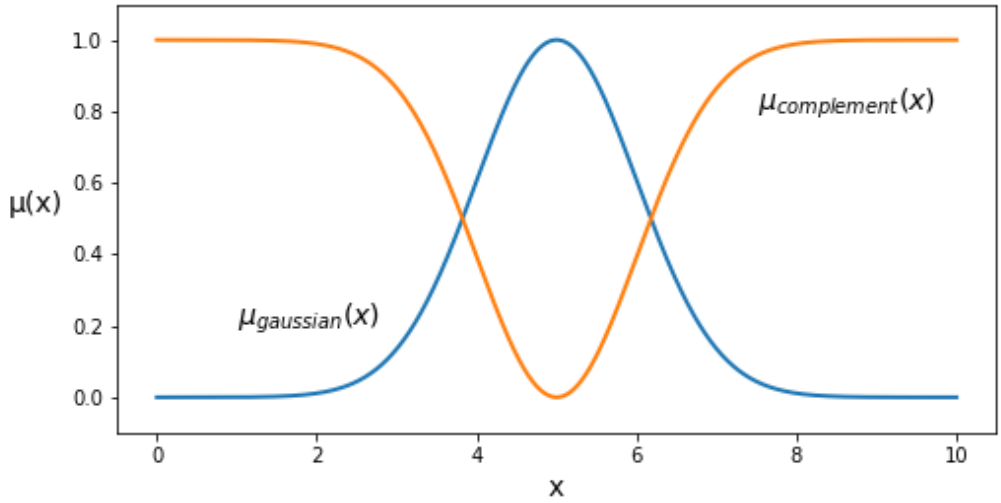


Figure 4.3. Complement operation of a normal fuzzy set (Rutkowska, 2001).

In fuzzy sets the classical union and intersection operations of ordinary subsets of X are valid. The union and intersection operations have the associative property, which means that *A U (B* U C) = *(A* U *B) U C* and *A* ∩ *(B* ∩ C) = *(A* ∩ *B)* ∩ *C* for fuzzy sets *A, B,* C in *X.* Generally, the union for N fuzzy sets A, B, C, … N is given by

$$\mu_{A \cup B \cup C \ldots \cup N} = max[\mu_A(x), \mu_B(x), \mu_C(x), \ldots, \mu_N(x)] \qquad \forall\, x \in X$$

Whereas, in a similar manner the intersection for N fuzzy sets A, B, C, … N is given by

$$\mu_{A \cap B \cap C \ldots \cap N} = min[\mu_A(x), \mu_B(x), \mu_C(x), \ldots, \mu_N(x)] \qquad \forall\, x \in X$$

In addition to the standard operations of union, intersection, and complement, it is important to note that these are not the sole feasible extensions of classical set theory to fuzzy set theory. Table 4.3 provides additional definitions of fuzzy set operations as suggested by Zadeh.

One such extension involves the formulation of a general class of intersection operators for fuzzy sets using triangular norms, also known as T-norms. These T-norms serve as a means to determine the degree of intersection between fuzzy sets. On the other hand, a general class of union operators is defined by S-norms, which are also referred to as T-conorms. S-norms facilitate the determination of the degree of union between fuzzy sets.

By introducing these additional definitions of operations, Zadeh aimed to provide a broader range of tools and methods for manipulating and analyzing fuzzy sets within fuzzy set theory.

4.6. Knowledge Base

Once the fuzzy sets have been manipulated using fuzzy operators, the resulting fuzzy outputs serve as the foundation for the knowledge base. By incorporating human reasoning into the model, fuzzy logic provides a powerful mechanism for handling complex and uncertain decision-making scenarios. The knowledge base serves as a vital component in the overall fuzzy inference system, bridging the gap between fuzzy inputs and fuzzy outputs, and enabling the system to make informed and contextually appropriate decisions.

Reasoning with fuzzy logic theory is approximate since it is based on fuzzy premises and implications, which lead to fuzzy conclusions. Approximate reasoning is applied by intelligent systems thanks to human thinking imitation. In this way, all fuzzy systems perform approximate reasoning processes through fuzzy inference. The most crucial component of approximate reasoning is the compositional rule of inference. Below some translation rules introduced by Zadeh are represented in Table 4.4 and provide the possibility of representing common linguistic statements in terms of propositions in human language (Zadeh, 1979).

According to Greek philosopher Theophrastus or Tyrtamus, the combination of a premise A' and an implication A ⇒ B, leads to a conclusion B' (Bobzien, 2002). This syllogistic introduces the modus tollens argument, and in its generalized form, it is considered a fuzzy relation assuming that a fuzzy condition, if A then B, as implication A B, with the complement fuzzy sets A' (antecedent) and infer a complement fuzzy set B' (consequent) (Zadeh, 1973). In a similar manner, under some properties which are presented in Table 4.5 the argument of generalized modus ponens has been introduced, which supports that given as premises:

I. if x is A then y is B and
II. x is a complement of A,
then, as a consequence, y is a complement of B (Fuller, 2000).

Below presented the definition of an implication function (Trillas & Valverde, 1985).

Definition 4.3 *A continuous function*

$$I: \ [0,1] \times [0,1] \rightarrow [0,1]$$

is an implication function if and only if every a, a', b, b', c $\in$ [0,1] satisfy the properties (Rutkowska, 2001):

$$If \ a \leq a' \ then \ I(a,b) \geq I(a',b)$$

$$If\ b \leq b'\ then\ I(a,b) \leq I(a,b')$$

Falsity $I(0,a) = 1$

Neutrality $I(1,a) = a$

Exchangeability $I(a,I(b,c)) = I(b,I(a,c))$

Fuzzy sets serve as a powerful tool for handling vagueness within various financial models. In particular, fuzzy sets could be effectively represented as linguistic variables, where the value of a linguistic variable typically corresponds to a fuzzy number or a linguistic term.

To illustrate, consider the example of speed levels, where three membership functions are employed to define the spaces of discourse. Linguistic variables associated with this scenario consist of three distinct values, namely "slow," "medium," and "fast," which map to their respective speed levels. Figure 4.4, as presented by Fuller (2000), visually represents these linguistic variables and their corresponding membership functions.

By incorporating linguistic variables and fuzzy sets into financial models, the ability to capture and analyze imprecise or uncertain information is significantly enhanced. This, in turn, contributes to more robust and flexible decision-making processes within the financial domain.

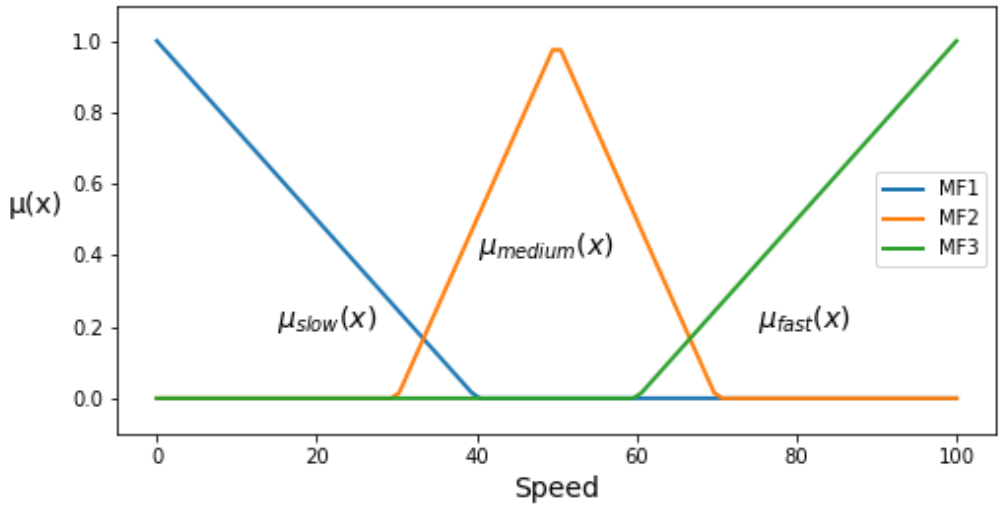


Figure 4.4. Values of linguistic variable speed (Author, 2023).

In the fuzzy inference process, firstly the crisp values are transformed into fuzzy values, and a fuzzy set is created. In continuance, rules based on knowledge are

constructed and aggregated. In the *aggregation of rules*, all the rules are combined by an aggregation operator (Agg) into one rule which is used to obtain C from A.

$$R = Agg(R_1, R_2, \dots, R_n)$$

For aggregation operations, three conditions should be verified. The first condition is commutativity, which means that all objects on which the aggregation operation would be applied could be unordered and could contain duplicate values. The second one is monotonicity, for each $x_i \geq y_i$ in fuzzy sets Y and X, respectively, membership values of x are greater than or equal to the membership values of y, $R(x_i) \geq R(y_i)$. The last condition is fixed identity, supposing that if a few rules don't satisfy an output, these rules would not impact the output of other rules (Singh & Lone, 2020).

The combination of the above three conditions is called Monotonic, Identity, Commutative, Aggregation (MICA), and its main operators, which depend on the rule's sentence connective, are presented below (Fuller, 2000).

If the sentence connective in a rule could be interpreted as AND, then:

$$R = \bigcap_{i=1}^{n} R_i$$

or

$$R(u, w) = \bigcap_{i=1}^{n} R_i\,(u, v, w) = min(A_i(u) \rightarrow C_i(w))$$

or, by using a t-norm T,

$$R(u, w) = T(R_1(u, w), \dots, R_n(u, w))$$

In the other case, if the sentence connective is OR, then:

$$R = \bigcup_{i=1}^{n} R_i$$

or

$$R(u, w) = \bigcup_{i=1}^{n} R_i\,(u, v, w) = max(A_i(u) \rightarrow C_i(w))$$

or, by using a t-conorm S,

$$R(u,w) = S(R_1(u,w),\ldots,R_n(u,w))$$

Then C from A is computed by the compositional rule of inference as

$$C = A \circ R = A \circ Agg(R_1,\ldots,R_n)$$

4.7. Fuzzy Inference Mechanism

Once the knowledge base, or rules base, has been established, it serves as the guiding framework for the fuzzy inference mechanism. The fuzzy inference mechanism is responsible for interpreting and applying the fuzzy rules to the current input values in order to generate appropriate output fuzzy sets. This mechanism follows a set of predefined inference methods, such as Mamdani or Takagi-Sugeno, which determine how the fuzzy rules are combined and how the output fuzzy sets are calculated. In the inference process, the mechanism evaluates the degree of membership of the input values in the fuzzy sets defined by the rules. It then applies fuzzy logic operations, such as implication and aggregation, to determine the overall degree of membership for each output fuzzy set. The fuzzy inference mechanism utilizes the knowledge base and leverages the fuzzy rules to make informed decisions and draw conclusions based on the given inputs. By incorporating human expertise and linguistic reasoning, the fuzzy inference mechanism enables the system to effectively handle uncertainty and imprecision, providing valuable insights and solutions in complex decision-making scenarios.

The most often used Fuzzy Inference mechanisms were derived from Mamdani, Takagi-Sugeno, Larsen, and Tsukamoto (Singh & Lone, 2020). Mamdani's fuzzy inference operator is described below in more detail and other fuzzy inference operators are in Table 4.6.

For simplicity, two fuzzy control rules are assuming, that is:

R_1: if x is A_1 and y is B_1 then z is C_1

R_2: if x is A_2 and y is B_2 then z is C_2

Fact: x is x_0 and y is y_0

Consequence: z in C

Mamdani

The fuzzy implication is expressed by Mamdani's minimum operator and the sentence connective is defined by the max operator. The firing levels of the rules α_i , i = 1,2.

$$\alpha_i = A_i(x_0) \wedge B_i(y_0)$$

For individual rule output,

$$C'_i(w) = (\alpha_i \wedge C_i(w))$$

Then the global system output is calculated,

$$C(w) = C'_1(w) \, V \, C'_2(w)$$

At last, one of the defuzzification methods described above would be applied in order to determine the output. Graphically Mamdani's implication operator is portrayed in Figure 4.5.

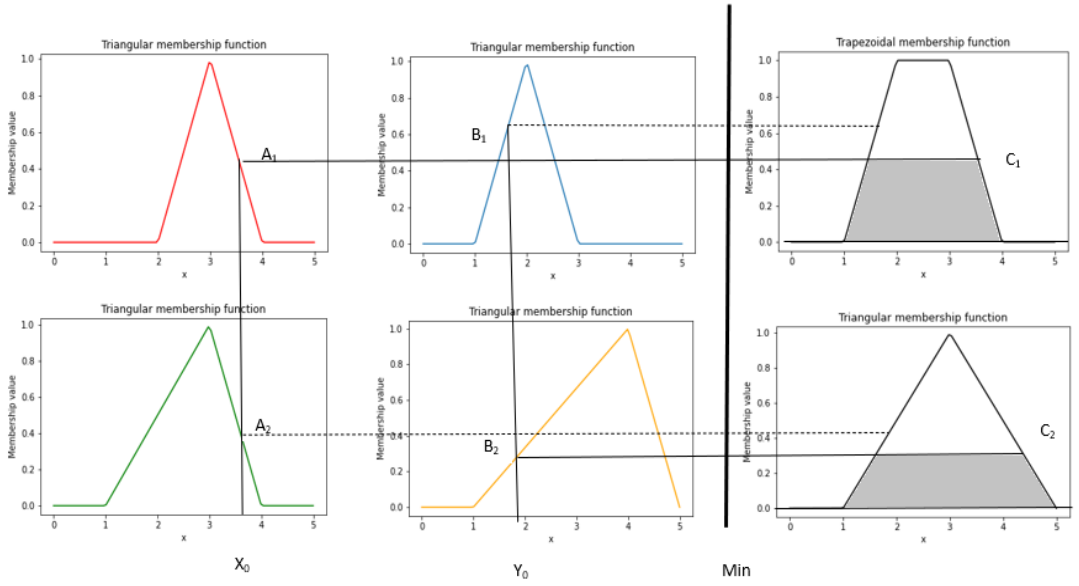


Figure 4.5. Mamdani's implication operator (Fuller, 2000).

4.8. Defuzzification

After the fuzzy inference mechanism has generated the output fuzzy sets based on the input values and fuzzy rules, the next step in the fuzzy logic process is defuzzification. Defuzzification is the process of converting the fuzzy output sets into crisp values that could be easily understood and utilized. This step involves mapping the fuzzy sets to a

crisp output value or a set of crisp values that represent the final decision or action and its definition could be the following:

Definition 4.4. *Defuzzification is a selection process of a representative object from the fuzzy output C which is inferred by the fuzzy system algorithm* (fuller, 2000).

Various defuzzification methods could be employed, such as centroid, maxima methods, and the most popular defuzzification operators are displayed in Figure 4.6 and their definitions in Table 4.7.

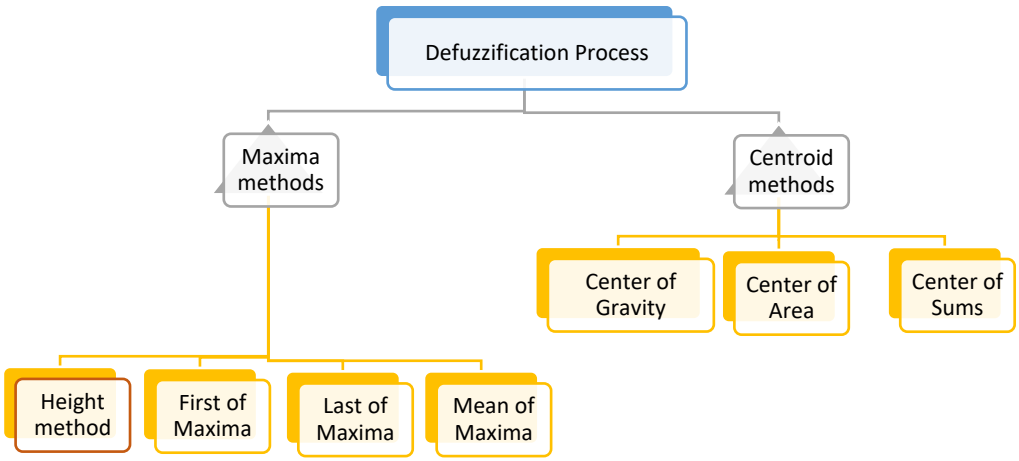


Figure 4.6. Popular defuzzification operators (Author, 2023).

These methods take into account the shape and distribution of the fuzzy output sets to determine the most appropriate crisp value or values. Defuzzification plays a crucial role in making the fuzzy logic results interpretable and actionable. By converting the fuzzy outputs into crisp values, the information obtained from the fuzzy inference mechanism could be effectively utilized in practical applications and decision-making processes.

In conclusion, the fuzzy inference system (FIS) serves as a powerful tool for handling uncertainty and making decisions in complex and ambiguous domains. By utilizing fuzzy sets, fuzzy rules, and the fuzzy inference mechanism, FIS could effectively capture and represent the imprecise and uncertain nature of Real-world problems. The process of fuzzification allows for the conversion of crisp inputs into fuzzy sets, enabling the incorporation of linguistic variables and human reasoning into the system. The fuzzy rules, derived from expert knowledge or data-driven approaches, provide a framework

for making informed decisions based on the defined relationships between inputs and outputs. The fuzzy inference mechanism combines the fuzzy inputs and rules to generate fuzzy outputs, which are then defuzzified to obtain crisp results. This process facilitates the modeling of complex decision-making processes and enhances our ability to handle subjective and uncertain information. The flexibility and interpretability of FIS make it a valuable approach in finance. By leveraging the strengths of fuzzy logic, FIS offers a robust framework for addressing Real-world problems and improving decision-making processes in the face of uncertainty.

4.9. Fuzzy Analytic Hierarchy Process (FAHP)

Moving forward from the fuzzy inference system (FIS), another powerful methodology in the realm of decision-making is the Fuzzy Analytic Hierarchy Process (FAHP). While FIS focuses on capturing and processing imprecise information through fuzzy logic, FAHP aims to incorporate human expertise and preferences into the decision-making process.

In recent years, the field of multi-criteria decision-making (MCDM) has been expanded quite quickly. In MCDM, a decision-maker could rank options according to a finite number of criteria that are weighted by the decision-maker according to the significance of each criterion. Hence, MCDM is a cutting-edge field that might be used to solve complicated decision-making issues by offering a practical method for evaluating options. If the relevant decision-making problem's proper criteria are followed, the intended outcome could be obtained.

Thomas Saaty (1980) initially created the analytical hierarchy process (AHP), a commonly used MCDM approach. AHP employs hierarchical dependencies between criteria, starting weights for criterion and sub-criteria, and both (Ziemba, 2019). When the options in an issue are hard to be quantified and compare with each other, AHP is a potent organized strategy used to organize and evaluate complicated judgments. If the uncertainty in comparison judgment is not taken into account, AHP yields suitable conclusions. On the other hand, decision-makers preferences or judgments are inevitably riddled with contradictions and confusing language. The priority derivations technique, the comparison scale, and the decision maker's preference aggregation from pairwise comparison matrices in the setting of erroneous assessments for identifying a suitable solution for decision-making problems are further issues that the AHP method might create.

As a result, AHP is not regarded as a practical strategy for making decisions under ambiguity. Thus, it is required to use the fuzzy analytic hierarchy process extension of AHP to achieve better results. Fuzzy sets could be used in pairwise comparisons to deal with uncertainty and ambiguity in situations, even though doing so might occasionally cause the eigenvector of the matrix of pairwise comparisons to become disoriented when perturbing the elements of the matrix (Saaty, 2008). In the fuzzy concept, the FAHP model enables decision-makers to more precisely describe their preferences, and its methodology is described in Table 4.8.

Fuzzy utilities or weighted sums are two ways that fuzzy or uncertain preferences might be expressed in MCDM. Fuzzy numbers include these fuzzy utilities and fuzzy

weighted sums. When there exist confidence and uncertainty preferences, a fuzzy preference is used to create the degree of preference between two options. It is an important sort of fuzzy binary relation.

Let A represent a set of alternatives A_1, A_2, …, A_n and n > 1. A fuzzy preference for the set of alternatives A is a fuzzy relation on A which is denoted by $R = (r_{ij})_{n\times n}$ which has a membership function denoted by $u_R : A \times A\ [0,1]$. Here, $u_R\ (A_i, A_j) = r_{ij}$ represents the degree of preference for alternative A_i over A_j (Aliyev & Temizkan, 2020).

To conclude, in FAHP, the pairwise comparisons are conducted in a fuzzy environment, where linguistic terms and fuzzy sets are used to represent subjective opinions and uncertainties. The use of FAHP enables the integration of both quantitative and qualitative factors in decision-making, providing a comprehensive and flexible approach to handle complex and uncertain decision problems. By combining the strengths of fuzzy logic and the AHP, FAHP offers a valuable framework for decision-makers to assess and prioritize alternatives, taking into account both objective and subjective criteria.

4.10. Technique for Order Preference by Similarity to Ideal Solution

After applying the Fuzzy Analytic Hierarchy Process (FAHP) to determine the relative weights of criteria in a decision-making process, the next step involves utilizing the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) methodology. While FAHP provides a structured approach to capturing the preferences and priorities of decision-makers, TOPSIS offers a systematic way to rank and select the best alternative among a set of options.

The proposed Neuro-Fuzzy model, discussed in the subsequent chapter, incorporates the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) as an additional Multi-Criteria Decision Making (MCDM) method. TOPSIS is utilized within the Neuro-Fuzzy framework to aid in the decision-making process.

One of the various criteria decision-making techniques, TOPSIS was first developed by Yoon and Hwang in 1981. According to the rule that the chosen alternatives must be the closest to the positive ideal solution and the farthest from the negative ideal solution from a geometrical point, TOPSIS uses the Euclidean distance to estimate how close an option is to the ideal solution (Łatuszyńska, 2014). The sum of all potential values for each attribute makes up the positive-ideal solution, whereas the sum of all possible values for each attribute makes up the negative-ideal solution. By considering the relative proximity to the positive ideal solution and the negative ideal solution, TOPSIS takes both of these distances into consideration (Zanakis, et al., 1998). Multiple priority orders could be obtained relying on the comparison of the relative distance (Rahim et al., 2017).

This technique is commonly used to conclude the decision-making process. This is because the idea is straightforward, understandable, and capable of performing efficient calculations and measuring the relative effectiveness of different decisions

(Ding & Schmidt, 2005; Kabir & Hasin, 2012). The methodology for TOPSIS calculation is described in Table 4.9. (Opricovic & Tzeng, 2004).

In conclusion, the chapter on Fuzzy Logic and Multi-Criteria Decision Making (MCDM) has shed light on the foundations and principles of fuzzy logic, emphasizing its significance in handling imprecise and uncertain information within decision-making processes. Fuzzy logic, with its ability to capture vagueness and incorporate human reasoning through fuzzy rules, offers a flexible and powerful approach to modeling complex systems.

Furthermore, the chapter explored the integration of MCDM methodologies, such as the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), within the context of fuzzy logic. These methodologies provide a structured framework for evaluating and ranking alternatives based on multiple criteria, enhancing the decision-making process in complex scenarios.

Building upon the insights gained from the fuzzy logic and MCDM foundations, the next chapter would delve into the Neuro-Fuzzy model proposed in this thesis. The Neuro-Fuzzy model represents an innovative approach that combines the adaptive and learning capabilities of neural networks with the interpretability and linguistic modeling of fuzzy logic. Through the integration of these two methodologies, the Neuro-Fuzzy model aims to enhance decision-making and improve the understanding of complex systems.

The subsequent chapter would delve into the development, architecture, and application of the Neuro-Fuzzy model, providing a detailed analysis of its implementation and its potential to address Real-world challenges. By leveraging the synergies between fuzzy logic, MCDM, and neural networks, the Neuro-Fuzzy model seeks to contribute to the advancement of intelligent decision support systems and pave the way for more effective and informed decision-making processes in various domains.

Chapter 5:
Proposed Neuro-Fuzzy Approach for Classifier Selection

5.1. Introduction

Considering the computational speed, accuracy, and complexity of nowadays models, a great number of researchers develop soft computing techniques for predictive or control applications models. Artificial Neural Networks (ANN) and fuzzy logic systems commonly use soft computation techniques, due to the combination of their advantages (Shihabudheen & Pillai, 2018). On the one side, fuzzy logic could increase the interpretability, reasoning, and inference in a machine learning model, since expert knowledge and linguistic terms are embedded. On the other side, neural networks provide the learning capability via the training process. Hence, the combination of them introduces a Neuro-Fuzzy system, which is a powerful methodology for machine learning problems (Mitra & Hayashi, 2000). Nowadays, the most upcoming hybrid type of Neuro-Fuzzy model in literature is the Deep Neuro-Fuzzy System (DNFS) as it was illustrated in Talpur et al. (2022) since it combines the tremendous learning force of Deep Learning with fuzzy reasoning.

In continuance of chapter 4 in which Fuzzy Inference Systems (FIS) are analyzed, it was notable that the FIS outputs depended on the membership function with its parameters, rules, and defuzzification method which would be applied (Nauck & Kruse 1993). The membership functions chosen have a substantial impact on an application's success. There is, however, no automated way to develop membership features. They are mostly accomplished through trial and error or by human expertise. It is well known that rule acquisition has been seen as a bottleneck for the adoption of rule-based systems and continues to be so today. In the majority of currently used applications, fuzzy rules are frequently created by a subject-matter expert for systems with limited inputs. Because the number of feasible rules for the system exponentially rises with an increase in inputs, outputs, and linguistic factors, experts find it difficult to specify a full set of rules and related membership functions for acceptable system performance.

Plentiful researchers have tried to address this issue, using various methodologies, such as the evolutionary approach. Evolutionary learning algorithms are a quite promising solution in terms of parameter optimization, in case there is no a priori information about the membership function and the rule base is available. However, the main disadvantage of evolutionary algorithms in an evolutionary fuzzy system is the time-consuming process and its performance is essentially dependent on the population

size and the number of generations necessary for a solution to be robust for certain issues. (Siddique, 2013).

In the literature, there are various available Neuro-Fuzzy models applied to different industries. For example, Bendre et al. (2020) used fuzzy logic rules with the neural-based action recognition model to rate the intensity of human action as intense or mild. In Ozkan (2020), a concurrent neuro-fuzzy system (CNFS) was coupled with Fuzzy logic to evaluate whether a basketball team was the favorite, with an upper aim to predict basketball game results. Furthermore, Fuzzy Inductive Reasoning (FIR) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) were applied for modeling burned areas of forests in Portugal by Nebot and Mugica (2021).

This chapter offers a concise exploration of the relationship between Neural Networks and fuzziness through fuzzy neurons. It introduces popular types of Neuro-Fuzzy models and provides a detailed description of a proposed Neuro-Fuzzy model. By combining the strengths of fuzzy logic and neural networks, these models present promising opportunities for addressing complex problems and making accurate predictions. This chapter sets the stage for further advancements in the field of hybrid intelligent systems.

5.2. Fuzzy Neurons

Neurons in Neural Networks were addressed in Chapter 3, comprised of processing units that handle numeric inputs and outputs. There are several instances in real-world applications where such numerical measures are either not accessible or are imprecise and noisy. As a result, some scientists concluded that neurons ought to be able to handle these real-world occurrences and tried to include them in a fuzzy neuron. A fuzzy neuron has a similar fundamental structure to an artificial neuron, with the exception that inputs, processing, and outputs are determined using fuzzy logic. As a result, several fuzzy neurons have been created and may be read about in the literature.

In a simple Fuzzy-Neuro, there are two inputs, and its output is calculated as a weighted sum of them,

$$y = f(w_1 x_1 + w_2 x_2)$$

where y, x, and w are the output, inputs, and weights, respectively, whereas the function f could be of any activation function type as Linear, Sigmoid, or other. Notably, regular neural networks apply mathematical operations as addition, subtraction, or activation functions like Sigmoid or ReLu. However, hybrid neural networks are defined by fuzzy operators as T-Norm or S-Norm are applied. Specifically, a fuzzy neuron with T-Norm fuzzy operator which represents an AND Fuzzy-Neuron is defined by (Singh & Lone, 2020),

$$y = T(S(w_1 x_1) + S(w_2 x_2) + \ldots + S(w_n x_n))$$

whereas, T-Co-Norm operation which implies OR Fuzzy-Neuron is defined as,

$$y = S(T(w_1 x_1) + T(w_2 x_2) + \ldots + T(w_n x_n))$$

Implication-OR Neuron.

In this type of neuron an implication operator between the input x and the weight w and the Triangular Co-norm operator on the output are applied. That is,

$$y = S(w_1 \leftarrow x_1) + (w_2 \leftarrow x_2) + \ldots + (w_n \leftarrow x_n))$$

Kwan and Cai's Fuzzy Neurons.

It has a complex structure because all products of each input value and weight are aggregated to a single input. That means the equation,

$$s = f(z - \theta)$$

where f is a selected activation function, θ is an activation threshold and z in K&C Max neurons is represented by,

$$z = max(w_1x_1 + \ldots + w_nx_n)$$

or z in K&C Min neurons represented by,

$$z = min(w_1x_1 + \ldots + w_nx_n)$$

Figure 5.1 displayed a graphical representation of the K&C neuron.

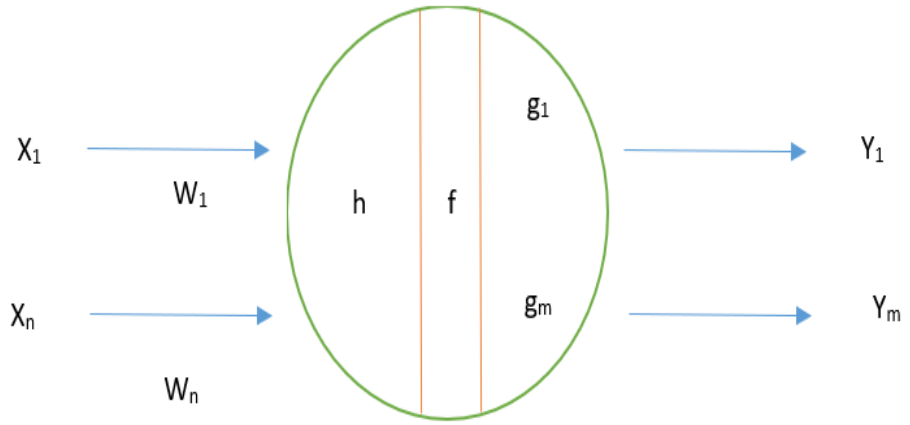


Figure 5.1. K&C Neuron (Singh & Lone, 2020).

5.3. Combinations of Neural Networks and Fuzzy Logic

Cooperative, concurrent, and hybrid Neuro-Fuzzy models are three different types of neural network and fuzzy system combinations that might be used to tune or train a fuzzy system.

In cooperative neuro-fuzzy systems, fuzzy systems and neural networks operate independently, since they use a neural network to determine specific fuzzy system parameters while the fuzzy system is operating. A cooperative model could be thought

of as a preprocessor in which the artificial learning mechanism extracts the membership functions or fuzzy rules from the training data for the Fuzzy Inference System (FIS). When the FIS has been determined, ANN has no function. Algorithms for fuzzy clustering often establish the rules. The fuzzy membership functions from the training data are approximated using ANN. Figure 5.2 visualized the cooperative concept (Rana & Prasad, 2010).

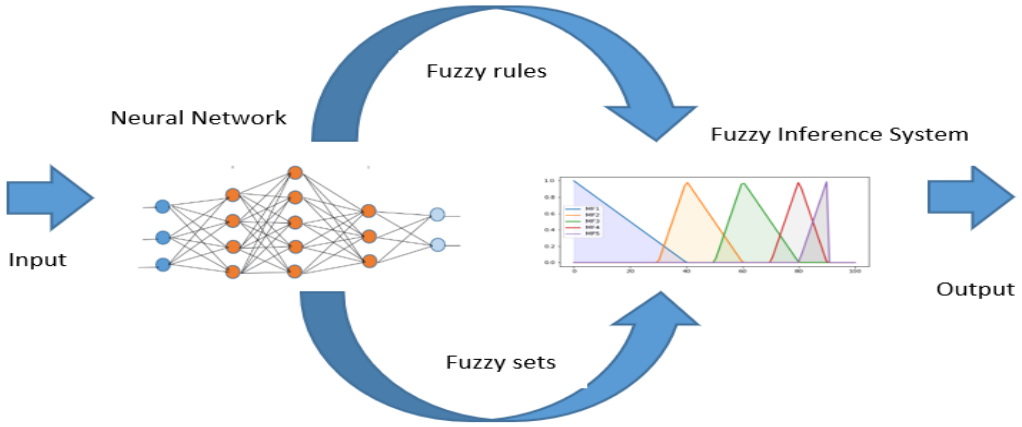


Figure 5.2. Cooperative Neuro-Fuzzy model (Rana & Prasad, 2010).

In case of unmeasured input variables are introduced, a Concurrent Neuro-Fuzzy system would be applied. In the concurrent model, the neural network and the fuzzy systems continually collaborate to identify the necessary parameters, particularly when the controller's input variables could not be monitored directly. This combination merely helps to increase the system's overall performance rather than optimizing the fuzzy system. The fuzzy system is unaffected throughout the period when the neural network is learning. In Figure 5.3 a graphical representation of a concurrent Neuro-Fuzzy model is depicted.

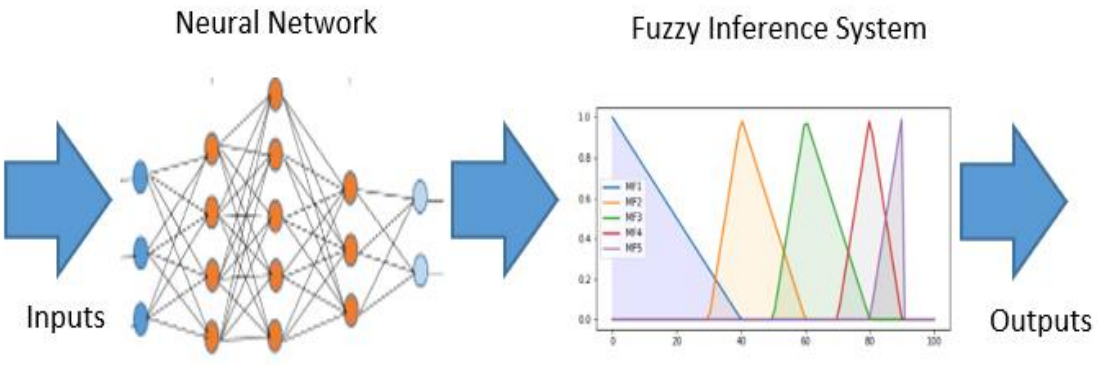


Figure 5.3. Concurrent Neuro-Fuzzy model (Rana & Prasad, 2010).

According to Nauck (1997), a hybrid Neuro-fuzzy system is a fuzzy system that processes input and output patterns to learn its parameters (fuzzy sets and fuzzy rules) using a gradient-based learning algorithm or a system that's motivated by the theory of neural networks. These systems have a common knowledge representation and data

structure. Due to the fact that hybrid Neuro-fuzzy systems might be developed in various methods and consist of a relatively new study topic, each researcher has established his unique models. Although there are fundamental distinctions between these models, they are fundamentally similar. With successful applications in a variety of fields such as process control, engineering design, financial trading, credit evaluation, medical diagnosis, and cognitive simulation, the usage of intelligent hybrid systems is expanding quickly. The development of a fuzzy neuron, inspired by biological neurons, is the first step in the computational process for fuzzy neural systems. Next, models of synaptic connections were constructed that incorporate fuzziness into neural networks, and eventually, the method of adjusting the synaptic weights led to the creation of learning algorithms. Well-known hybrid NF models are FALCON (Lin & Lee,1991), ANFIS (Jang, 1992), GARIC (Bherenji & Khedkar, 1992), NEFCON (Nauck & Kruse, 1997), FUN (Sulzberger et al., 1993), SONFIN (Juang & Lin, 1998), FINEST (Tano et al., 1996), EFuNN, etc.

With a target for the function of hybrid NF models to be more comprehensive, Adaptive Neuro-Fuzzy Inference System (ANFIS) would be analyzed since it is the most commonly represented and used Neuro-Fuzzy model. An artificial neural network that is based on the Takagi-Sugeno fuzzy inference system is known as an adaptive neuro-fuzzy inference system (ANFIS). It appears the ability to combine the advantages of neural networks and fuzzy logic in a single framework since it incorporates both of these concepts. Its inference mechanism resembles a collection of IF-THEN fuzzy rules with the capacity to learn and approximate nonlinear functions. As a result, ANFIS is regarded as a global estimator. The ANFIS architecture of the Takagi-Sugeno Fuzzy model, Tsukamoto Fuzzy model, and Mamdani Fuzzy model are described in Appendix C.

5.4. Neuro-Fuzzy proposed model

Within this thesis, an innovative Neuro-Fuzzy model is introduced with the objective of effectively selecting the optimal neural network for credit risk assessment. The proposed optimization method, devised to integrate expert knowledge with neural network predictions in the context of credit risk assessment, comprises two concurrent stages, followed by a third step that combines and consolidates the outcomes. The fundamental inputs of this algorithm encompass the credit datasets (refer to Section 6.2), the tuning techniques (see Section 6.4), and expert knowledge. The resulting output of the algorithm is the identification of the most recommended model based on the insights and perspectives provided by domain experts. The visual representation in Figure 5.4 elucidates the parallel phases, the aggregation phase, and the intermediary stages encompassing this comprehensive process.

The initial phase of this process involves the creation of three artificial neural networks with varying topologies, specifically employing 1, 2, and 3 layers. To commence, the datasets undergo a comprehensive cleaning and preparation procedure, which includes eliminating duplicate rows and columns, handling missing values and outliers, ensuring a balanced quantity for the target variable, and encoding categorical

variables. In the subsequent stage, four tuning techniques are applied to each of these three models, and the model that demonstrates superior performance across multiple evaluation metrics is identified as the optimal choice. The outcome of this phase is a tabulated representation of the performance metrics for the three neural network models, serving as input for the subsequent phase.

In the second phase, the human aspect is incorporated through the integration of expert knowledge. The Fuzzy Analytic Hierarchy Process (FAHP) is employed to capture the preferences of four distinct experts, each assigning varying levels of significance to the evaluation metrics for model selection. The result of this phase involves the generation of normalized non-fuzzy scores for each metric, which would subsequently be utilized as inputs in the subsequent phase.

In the final phase, the performance metrics of the three neural networks and the weights derived from the FAHP process are combined as inputs in a selection model employing the Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS) approach. The ultimate outcome of this comprehensive process is the identification of the most robust model for credit risk assessment based on expert viewpoints and preferences.

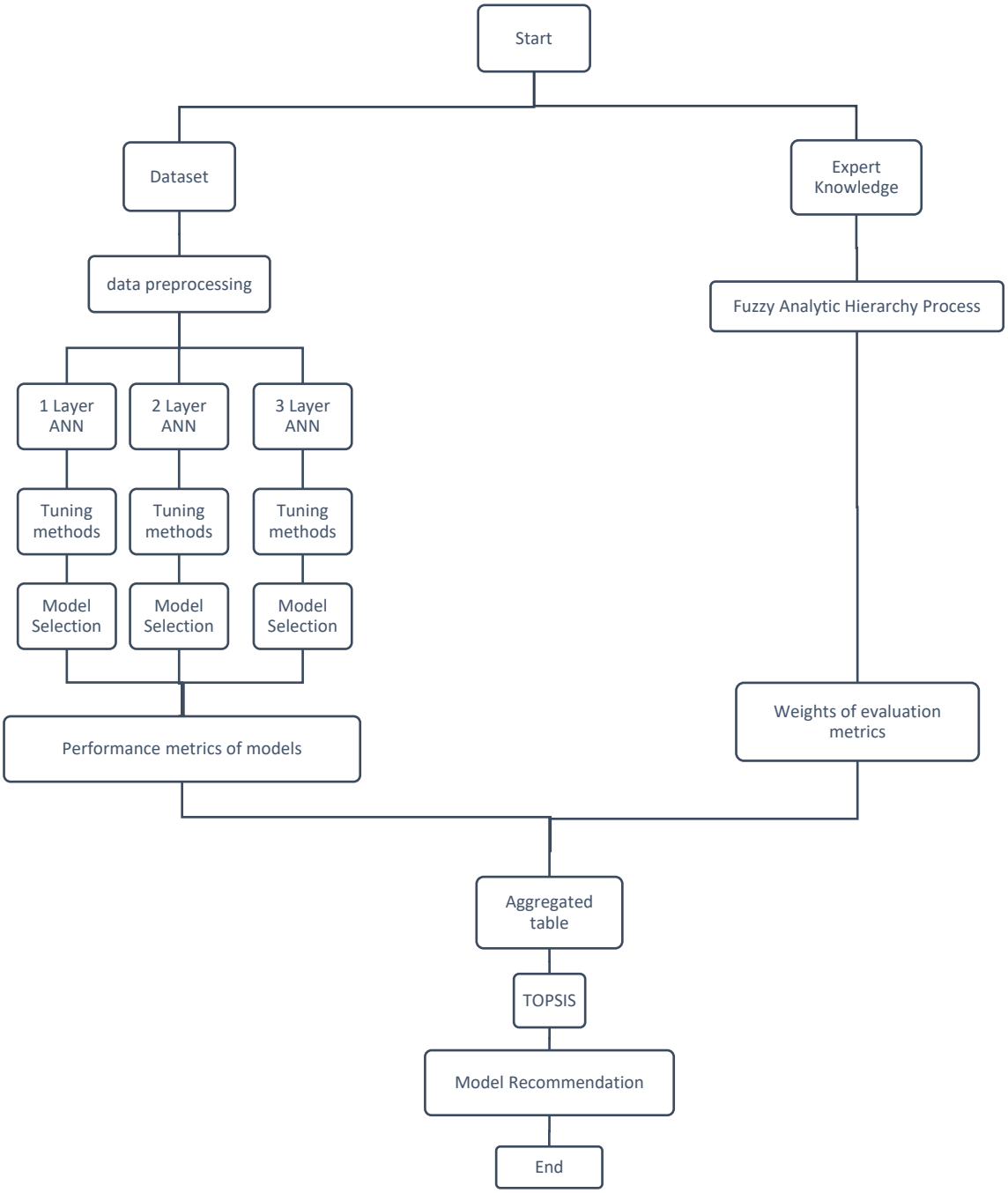


Figure 5.4. Flowchart of the recommended model (Author, 2023).

In conclusion, this chapter initially presented the fundamental Neuro-Fuzzy concepts and then introduced a comprehensive methodology for selecting the optimal

neural network model for credit risk assessment. The multi-phase approach involved the creation of three neural network models with different topologies, followed by the application of tuning techniques to evaluate their performance across various metrics. The integration of expert knowledge through the Fuzzy Analytic Hierarchy Process (FAHP) further enhanced the decision-making process by considering diverse perspectives and ranking preferences. The Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS) was then employed to synthesize the performance metrics and expert weights, resulting in the identification of the most robust model for credit risk assessment. The methodology presented in this chapter provides a systematic and comprehensive framework for model selection, incorporating both technical evaluations and expert insights. The next chapter would present the computational experiments conducted to validate the effectiveness of the proposed methodology and discuss the results obtained.

Chapter 6:
Computational Experiments and Results

This chapter focuses on the implementation of the proposed Neuro-Fuzzy model across three different datasets. It provides a detailed description of each stage involved in the implementation process. Subsequently, the results obtained from the conducted experiments are presented. The chapter then proceeds to discuss and analyze these results in order to gain insights into the performance and effectiveness of the Neuro-Fuzzy model. By examining the outcomes and engaging in a comprehensive discussion, the chapter aims to evaluate the model's capabilities and its applicability to the given datasets.

6.1. Experimental setup

The proposed model -in Chapter 5- was developed in the Python programming language (3.9.12) and executed on a HP personal computer with an Intel(R) CoreTM i5-7200U CPU, clocked at 2.5 GHz, a microprocessor with 8 GB of RAM memory under the operating system Microsoft Windows 10 professional.

6.2. Datasets

It is worth mentioning that this thesis aims to describe a framework developed using machine learning and fuzzy logic approaches, emphasizing its methodological components. Despite the possibility of selecting from a variety of datasets in the literature, it is notable that for the scope of this study, three datasets derived from different fields are used, such as academic (Australian credit approval), competitional (Credit risk dataset of Kaggle), and Real-world (Real-world).

The evaluation of the proposed algorithms involved the utilization of three distinct datasets: the Australian (AUS) credit approval dataset obtained from the UCI Machine Learning Repository, the Credit risk dataset acquired from the Kaggle website, and a Real-world dataset sourced from an anonymous financial institute, as provided by the supervisor professor. These datasets were employed to assess the performance of the proposed algorithms in the context of credit risk analysis. The features of the three datasets are presented in Table 6.1, Table 6.2, and Table 6.3, respectively.

The first dataset contains instances for 690 past credit applicants on six numerical and eight categorical features. The desirable class (i.e., the value of the dependent variable is equal to 0) comprises 383 instances, whereas 307 are associated

with the undesirable (i.e., the value of the dependent variable is equal to 1). The second dataset consists of 32.581 instances, seven numerical and four categorical features. Well-payers are 25.327 instances (i.e., the value of the dependent variable is equal to 0), whereas 7.089 are associated with defaulters (i.e., the value of the dependent variable is equal to 1). The third dataset contains 50.000 instances with 55 numerical features. The target variable is equally divided into 25.000 per category.

6.3. Data Preprocessing

Regarding the first stage of the proposed model, initially the datasets underwent extensive cleaning and correction, considering their unique characteristics and requirements. To ensure data integrity, duplicate rows, and columns, constant and quasi-constant variables, were eliminated from the datasets. A random selection approach was employed for imputation to address missing values, ensuring that every variable within the dataset would preserve its previous distribution. 1st-dimensional outliers were detected by Tukey's fence technique (Tukey, 1977) and handled by winsorize method using as the upper value the 99th quantile of variable's values (Horn & Kafadar, 2006). To identify multi-dimensional outliers, a one-class SVM (Support Vector Machine) was utilized. Subsequently, 10% of these outliers were removed from the datasets (Manevitz & Yousef, 2001).

A label encoder was applied, assigning ascending numerical values to each category (e.g., 1, 2, etc.) to handle categorical variables. This encoding scheme facilitated the representation of categorical variables in a numerical format within the dataset. In classification problems, it is essential to ensure that the classes of the target variable are balanced. This balance is crucial for generating robust assessments and reliable model performance evaluations. In this manner, if there were a different count of instances per class, Synthetic Minority Over-sampling Technique (SMOTE) would be used (Chawla et al., 2002).

In the final stage, each dataset was partitioned into training, validation, and testing sets, utilizing proportions of 64%, 16%, and 20% of instances, respectively. This division of the data enabled the implementation of effective training, validation, and evaluation procedures for the proposed models. Training data would be used to train the model, validation data would enhance the training process, whereas testing data would assess the final model with respect to its consistency.

6.4. Neural Networks parameter selection

In continuance of the first stage, Artificial Neural Networks (ANN) for clients' credit risk evaluation, as described in Chapter 3, were used. This study examined three distinct topologies of Artificial Neural Networks (ANNs) characterized by different layer configurations. These topologies were evaluated using a consistent methodological approach.

To be more specific, ANN was constructed using Keras.TensorFlow python library and the best hyperparameter of each model were retrieved from four different tuning techniques using the Optuna library. As the loss function was concerned, binary cross entropy with evaluation metric AUC was selected, and the number of units, alpha of LeakyReLU, dropout, learning rate, batch size, kernel regularizer, and bias regularizer were tuned. A pipeline consisted of an ANN model and normalization scaler via the z-score method, in which values of dataset columns were standardized using mean 0 and standard deviation 1.

The aforementioned hyperparameters, including the validation data and early stopping mechanism, were incorporated into the pipeline to mitigate the risk of model overfitting. The training set was divided into three equal portions to produce an average result using a stratified k-fold cross-validation with k = 3, one subset used as a validation set and the other two as training sets. When the training data is divided into folds, cross-validation is referred to as stratified since this ensures that each fold contains an equal amount of the two types of class labels (i.e., well-payers/defaulters). Furthermore, a pruner for every trial is used because it allows Optuna to quickly identify and eliminate configurations that are unlikely to improve performance, allowing it to allocate more computational resources to more promising configurations.

Tuning techniques of Bayesian optimization (Tree-structured Parzen Estimator), Random Sampling, Genetic Algorithm, and Quasi-Monte Carlo (see section 3.7) were applied independently for 100 trials. These trials aimed to optimize the model's performance based on the pre-defined objective function and initial model. In addition, this process is aided by the Median Pruner, which serves as an "early stopping" algorithm. The Median Pruner terminates unfavorable trials based on the comparison of their intermediate outcomes with the median value. This approach efficiently identifies and terminates trials that were unlikely to yield satisfactory results, contributing to the overall efficiency of the tuning process. During the trials, if the best intermediate outcome of a particular trial is lower than the median of the intermediate results obtained from previous trials at the same phase, the trial was pruned. As a result, after 100 trials the best hyperparameters were produced and used for the best ANN model. The outcome of this phase retrieved evaluation metrics of Accuracy, AUC, Precision, Recall, f1-score, and H-measure for every tuning method.

6.5. Fuzzy - AHP

According to the second stage of the proposed model, the Fuzzy-AHP method applied to model decision makers' pair-wise comparison judgments depends on the importance of evaluation metrics. For the scope of this study, four random decision makers support that four different metrics each time were the most important in the model selection decision. The decision-makers inserted their preferences with respect to evaluation metrics comparison in a quadratic matrix with a scale of 1 to 9.

Furthermore, the triangular membership function used with intervals and linguistic variables is represented in Table 6.4. The primary objective of this process was

to compute non-fuzzy normalized values for the metrics based on experts' opinions. These normalized values would then be utilized as weights in the TOPSIS algorithm. By incorporating experts' judgments, the process aimed to assign appropriate weights to the metrics, enabling an effective assessment and ranking of alternatives using the TOPSIS technique.

The Fuzzy Analytic Hierarchy Process (AHP) algorithm described in section 4.9 was implemented using Python programming language to generate non-fuzzy normalized values. This implementation allowed for the practical application of the algorithm, enabling consistent and interpretable decision-making based on reliable results. Tables 6.5 to 6.8 present the pair-wise comparison matrices related to four decision makers. Consistency Indices (CI) and Ratios (CR) were calculated to assess the validation models and verify that the consistency ratios remained below 10%. The objective of this analysis was to ensure that the pair-wise comparison matrices were considered consistent. By examining the CR values, it was possible to evaluate the level of consistency and determine if the matrices met the desired criterion for reliability and coherence in the decision-making process.

6.6. TOPSIS

The outputs of the evaluation metrics from the three Artificial Neural Network (ANN) models were obtained in the initial phase. Additionally, non-fuzzy normalized weight values derived from the sequent one using Fuzzy AHP were retrieved. These outputs and weight values are utilized as inputs to the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) algorithm.

The TOPSIS algorithm, incorporating both the evaluation metrics and weight values, facilitated the selection of the most suitable model based on the expert's knowledge and preferences. By combining the results from both phases, the decision-making process was enhanced, allowing for an informed model selection based on the expertise of the domain experts.

To streamline the analysis, it was essential to identify the dominant metrics obtained from the tuning methods applied to the ANN model. These dominant metrics were selected for each dataset individually and subsequently merged into the respective datasets. By incorporating the most influential metrics, the datasets were tailored to include the crucial performance indicators, facilitating further analysis and decision-making processes. Following the methodology of Table 4.9 would be proposed as an ANN model depending on the decision-makers’ aspect.

6.7. Results

Following the implementation of the aforementioned sections, section 6.4 provides a comprehensive analysis of various neural network topologies in conjunction with the evaluation metric outcomes obtained from four distinct tuning methods. Within this section, tabular representations are employed to illustrate the performance of each

approach, with the tuning methods (T-str Parzen Est., Random Sampler, Genetic Algorithm, Quasi-Monte Carlo) delineated as columns and the evaluation metrics (Accuracy, Precision, Recall, F1-score, AUC, H-measure) delineated as rows. These tables serve as a valuable resource for evaluating the effectiveness of each neural network topology in relation to the employed tuning methodologies and the associated evaluation criteria

Results from the Australian credit approval dataset are shown in Tables 6.9 to 6.11, revealing the superior performance of Quasi-Monte Carlo, Tree-structured Parzen Estimator, and Genetic Algorithm for 1 Hidden Layer Neural Network (HL NN), 2 HL NN, and 3 HL NN, respectively. Similarly, Tables 6.15 to 6.17 display results from the Credit risk dataset of Kaggle, indicating the dominance of Genetic Algorithm, Quasi-Monte Carlo, and Tree-structured Parzen Estimator for 1 HL NN, 2 HL NN, and 3 HL NN, respectively. Furthermore, Tables 6.21 to 6.23 describe the outcomes of the Real-world dataset, highlighting the higher metric values achieved by the Tree-structured Parzen Estimator in each NN model.

To construct tables containing the best hyperparameters for each neural network topology, the optimal hyperparameter settings were identified for every NN model in the three datasets. These tables present the tuned hyperparameters (Units, Alpha of LeakyRelu, Dropout, Learning Rate, Batch Size, Kernel Regularizers, Bias Regularizers, and Time) as indices and the aforementioned tuning methods as columns. Specifically, Tables 6.12 to 6.14, 6.18 to 6.20, and 6.24 to 6.26 showcase the hyperparameters of NN models in the Australian credit approval, the Credit risk of Kaggle, and Real-world datasets, respectively. Importantly, the tuning process successfully prevented model overfitting by imposing value intervals for each hyperparameter, as evidenced by the absence of suspicious hyperparameter values in the tables

During the second phase of the investigation, input and opinions were solicited from four decision-makers. Each decision-maker independently advocated for AUC, Accuracy, F1 score, or H-measure as the most appropriate evaluation metric for model selection. Consequently, the combined approach outlined in Section 6.5, which incorporates Section 4.9 (Fuzzy AHP), resulted in the generation of four FAHP models. The outputs of these models are presented in Tables 6.27 to 6.30 and visually represented in Figure 6.1. Figure 6.1 illustrates the fuzzy scores assigned to each metric, while Tables 6.27 to 6.30 display the FAHP process outputs (Lower, Medium, Upper, Expected Value, Standard Deviation) in columns and the evaluation metrics (AUC, Accuracy, F1 score, H-measure) in rows. It is noteworthy that the Consistency Ratios (CR) for the four pairwise matrices were calculated to be 0.0853, 0.0975, 0.097, and 0.0997, respectively. These CR values indicate a high degree of consistency among the pairwise matrices utilized in the decision-making process. The low CR values imply that the judgments and comparisons made by the decision-makers were consistent and reliable. This validation serves to bolster the credibility and robustness of the decision-making process grounded in the pairwise matrices.

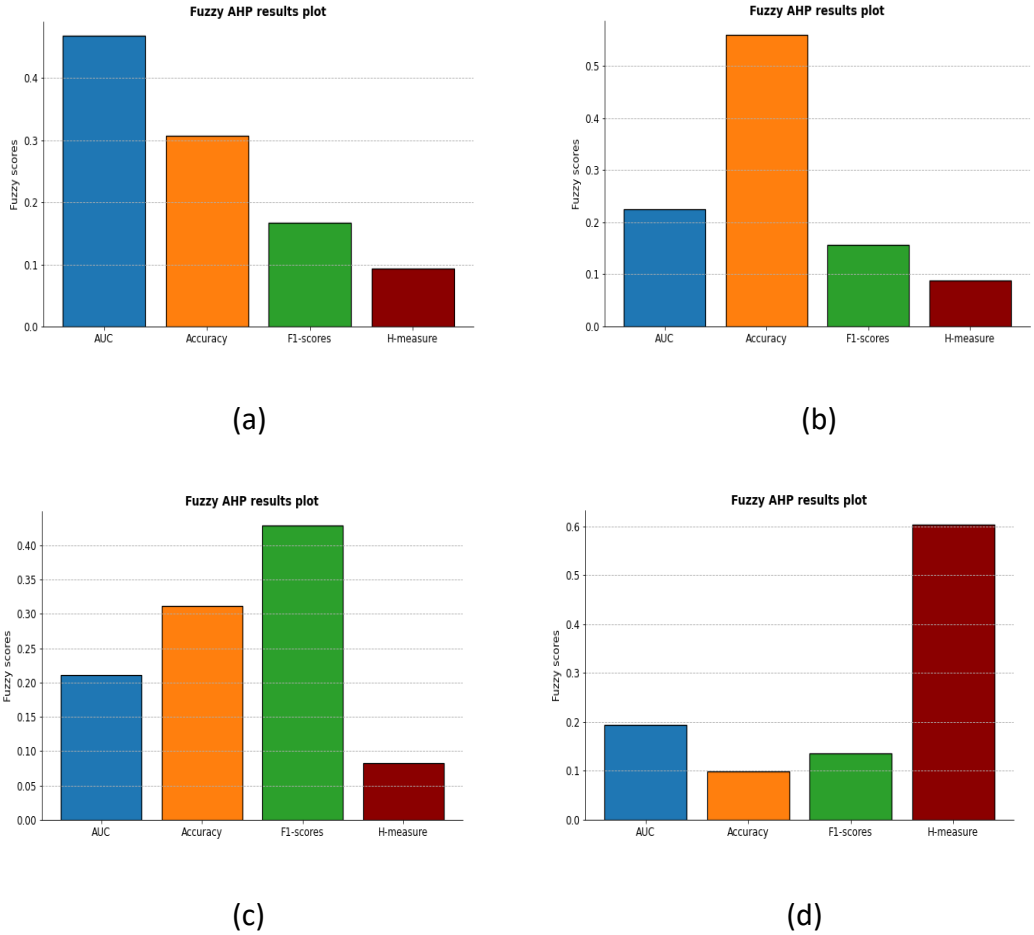


Figure 6.1. Fuzzy AHP results of 4 (a-d) decision-makers.

In the final phase of the study, the TOPSIS method, as detailed in sections 4.10 and 6.6, was employed to derive results based on datasets and expert opinions. Furthermore, a sensitivity analysis of metric weights was performed. To provide more granular information, Tables 6.31 to 6.34 present the TOPSIS results specifically for the Australian credit approval dataset, utilizing the weights obtained from the FAHP models in conjunction with the evaluation metrics from the NN models. Additionally, Figures 6.2 visually represent the TOPSIS results. The findings indicate that the 3 Hidden Layer NN model consistently emerged as the most favorable choice across all scenarios, while the 2 Hidden Layer NN model should be disregarded.

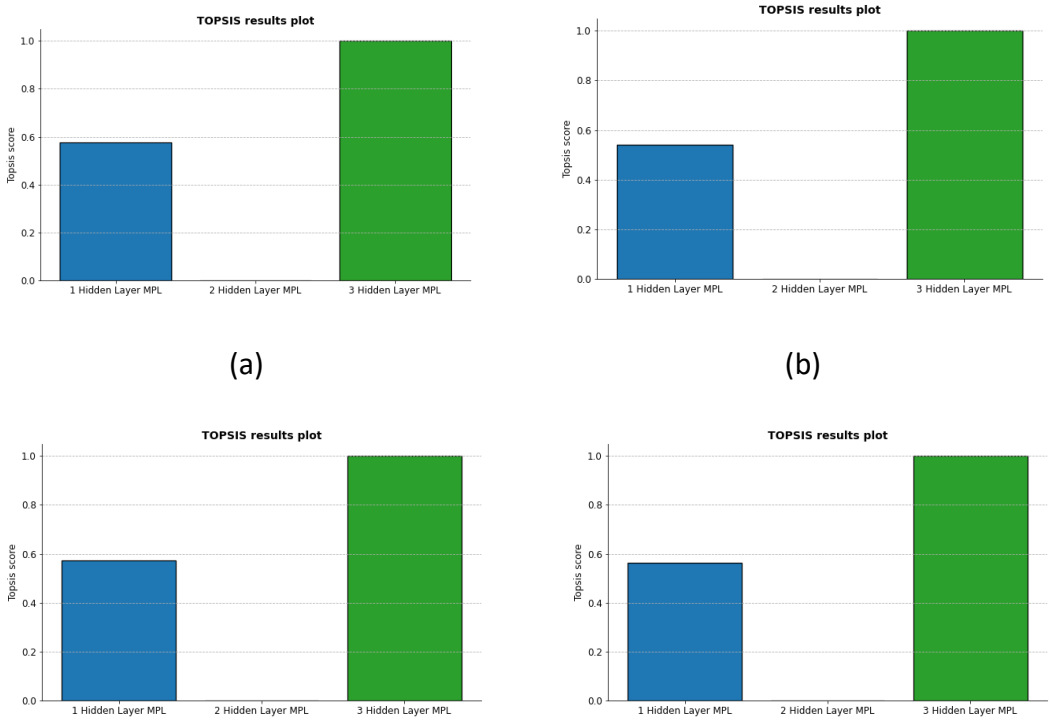
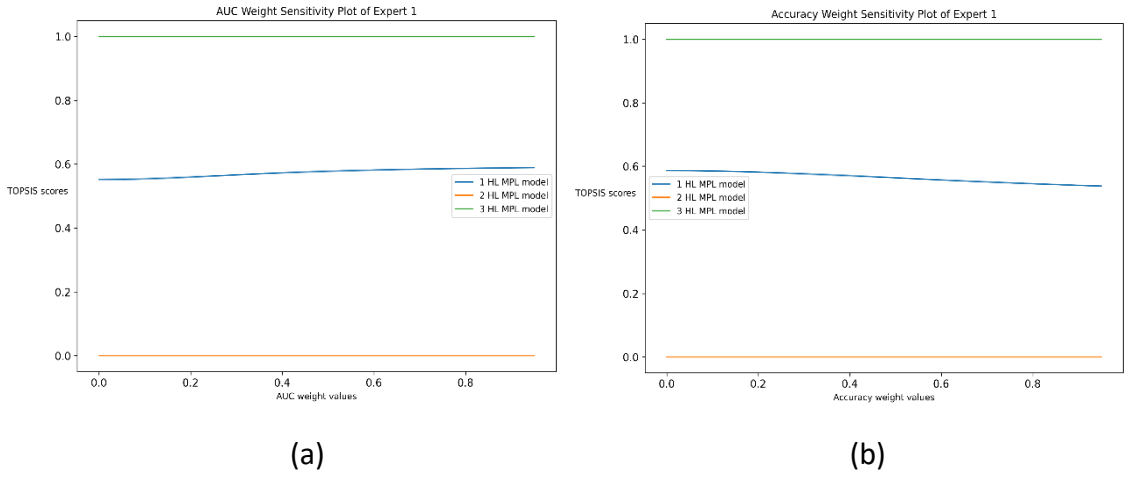
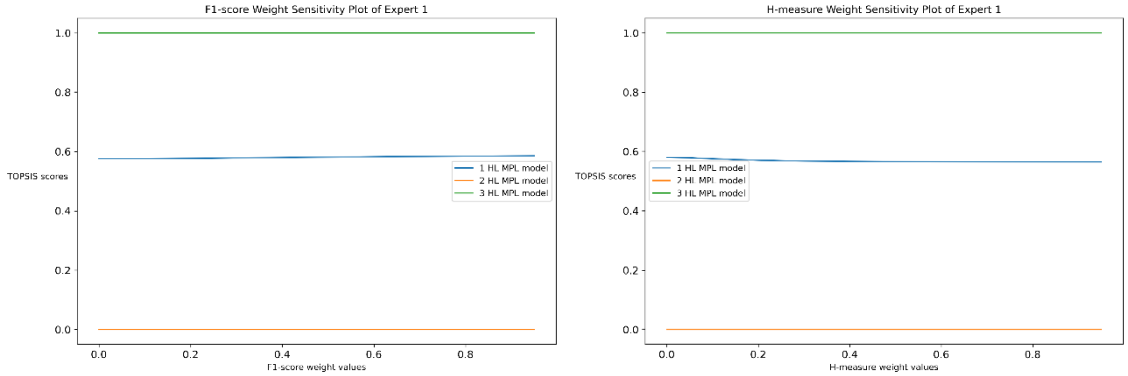


Figure 6.2. TOPSIS results of 4 (a-d) decision-makers in the Australian credit approval dataset.

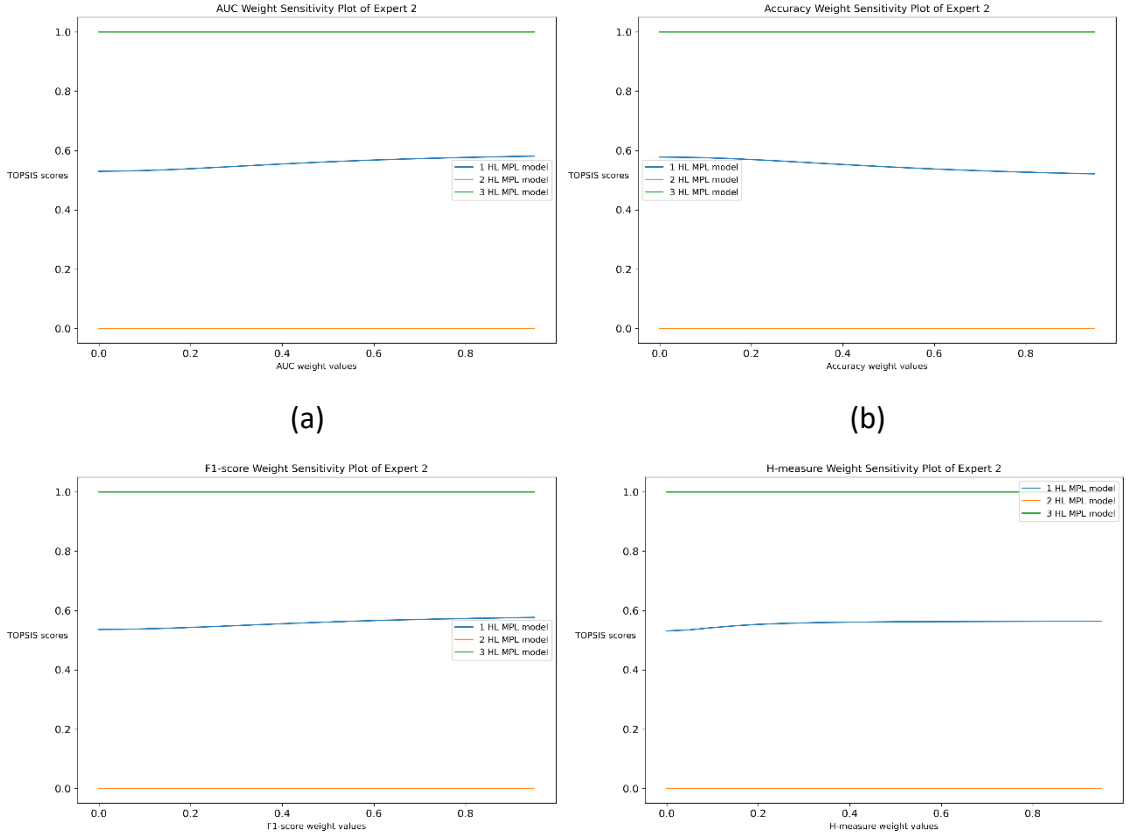
Concerning the experts' perspective, Figures 6.3 to 6.6 provide a sensitivity analysis of the evaluation metrics. These figures effectively illustrate that the 3 Hidden Layer NN model consistently outperforms the other models across various weight values within a specific interval. The 1 Hidden Layer NN model follows suit, albeit with significantly lower scores in comparison to the top-performing model. Conversely, there is no justification to select the 2 Hidden Layer NN model, as its scores converge towards zero, indicating its inferior performance.





(c) (d)

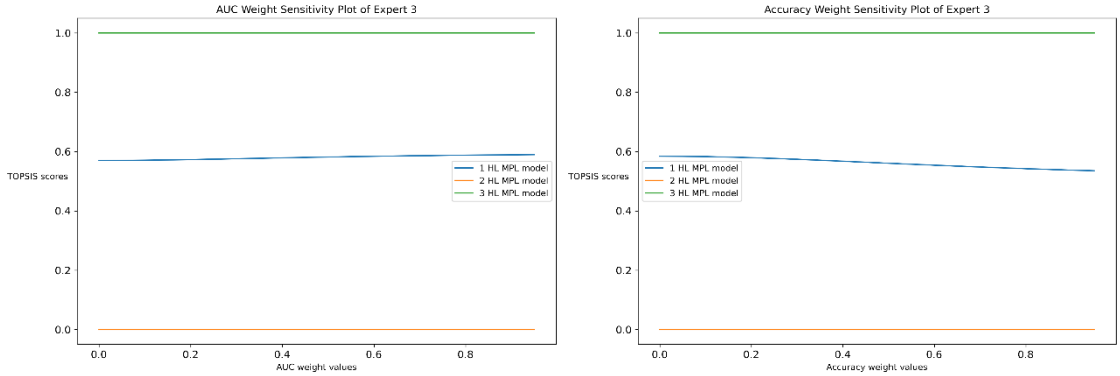
Figure 6.3. First expert's opinion sensitivity analysis of metric (a-d) in Australian credit approval dataset.



(a) (b)

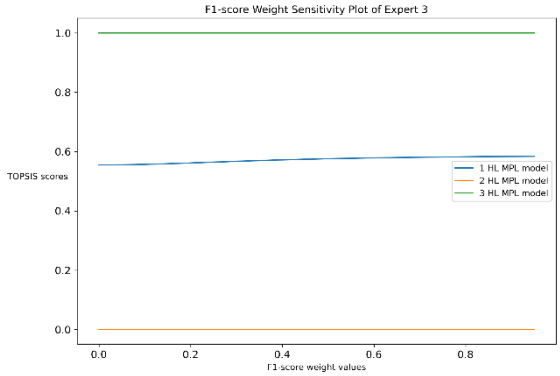
(c) (d)

Figure 6.4. Second expert's opinion sensitivity analysis of metric (a-d) in Australian credit approval dataset.

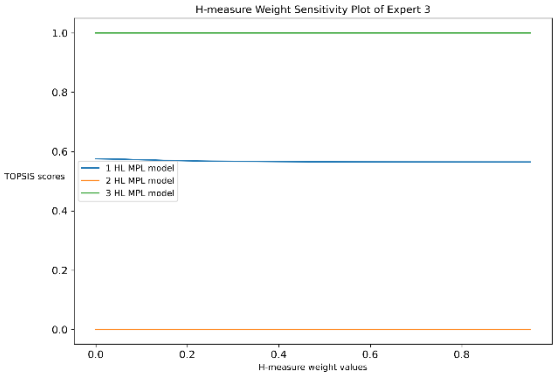


(a)

(b)

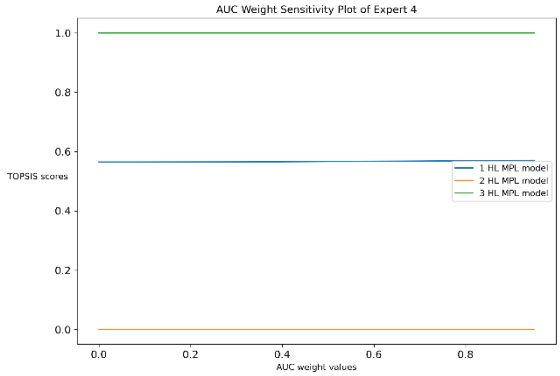


(c)

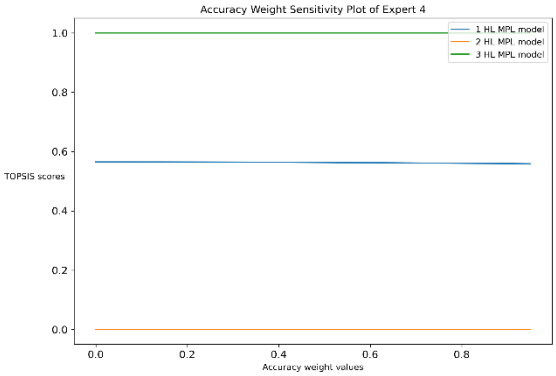


(d)

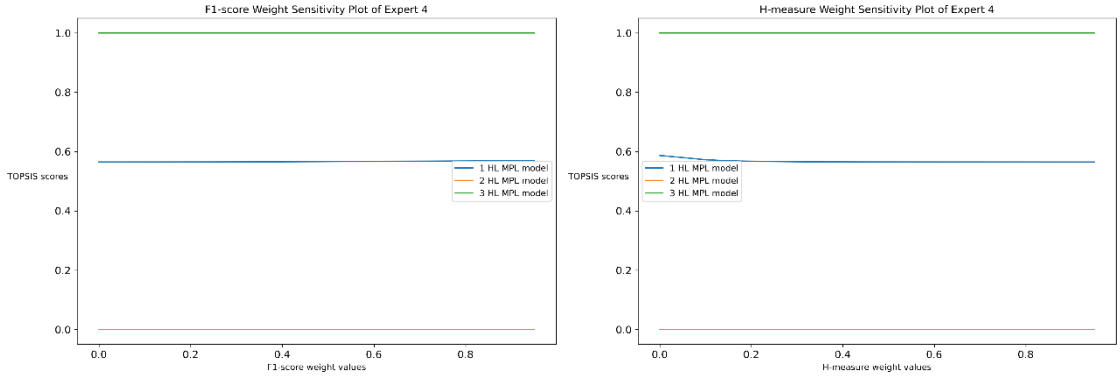
Figure 6.5. Third expert's opinion sensitivity analysis of metric (a-d) in Australian credit approval dataset.



(a)



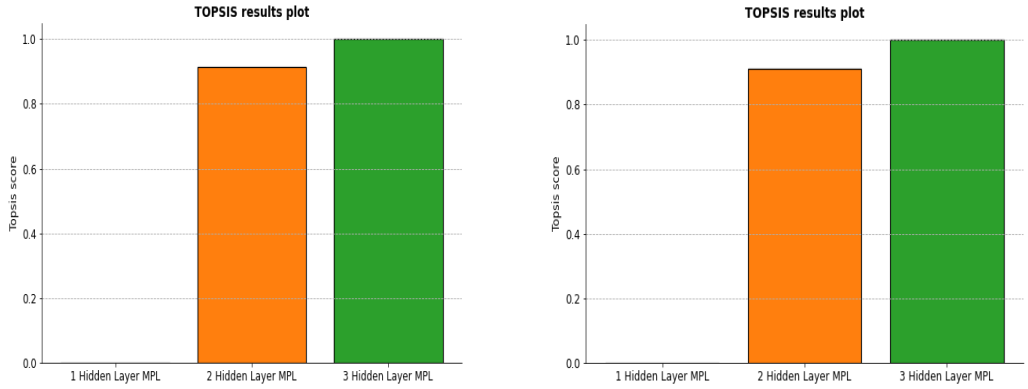
(b)



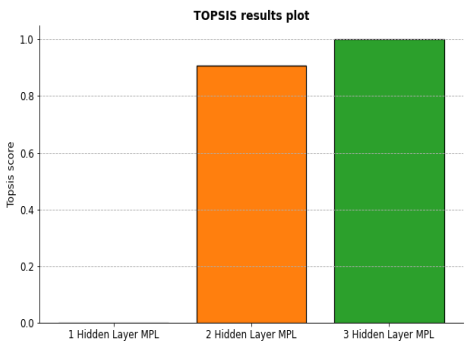
(c) (d)

Figure 6.6. Fourth expert's opinion sensitivity analysis of metric (a-d) in Australian credit approval dataset.

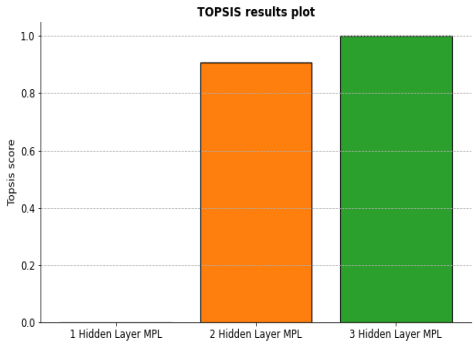
Similarly, Tables 6.35 to 6.38 showcase the TOPSIS results of the Credit risk dataset of Kaggle for each scenario, while Figure 6.7 visually presents the TOPSIS results. The figures indicate that the 3 Hidden Layer NN model is the optimal choice, although the margin between the 3 Hidden Layer NN model and the 2 Hidden Layer NN model is relatively small.



(a) (b)



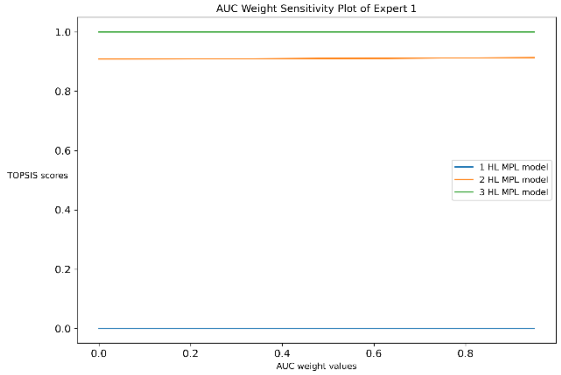
(c)



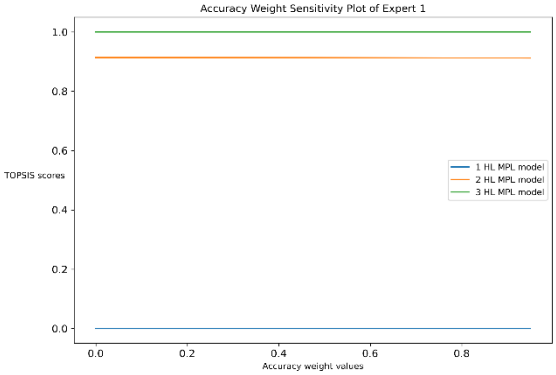
(d)

Figure 6.7. TOPSIS results of four (a-d) decision-makers in the Credit risk dataset of Kaggle.

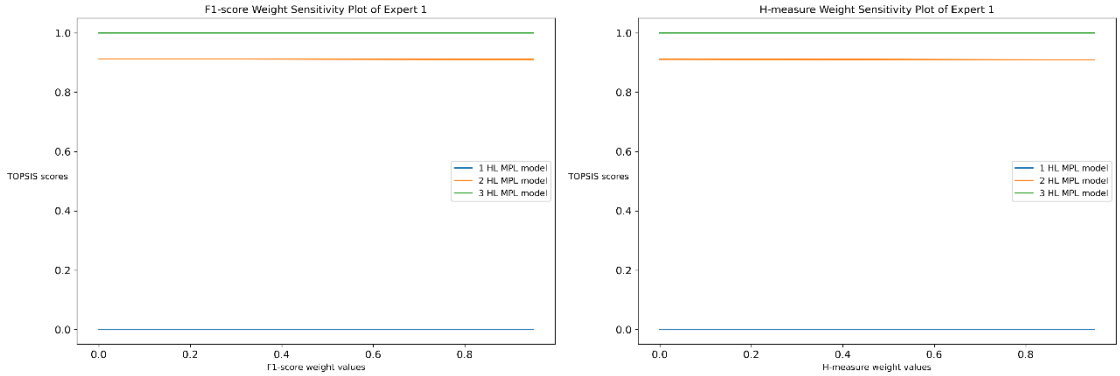
In addition, Figures 6.8 to 6.11 display the results of the sensitivity analysis of evaluation metrics for the Credit risk dataset of Kaggle. It is worth highlighting that the 3 Hidden Layer NN model consistently emerges as the dominant choice across the entire range of weight values within the analysis interval. The 2 Hidden Layer NN model closely follows as the second option, with its scores being close to those of the dominant model. In contrast, the 1 Hidden Layer NN model performs as a subordinate option, with its scores approaching zero.



(a)

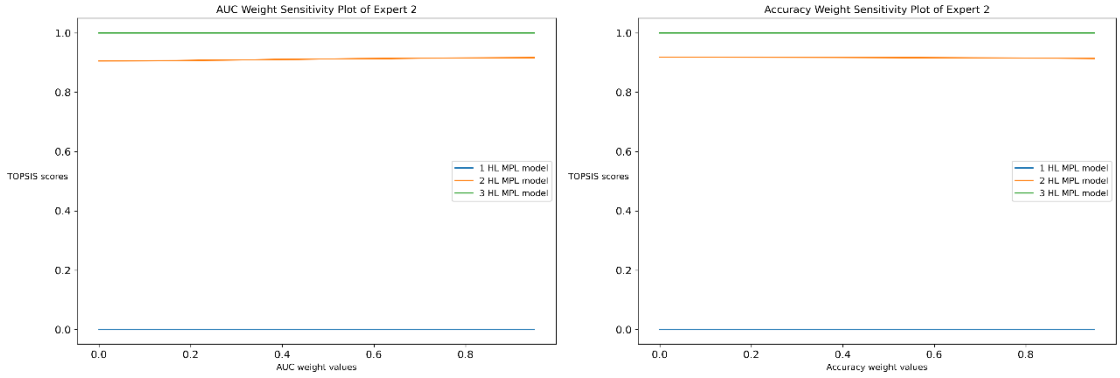


(b)

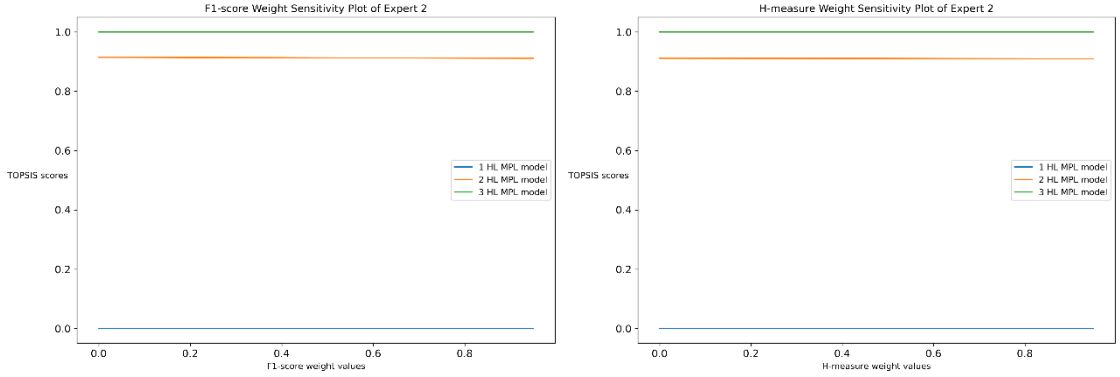


(c) (d)

Figure 6.8. First expert's opinion sensitivity analysis of metric (a-d) in the Credit risk dataset of Kaggle.

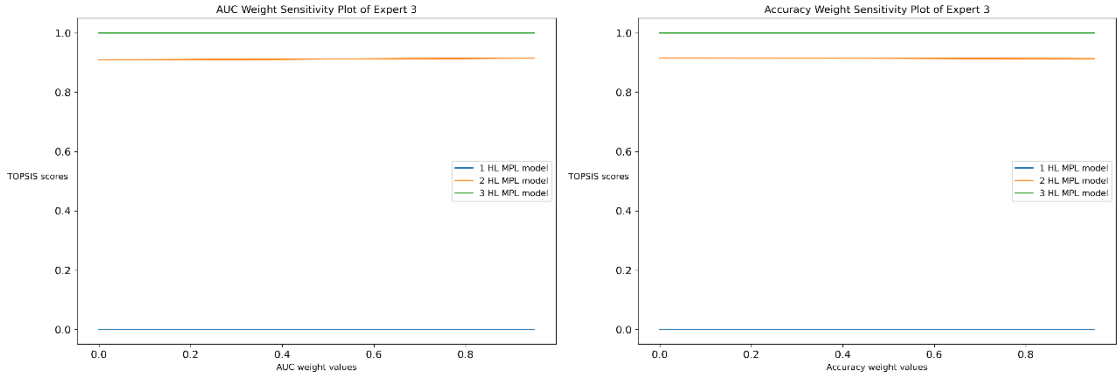


(a) (b)



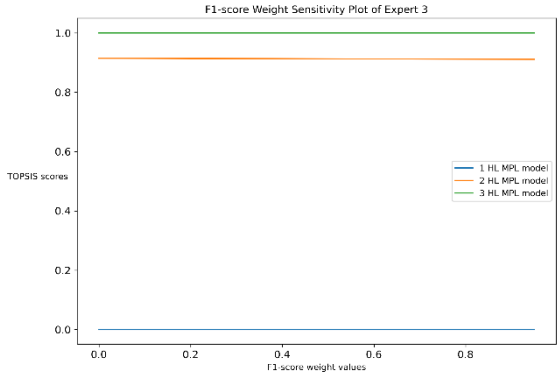
(c) (d)

Figure 6.9. Second expert's opinion sensitivity analysis of metric (a-d) in Credit risk dataset of Kaggle.

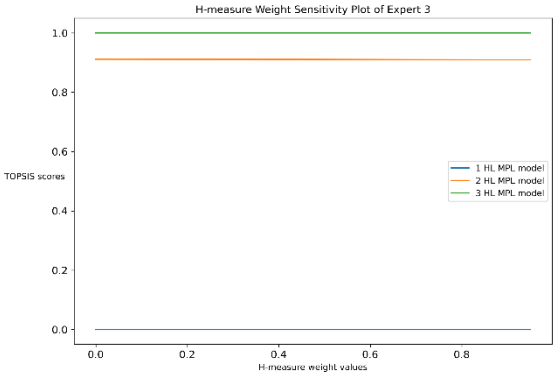


(a)

(b)

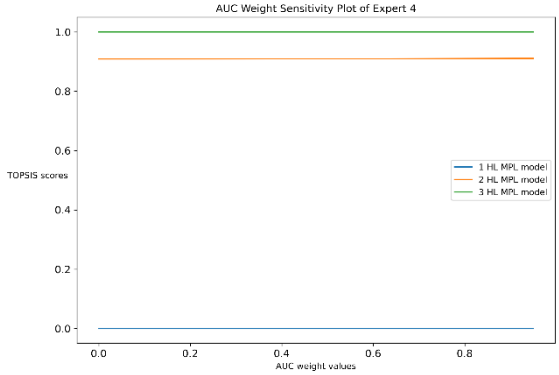


(c)

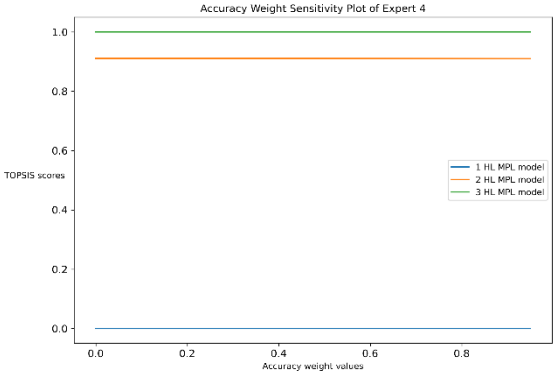


(d)

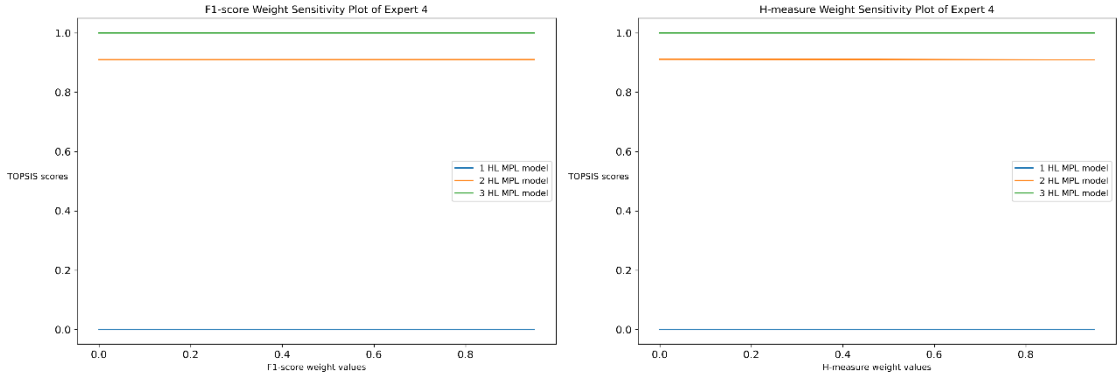
Figure 6.10. Third expert's opinion sensitivity analysis of metric (a-d) in Credit risk dataset of Kaggle.



(a)



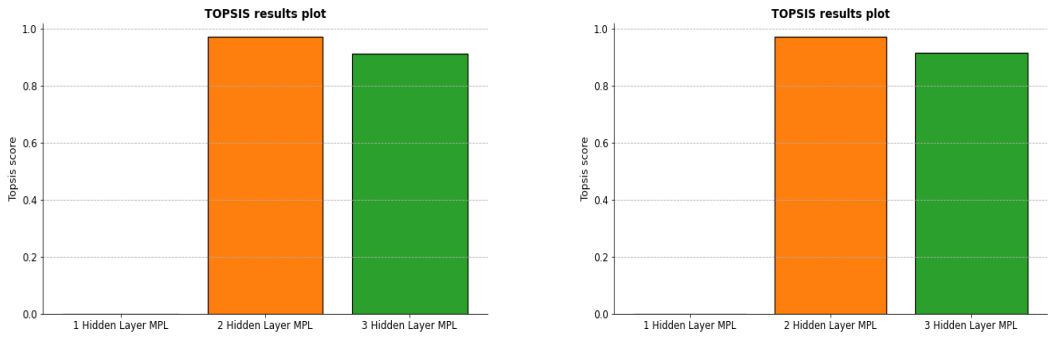
(b)



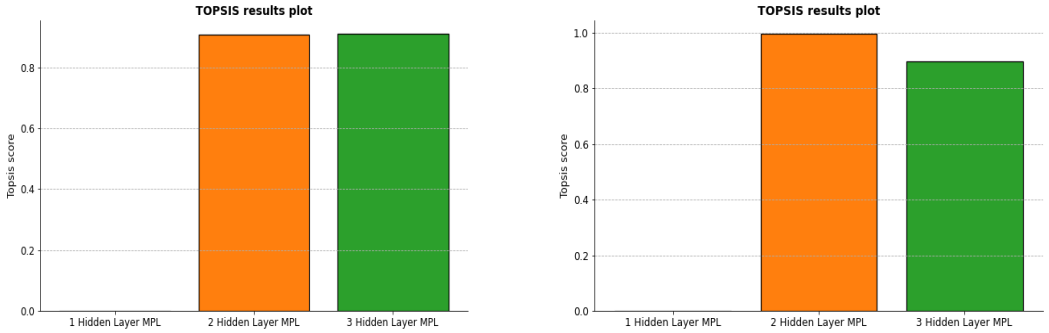
(c) (d)

Figure 6.11. Fourth expert's opinion sensitivity analysis of metric (a-d) in Credit risk dataset of Kaggle.

In a similar manner, Tables 6.39 to 6.42 present the TOPSIS results of the Real-world dataset for each scenario, while Figure 6.12 provides a visual representation of the TOPSIS results. These results indicate that in all scenarios, the 2 Hidden Layer NN model and the 3 Hidden Layer NN model yield similar scores. Specifically, the 2 Hidden Layer NN model performs best in cases (a), (b), and (d). However, in case (c), the 3 Hidden Layer NN model exhibits a slightly higher score compared to the 2 Hidden Layer NN model.



(a) (b)

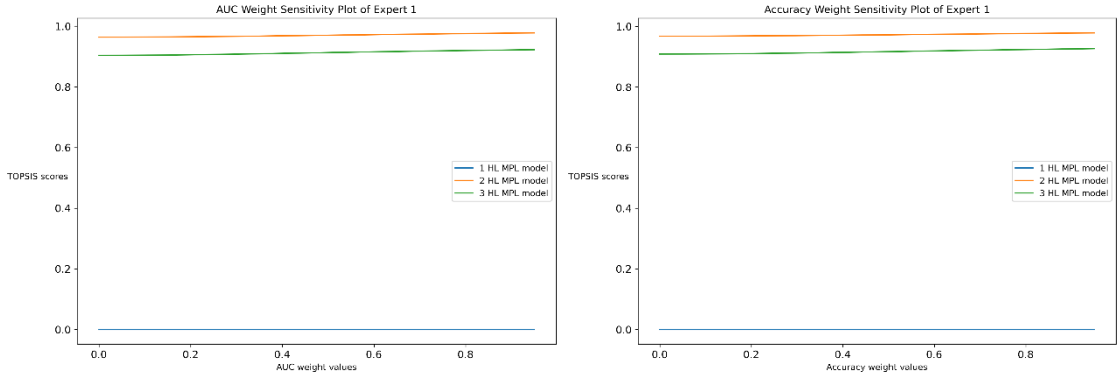


(c) (d)

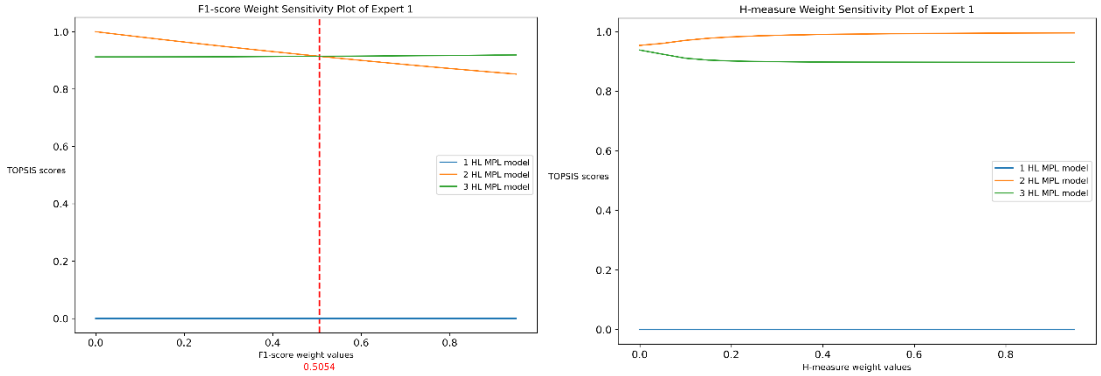
Figure 6.12. TOPSIS results of four (a-d) decision makers in Real-world dataset.

Furthermore, Figures 6.13 to 6.16 provide visual representations of the sensitivity analysis conducted on the evaluation metrics for the Real-world dataset. These figures showcase the variations in the optimal model selection based on different evaluation criteria. The results of the sensitivity analysis highlight the importance of considering multiple evaluation metrics and their respective weights in order to identify the most suitable model for each specific case. These findings emphasize the need for a comprehensive and multidimensional approach when making informed decisions regarding model selection in the context of the Real-world dataset.

Specifically, for the first and second experts, it is important to highlight that the 2 Hidden Layer NN model consistently emerges as the dominant choice for all weight values within the analysis interval, except for scenario (c) where the F1 score dominates. In scenario (c), the 3 Hidden Layer NN model becomes the dominant choice for weight values of F1 score higher than 0.5054 and 0.4809 according to the first and second experts, respectively. In other scenarios, the 3 Hidden Layer NN model becomes the second option, with its scores being in close proximity to those of the dominant model. Conversely, the 1 Hidden Layer NN model performs as the most subordinate option, with its scores approaching zero.

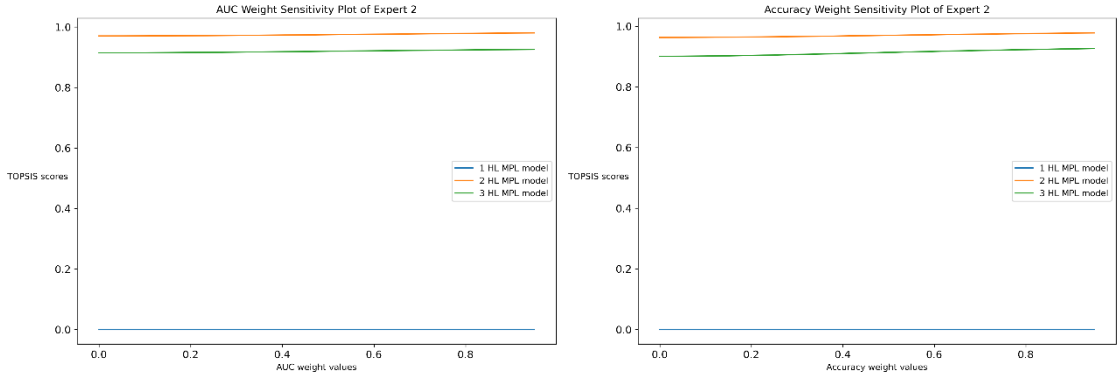


(a) (b)

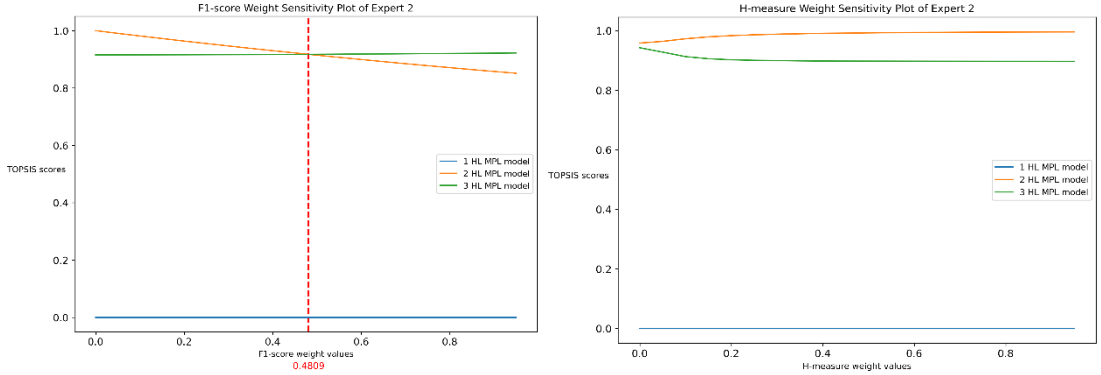


(c) (d)

Figure 6.13. First expert's opinion sensitivity analysis of metric (a-d) in Real-world dataset.



(a) (b)



(c) (d)

Figure 6.14. Second expert's opinion sensitivity analysis of metric (a-d) in Real-world dataset.

Regarding the perspective of the third expert, it is important to note that in scenarios (a) and (b), both the 2 Hidden Layer NN model and the 3 Hidden Layer NN model receive approximately equal scores for every weight value within the analysis interval. Typically, for weight values of AUC and Accuracy scores higher than 0.2565 and

0.3597, respectively, the 2 Hidden Layer NN model emerges as the dominant option. In scenario (c), the 3 Hidden Layer NN model becomes the dominant choice for weight values of F1 score higher than 0.4241. However, in scenario (d), for weight values of H-measure score higher than 0.087, the 2 Hidden Layer NN model emerges as the dominant option. Similar to the previous analyses, the 1 Hidden Layer NN model remains the most subordinate, with its scores approximating zero.

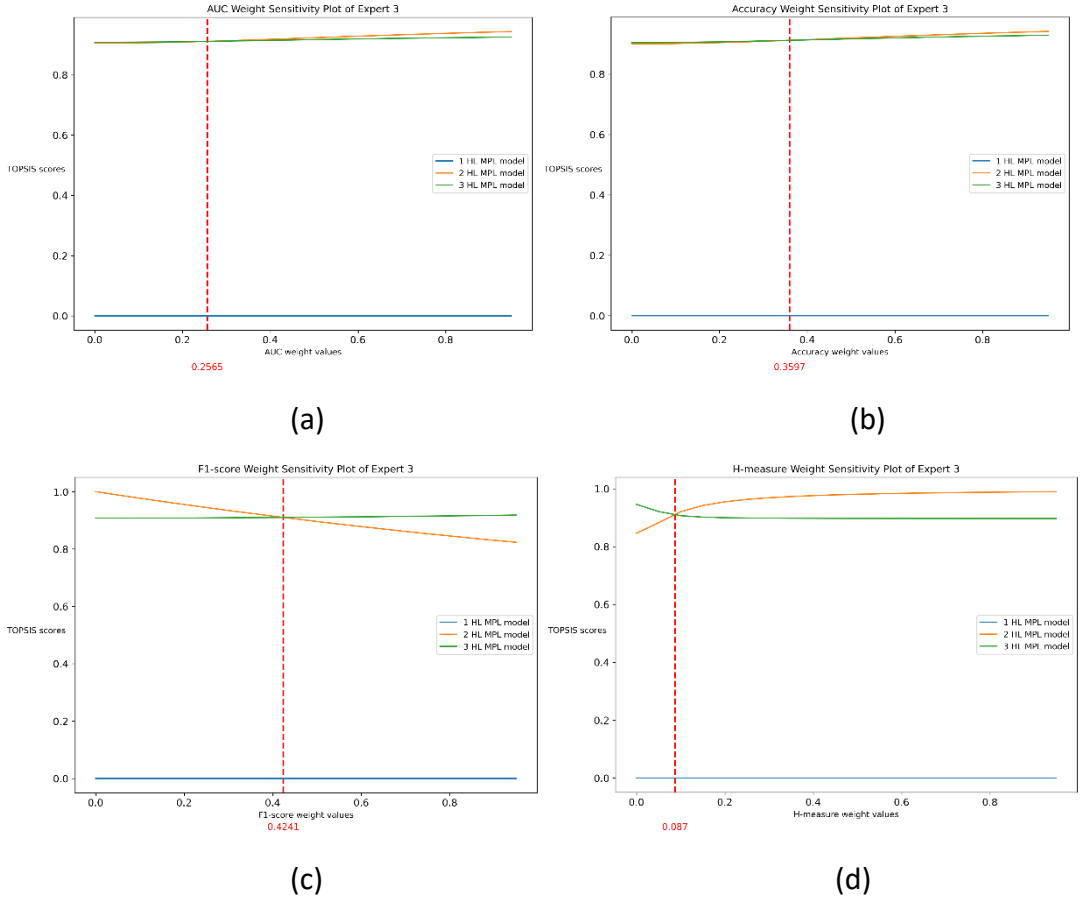


Figure 6.15. Third expert's opinion sensitivity analysis of metric (a-d) in Real-world dataset.

Taking into consideration the perspective of the fourth expert, it is noteworthy that in scenarios (a), (b), and (c), the 2 Hidden Layer NN model emerges as the dominant choice, although its scores closely resemble those of the 3 Hidden Layer NN model. However, in scenario (d), for weight values of H-measure score higher than 0.0203, the 2 Hidden Layer NN model becomes the dominant option. As observed in previous analyses, the 1 Hidden Layer NN model remains the most subordinate, with its scores approximating zero.

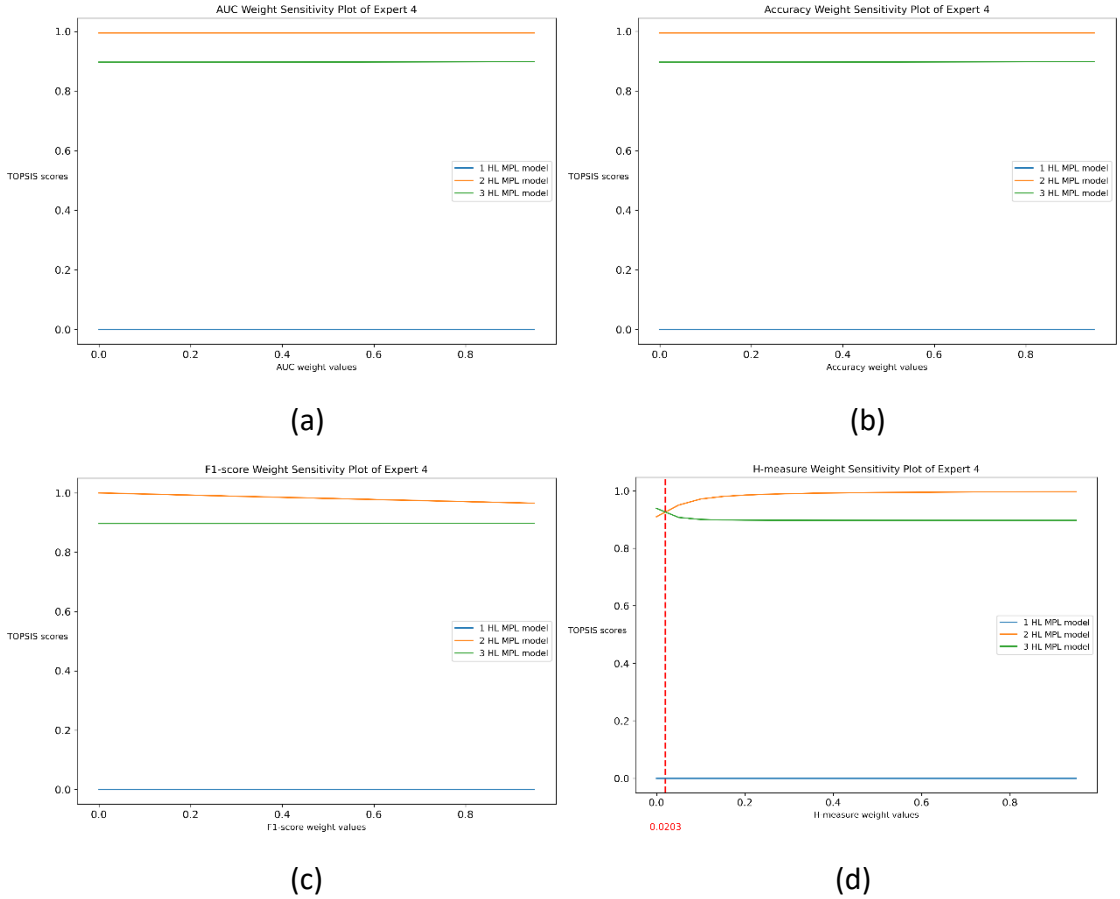


Figure 6.16. Fourth expert's opinion sensitivity analysis of metric (a-d) in Real-world dataset.

6.8. Discussion

The results are presented in section 6.7 indicate that there is no significant difference in the value of metrics among the four hyperparameter tuning methods. As a consequence, there is no dominant method among the three NN models in every dataset. For example, in the Australian credit approval dataset, the Quasi-Monte Carlo method was the best tuning technique for the 1 Hidden Layer (HL) NN, the Bayesian method of Tree-structured Parzen Estimator was the best for the 2 HL NN, and the genetic algorithm was the best for the 3 HL NN. Therefore, it could be concluded that a robust implementation of every hyperparameter tuning method would be able to generate similar evaluation metric scores.

Regarding Fuzzy AHP models, it has been demonstrated that the use of a pairwise comparison matrix (an Excel file in this study) with crisp values representing an expert's view about performance metric comparisons could return imprecise values with an order, reflecting the expert's preferences via FAHP.

The TOPSIS results provide insights into the optimal neural network (NN) models, and their stability is demonstrated through a sensitivity analysis of the weight assigned

to the evaluation metrics. In the Australian credit approval dataset and Credit risk dataset of Kaggle, the 3 HL NN model consistently emerged as the best choice across all scenarios. In contrast, for the Real-world dataset, the 2 HL NN model was optimal in three out of four cases, while the remaining case favored the 3 HL NN model. Notably, in the credit risk and Real-world datasets, the 2 HL NN model and the 3 HL NN model exhibited only marginal differences in their TOPSIS scores.

In terms of the sensitivity analysis of the weight assigned to the evaluation metrics, it is crucial to note that for the Credit risk dataset from Kaggle and the Australian credit approval dataset from UCI, there were no changes in the dominant model as the weight values varied. However, in the Real-world dataset, the preference for the dominant model varied in a few cases. Specifically, in scenario (c) of the Real-world dataset, the 3 Hidden Layer NN model became the dominant choice for weight values of the F1 score higher than 0.5054 and 0.4809, according to the first and second experts, respectively. Additionally, in scenario (c) according to the third expert, the 3 Hidden Layer NN model became the dominant choice for weight values of the F1 score higher than 0.4241. Furthermore, in scenario (d) according to the third expert, for weight values of the H-measure score higher than 0.087, the 2 Hidden Layer NN model emerged as the dominant option. Similarly, in scenario (d) according to the fourth expert, for weight values of the H-measure score higher than 0.0203, the 2 Hidden Layer NN model became the dominant choice. These variations in model preference based on the weight assigned to evaluation metrics highlight the sensitivity of the decision-making process and the importance of considering different weighting schemes to ensure robust and informed model selection.

In summary, the aggregated TOPSIS ranking results are consolidated in Table 6.43. These results highlight that the 3 HL MLP (Multi-Layer Perceptron) model is the most preferred choice in both the Australian credit approval and Credit risk dataset of Kaggle. However, in the Real-world dataset, there is a contention between the 2 HL MLP model and the 3 HL MLP model, as they have comparable rankings. Furthermore, based on these datasets, it could be inferred that the 1 HL NN model consistently performs as the most subordinate model, regardless of the opinions provided by the experts.

Table 6.43. Best MPL models based on TOPSIS.

Fuzzy AHP	Datasets		
	Australian credit approval	Credit risk	Real-world
1st expert	3 HL MLP model	3 HL MLP model	2 HL MLP model
2nd expert	3 HL MLP model	3 HL MLP model	2 HL MLP model
3rd expert	3 HL MLP model	3 HL MLP model	3 HL MLP model
4th expert	3 HL MLP model	3 HL MLP model	2 HL MLP model

Note: HL := Hidden Layer MLP := Multi-Layer Perceptron

Chapter 7:
Conclusions and Future Work

To conclude, in artificial neural network topology, one of the most complex decisions is the determination of selecting an optimal number of hidden layers. In recent decades, plenty of scientists have introduced novel theories and techniques for neural network structure determination. In the finance industry, human thinking has a vital role, thus it is necessary to be comprehended by neural networks. This difficulty could be overcome by fuzzy logic and in this study considering this, proposed a novel Neuro-Fuzzy model.

For the purpose of this thesis, the proposed Neuro-Fuzzy model was applied to three datasets and it takes into account the opinions of four decision-makers. Hence, experimental results could not be generalized and representative of all types of datasets and decision-makers. Contribution to the scientific community lies in the fact that every finance institute has its own loan policy and as a result, they determine the weights of evaluation metrics. For instance, in the case of an institute that considers the most significant factor is the prediction of solvent percentage then accuracy would be the main evaluation metric.

Limitations in this thesis consist of datasets and experts' views of diversity since there are abundant cases in reality. Another crucial restriction is the limited computational sources in terms of time-consuming training processes. Of course, it should be mentioned that datasets are time-limited, and future clients' behaviors would produce different results.

In future expansions of this study, experiments with different neural network topologies could be conducted, due to the plentiful potential architectures and types of neural networks that could be explored, such as convolutional neural networks or recurrent neural networks. Future studies could test additional topologies and compare their performance. Additionally, as far as tuning techniques are concerned, the current model description applied four tuning techniques to each neural network model, but many other possible techniques could be tested, such as grid search. Future studies could compare the performance of different tuning techniques. Furthermore, the current model description tested the model on specific three datasets for credit risk assessment, but future studies could test the model on more different datasets and in different domains to reveal whether it could be generalized. As well, future studies could investigate other methods for incorporating expert knowledge, such as fuzzy logic with Mamdani, Takagi-Sugeno, Larsen, or Tsukamoto Fuzzy Inference System mechanism implementation. Moreover, different methods for selecting the most robust model could be used instead of TOPSIS, such as VIKOR, Weighted Sum Models (WSM),

PROMETHEE, etc. Finally, more evaluation metrics except for the four metrics of this study could be used.

Appendix

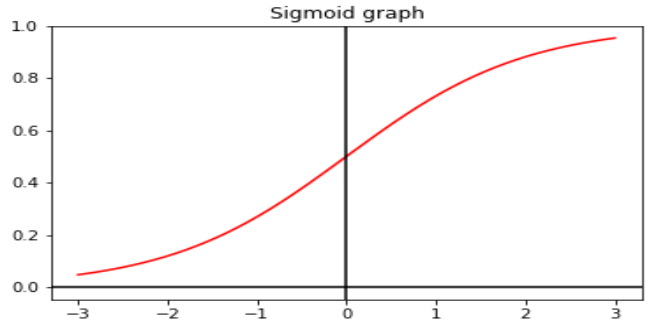
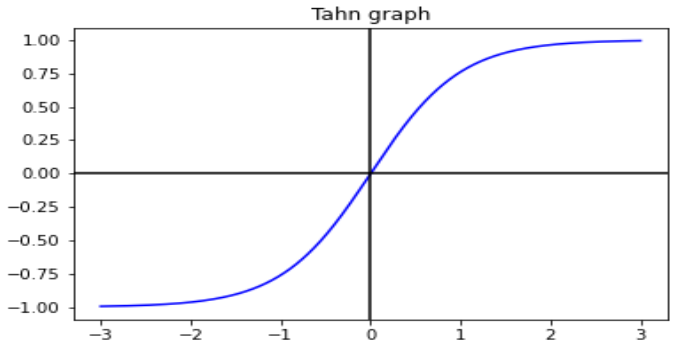
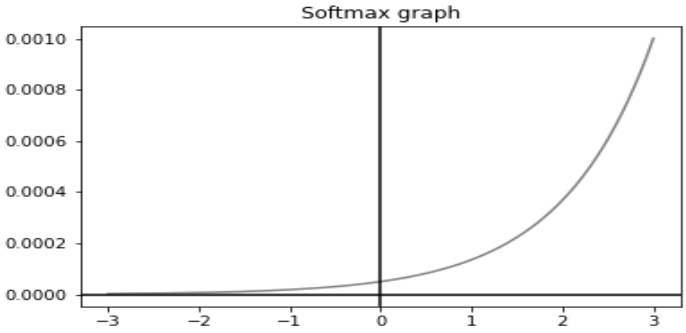
A. In Chapter 3, Neural Networks were described. However, it is essential to have an extended description of a few fundamental terminology. For this reason, the below tables are provided more information about the most popular loss functions (Table 3.1), activation functions (Table 3.2), neural network optimizers (Table 3.3), differences between L1 and L2 regularization noticed (Table 3.4), the combination of activation and loss functions (Table 3.5), and advantages and disadvantages of neural network pruning (Table 3.6).

Table 3.1. Loss Functions (LF).

Problem type	Name LF	Equation	Advantages	Disadvantages
Classification	Binary Cross-Entropy/Log Loss	$\frac{1}{n}\sum_{i=1}^{n} -(y_i log(p_i) + (1-y_i) log(1-p_i)$	This is used in binary classification models	
	Hinge loss	$\frac{1}{n}\sum_{i=1}^{n} max(0,1 - y_{ij} \times \hat{y}_{ij})$	It is used when the network must be optimized for a hard binary classification.	
	Categorical Cross-Entropy Loss	$-\frac{1}{n}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij})$	The number of classes is greater than two	
Regression	Mean Squared Error (MSE)	$\frac{1}{n}\sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$	Utilize gradient descent optimization to set the weight values.	it is very sensitive to outliers
	Mean Absolute Error (MAE)	$\frac{1}{n}\sum_{i=1}^{n} \lvert (y^{(i)} - \hat{y}^{(i)} \rvert$	It is used for a large number of outliers	gradient descent optimization could not apply

Huber Loss	$\begin{cases} \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2, & \lvert (y^{(i)} - \hat{y}^{(i)}\rvert \le \delta \\ \frac{1}{n}\sum_{i=1}^{n}\delta(\lvert y^{(i)} - \hat{y}^{(i)}\rvert - \frac{1}{2}\delta), & \lvert (y^{(i)} - \hat{y}^{(i)}\rvert > \delta \end{cases}$	It has the advantages of both MSE and MAE

Table 3.2. A few activation functions are portrayed.

Function	Equation	Graph
Sigmoid	$\frac{1}{1+e^{-x}}$	
Tanh	$\frac{2}{1+e^{-2x}}$	
Softmax	$\frac{e^i}{\sum e^i}$	

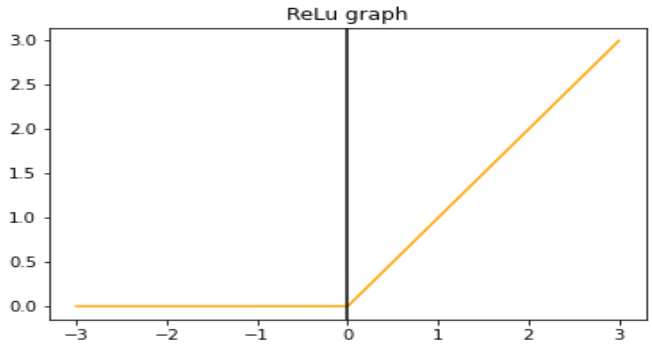
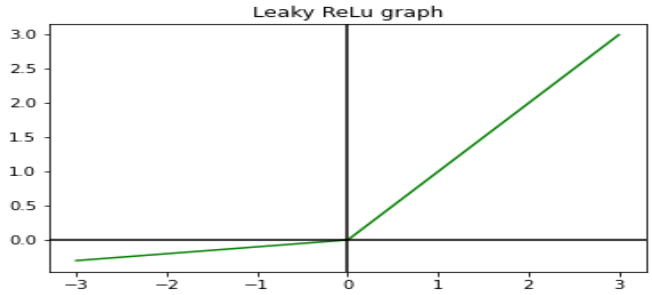
Rectified Linear Unit (ReLU)	$max(0, x)$	
Leaky ReLU	{x, if x>0} {ax, otherwise}	

Table 3.3. Neural network optimizers

	Advantages	Disadvantages
Gradient Descent	Easy computation, implementation and understanding.	Local minima might trap. After computing the gradient on the entire dataset, weights are modified. Therefore, it might take years for this to converge to the minimum if the dataset is too vast. considerable memory is needed to compute the gradient throughout the whole dataset.
Stochastic Gradient Descent	Model parameters are often updated; as a result, convergence occurs faster. Less memory is used since loss function values don't need to be stored. Perhaps new minimas	High variation in the model's inputs. Even after reaching the global minimum, might shoot. The value of the learning rate must be gradually decreased to get the same convergence as gradient descent.

Mini-Batch Gradient Descent	Adjusts the model's parameters often and has a lower variance. medium quantity of RAM is needed.	Choosing the best learning rate value. Gradient descent could take a very long time to converge if the learning rate is too slow. Have all the parameters learn at the same rate. We might not wish to modify some factors at the same rate. Maybe stuck at a nearby minima.
Momentum	Lowers the parameters' large variance and oscillations. quicker than gradient descent converges.	There is now an additional hyper-parameter that must be carefully and manually chosen.
Nesterov Accelerated Gradient	Avoid skipping the nearby minimum. if minima are present, slows down.	Hyperparameter needs to be selected manually.
Adagrad	For each training parameter, the learning rate varies. does not require manual adjustment of learning rate. capable of sparse data training.	Costly computationally since the second order derivative must be calculated. Slow training is caused by the constant decline in learning rate.
AdaDelta	The training is still ongoing and the learning rate is no longer declining.	Computationally expensive.
Adam	The approach converges too quickly and moves too quickly. Corrects large variance and vanishing learning rate.	Computationally costly.
RMS-Prop	For each parameter, RMS-Prop selects a different learning rate that is automatically adjusted.	Slow Learning

Table 3.4. Differences between L1 and L2 regularization

L1 Regularization	L2 Regularization
Panelizes the sum of absolute value of weights	penalizes the sum of square weights.
It has a sparse solution.	It has a non-sparse solution.
It gives multiple solutions	It has only one solution.
Constructed in feature selection.	No feature selection.
Robust to outliers	Not robust to outliers
It generates simple and interpretable models	It gives more accurate predictions when the output variable is the function of whole input variables
Unable to learn complex data patterns	Able to learn complex data patterns
Computationally inefficient over non-sparse conditions.	Computationally efficient because of having analytical solutions

Table 3.5. Combination of activation and loss functions.

Problem type	Last-layer activation	Loss function
Binary classification	Sigmoid	binary_crossentropy
Multiclass, single-label classification	Softmax	categorical_crossentropy
Multiclass, multilabel classification	Sigmoid	binary_crossentropy
Regression to arbitrary values	None	Mse
Regression to values between 0 and 1	Sigmoid	mse or binary_crossentropy

Table 3.6. Advantages and disadvantages of neural network pruning

Advantages	Disadvantages
Relies on the hardware and compression strategy to reduce the inference and training times.	There are fewer versions and pre-trained models available.
Storage needs decrease as the number of neurons, connections between layers, and weights decrease.	Selection compression is challenging because it requires knowledge of the hardware architecture of the target.
Reduces the heat generated by installed devices, such mobile phones. Saving energy.	Beyond initial accuracy, hardly much could be quantified.

A.1. Evaluation metrics. In machine learning (ML), there are plenty of metrics to evaluate ML models. The choice of evaluation metric completely depends on the type of model, that is classification or regression model, and the implementation plan of the model. In credit scoring, classification models are used to predict if a client is well-payer or not, whereas in regression models a credit score for every client is calculated. In this thesis classification models are created

and for this reason evaluation metrics, accuracy, f1-score, Area Under Curve (AUC), and H-measure would be described. It is notable that plenty of data scientists raised disagreements related to the significance of each metric. A part of them supports that AUC is the best metric to indicate the optimal model whereas other scientists believe that other metrics are better.

The fundamental concept of metrics relies on the confusion matrix in Table 3.7.

3.7. Confusion matrix of classification models

		Actually	
		Positive	Negative
Predicted	Positive	True Positive (tp)	False Negative (fn)
	Negative	False Positive (fp)	True Negative (tn)

Accuracy represents the percentage of correct predictions and is defined by the equation:

$$Accuracy = \frac{tp + fn}{tp + fp + tn + fn}$$

F1 score represents is the harmonic mean of precision and recall values for a classification problem. Precision is the proportion of positive identifications was actually correct and is formulated by:

$$Precision\ (P^c) \coloneqq \frac{tp}{tp + fp}$$

whereas Recall is the proportion of actual positives was identified correctly and is defined by:

$$Recall\ (R^c) \coloneqq \frac{tp}{tp + fn}$$

Therefore, F1-score is defined by,

$$F - score:\ F_1 = 2 * \frac{P * R}{P + R}$$

An area under the Receiver Operating Characteristic (ROC) curve is represented by the Area Under Curve (AUC). The receiver operating characteristic curve displays the true positive (TP) rate vs the false positive (FP) rate at various categorization levels. The thresholds in binary classification are distinct probability cutoffs that separate the two classes. It informs us of a model's ability to discriminate between classes using probability (Müller & Guido, 2016).

The H-measure is a classifier performance metric that considers the application environment without needing a fixed value to be established for relative misclassification costs. The issue of collecting performance across several prospective scenarios is effectively solved. Furthermore, it is significant because it puts forth a reasonable standard for the coherence of performance metrics, which the H-measure meets but which, surprisingly, is not met by a number of widely used alternatives, most notably the Area Under the Curve (AUC) and its variations, such as the Gini coefficient (Hand & Anagnostopoulos, 2022).

B. Fuzzy logic and Multicriteria decision analysis were discussed in Chapter 4. With an aim to be addressed a few concepts, the following tables explain basic terms of fuzzy theory (Table 4.1), the most popular Membership Functions (MF) (Table 4.2), operators of fuzzy sets (Table 4.3),

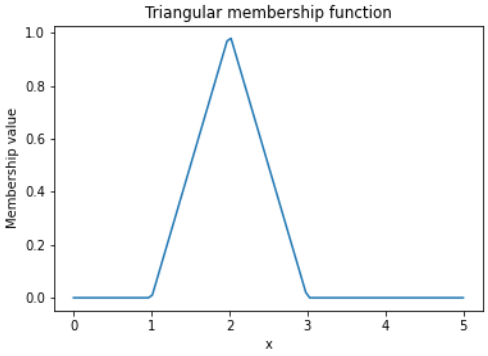
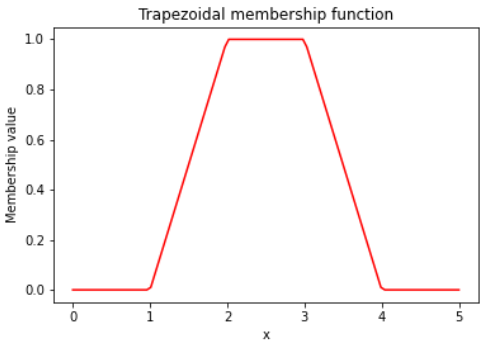
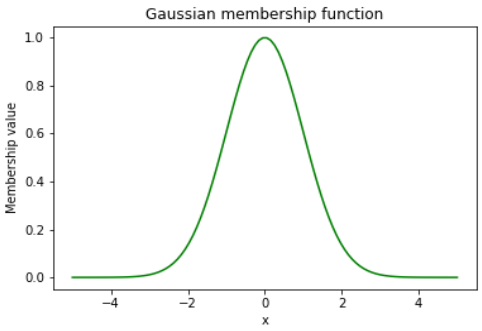
compositional rule of inference (Table 4.4), properties that generalized Modus Ponens should satisfy (Table 4.5), description of fuzzy inference operators (Table 4.6), the most famous defuzzification operators (Table 4.7), the steps of fuzzy AHP (Table 4.8) and TOPSIS (Table 4.9) methods.

Table 4.1. Terminology of Fuzzy Theory (Rutkowska, 2001).

Definitions	Equations (where $\mu_A(x)$ is the membership function of x in X)
*The **support** of a fuzzy set A, denoted by supp A, is the set of points in* X *at which the membership function $\mu_A(X)$ is positive.*	$supp\ A = \{x\ \in X\ ; \mu_A(x) > 0\}$
*A **fuzzy singleton** is a fuzzy set A whose support is a single point x in the universe of discourse* X.	$A = \frac{\mu_A(x)}{x}$
*The **core** of a fuzzy set A defined in the universe of discourse* X, *denoted by core(A), also referred* to *as kernel or nucleus, is the set of points in* X *at which the membership function $\mu_A(x)$ equals* 1, *that is*	$corr(A) = \{x\ \in X\ ; \mu_A(x) = 1\}$
*The **height** of a fuzzy set A defined in the universe of discourse* X, *denoted by hgt(A), is the maximal value of its membership function $\mu_A(x)$, that is*	$hgt(A) = \sup_{x \in X} \mu_A(x)$
*A fuzzy set A is called a **normal fuzzy set** if and only if the maximal value of its membership function equals* 1, *which means that hgt (A)* = 1.	
*A fuzzy set A defined in the universe of discourse X is an **empty set**, denoted A = 0, if and only if its membership function $\mu_A(x)$ = 0 for all x ∈ X.*	
*A fuzzy set A defined in the universe of discourse X, which we shall assume to be a real Euclidean N-dimensional space, is **convex** if and only if*	$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)]$ *for all* x_1 *and* x_2 *in X and all* λ *in* [0, 1].

*A fuzzy set A is a **fuzzy number** if the universe of discourse X is $\mathbb{R}$ and the following criteria are fulfilled: the fuzzy set A is convex, normal, the membership function of the fuzzy set $\mu_A(x)$ is piecewise continuous, and the core of the fuzzy set consists of one value only.*		
*A fuzzy set A is a **fuzzy interval** if the universe of discourse X is $\mathbb{R}$ and the following criteria are fulfilled: the fuzzy set A is convex, normal, and the membership function of the fuzzy set $\mu_A(x)$ is piecewise continuous.*		
***Two fuzzy sets A and B are equal**, written as A = B, if and only if their membership functions are equal, that is $\mu_A(x) = \mu_B(x)$ for all x in the universe of discourse X.*		
*The crisp (non-fuzzy) set of elements that belong to the fuzzy set A in X at least to the degree of a is called an **a-level set (or a-cut)** and defined by*	$A_\alpha = \{x \in X ; \mu_A(x) \geq \alpha\}$	$\forall \alpha \in [0,1]$
*A **fuzzy set of type 2** is defined by a fuzzy membership function, the grade (that is, fuzzy grade) of which is a fuzzy set in the unit interval [0,1], rather than a point in [0,1].* *This definition implies that the membership function of a fuzzy set of type 1 ranges over the interval [0, 1], the membership function of a fuzzy set of type 2 ranges over fuzzy sets of type 1, the membership function of a fuzzy set of type 3 ranges over fuzzy sets of type 2, etc. (Zadeh, 1975).*		

Table 4.2. The most popular Membership Functions (MF) (Singh & Lone, 2020).

Type	Definition	Equation	Figure
Triangular	Supposing a set X, ∀ x ∈ X, α, b, c ∈ $\mathbb{R}$ triangular MF is created by following equation:	$f(x;\alpha,b,c) = max\left(min\left(\frac{x-\alpha}{b-\alpha},\frac{c-x}{c-b}\right),0\right),$	
Trapezoidal	Supposing a set X, ∀ x ∈ X, α, b, c, d ∈ $\mathbb{R}$, which are α < b < c < d, trapezoidal MF is created by following equation:	$f(x;\alpha,b,c) = max\left(min\left(\frac{x-\alpha}{b-\alpha},1,\frac{d-x}{d-c}\right),0\right),$	
Gaussian	Supposing a set X, ∀ x ∈ X, with c, s, m mean, standard deviation and Fuzzification Factor, respectively, gaussian MF is created by following equation:	$\mu_A(x,c,s,m) = e^{-\frac{1}{2}\left\|\frac{x-c}{s}\right\|^m},$	

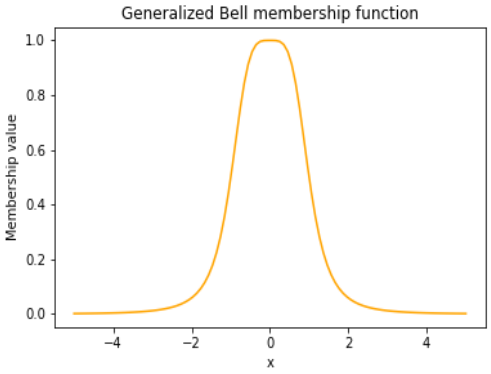
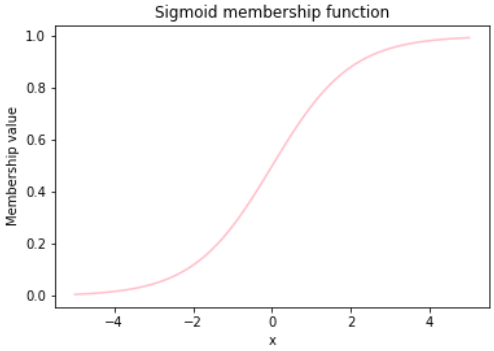
Generalized Bell	Supposing a set X, ∀ x ∈ X, with α, b, c to be slope, center, width of curve, respectively, generalized bell MF is constructed by following equation:	$$\mu_A(x,\alpha,b,c) = \frac{1}{1+\left\|\frac{x-c}{b}\right\|^{2b}},$$	
Sigmoidal	Supposing a set X, ∀ x ∈ X, with α, c to be slope and crossover point, respectively, sigmoidal MF is constructed by following equation:	$$Sigmoid(x;\alpha,c) = \frac{1}{1+e^{-\alpha(x-c)}},$$	

Table 4.3. Operators of fuzzy sets (Rutkowska, 2001).

Operators	Definition
Complement	*The complement of a fuzzy set A, denoted by $\tilde{A}$, is defined by* $$\mu_{\tilde{A}}(x) = 1 - \mu_A(x) \quad \forall x \in X$$ *The complement of the fuzzy set A is specified by a function* $$c: [0,1] \rightarrow [0,1]$$

for each membership grade $\mu_A(x)$ a value c ($\mu_A(x)$) is assigned by the above equation

Union

The union of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_A(x)$ is a fuzzy set denoted by A ∪ *B whose membership function is given by*,

$$\mu_{A \cup B} = max[\mu_A(x), \mu_B(x)] \qquad \forall\, x \in X$$

Intersection

The intersection of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set denoted by A ∩ *B whose membership function is given by,*

$$\mu_{A \cap B} = min[\mu_A(x), \mu_B(x)] \qquad \forall\, x \in X$$

Triangular-norm

A triangular norm T is a function of two arguments

$$T\colon [0,1]\ x\ [0,1] \rightarrow [0,1]$$

which satisfies the following conditions for a, b, c, *d* ∈ [0,1]

- *Monotonicity: T (a, b) ≤ T (c, d)*; *a* ≤ c; *b* ≤ *d*
- *Commutativity: T (a, b) = T (b, a)*
- *Associativity: T (T (a, b),* c) = *T (a, T (b, c))*
- *Boundary conditions: T(a,0)* = 0; *T(a,* 1) = *a*

Moreover, every triangular norm fulfils the following inequality

T_w (a, b) ≤ T (a, b) ≤ min (a, b)

where

$$T_w(a,b) = \begin{cases} a & if \quad b = 1 \\ b & if \quad a = 1 \\ 0 & if\ a, b \neq 1 \end{cases}$$

The T-norm would also be denoted as,

$$\mathrm{T}(\mathrm{a}, \mathrm{b}) = \mathrm{a}\ {}^{T}_{*}\ b$$

S-norm

An S-norm is a function of two arguments

$$S\colon [0,1]\ x\ [0,1] \rightarrow [0,1]$$

which satisfies the following conditions for a, b, c, *d* ∈ [0,1]

- *Monotonicity: S (a, b) ≤ S (c, d)*; *a* ≤ c; *b* ≤ *d*
- *Commutativity: S (a, b) = S (b, a)*
- *Associativity: S (S (a, b),* c) = *S (a, S (b, c))*

	Moreover, every S-norm fulfils the following in equality. Moreover, every S-norm fulfils the following inequality S_w (a, b) ≤ S (a, b) ≤ max (a, b) where $$S_w(a,b) = \begin{cases} a & if \quad b = 0 \\ b & if \quad a = 0 \\ 0 & if\ a, b \neq 0 \end{cases}$$ The S-norm, depicted in Definition 18, would also be denoted as, $$\mathrm{S(a,b)} = \mathrm{a} \overset{S}{*} b$$
Containment	*A fuzzy set A is contained in a fuzzy set B (or, equivalently, A is a subset of B, or A is smaller than or equal to B) if and only if* $\mu_A(x); \mu_B(x)$*. Formally,* $$A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x) \qquad \forall\, x \in X$$

Table 4.4. Compositional rule of inference (Fuller, 2000).

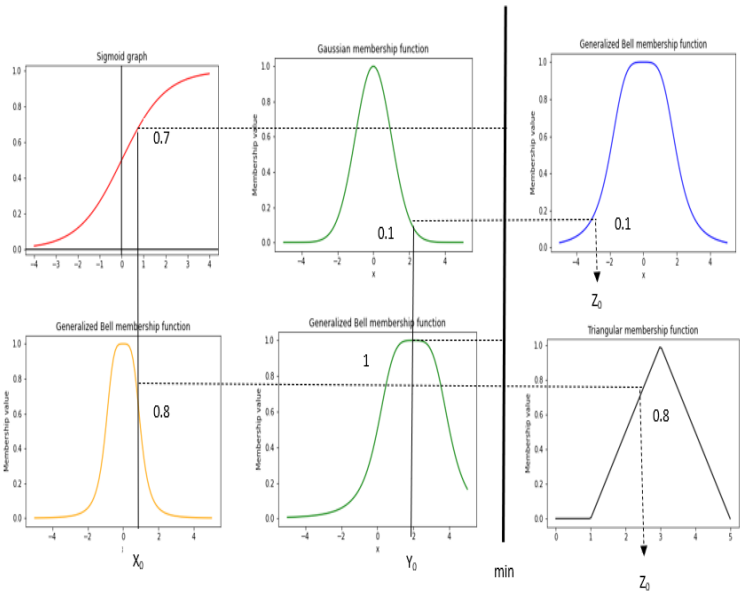
Rules	Conditions
Entailment	*if x is A and A ⊂ B, then x is B*
Conjuction	*if x is A and x is B, then x is A ∩ B*
Disjunction	*if x is A or x is B, then x is A ∪ B*
Projection	*if two members of Fuzzy Set X ∧ Y: x, y respectively, and between them there are a relation R, then you could define a projection rule on them. For example, If (x, y) is close to (4, 5), then x is close to 4 and y is close to 5.*
Negation	*if not (x is A), then x is ¬A*

Table 4.5. Properties that generalized Modus Ponens should satisfy (Fuller, 2000).

Basic property:	*If x is A, then y is B*	(Premise)
	x is A	(Fact)
	y is B	(Consequence)
Total Indeterminacy property:	*If x is A, then y is B*	(Premise)
	x is¬A	(Fact)
	y is Unknown	(Consequence)
Subset property:	*If x is A, then y is B*	(Premise)

	x is A' ⊂ A	(Fact)
	y is B	(Consequence)
Superset property:	*If x is A, then y is B*	(Premise)
	x is A'	(Fact)
	y is B' ⊃ B	(Consequence)

Table 4.6. Fuzzy inference operators (Fuller, 2000).

Operators	Definitions	Figures
Tsukamoto	Supposing that all linguistic terms characterized by a monotonic membership function. The firing levels of the rules, a_i, *with* i = 1,2, $$\alpha_i = A_i(x_0) \wedge B_i(y_0)$$ For individual control actions z_1, z_2 the global crisp control action is, $$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}$$ If there are n rules in rule-base then crisp control action is, $$z_0 = \frac{\sum_{i=1}^{n} \alpha_i z_i}{\sum_{i=1}^{n} \alpha_i}$$	

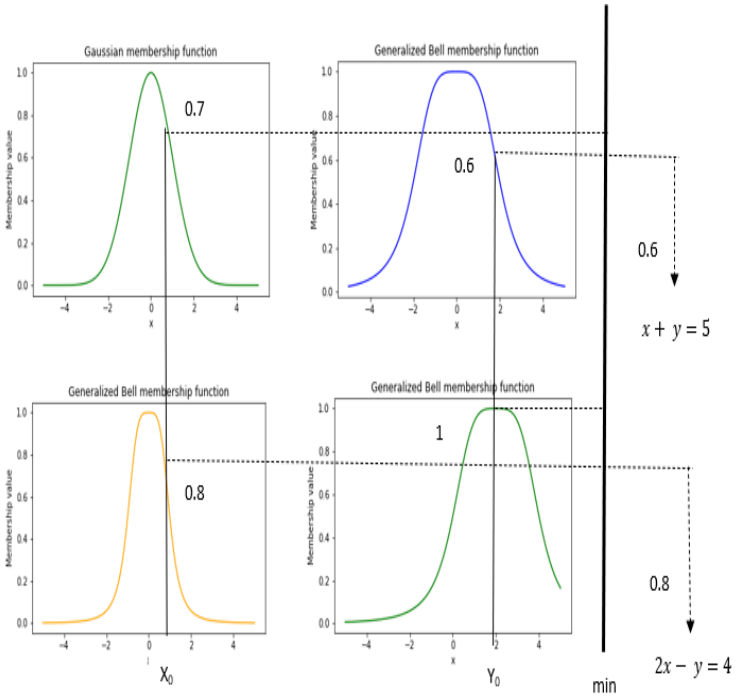
Sugeno

Sugeno and Takagi use the following architecture:

R_1: if x is A_1 and y is B_1 then $z_1 = \alpha_1 x + b_1 y$

R_2: if x is A_2 and y is B_2 then $z_2 = \alpha_2 x + b_2 y$

Fact: x is x_0 and y is y_0



Cons: z_0

The firing levels of the rules, α_i, *with* i = 1,2,

$$\alpha_i = A_i(x_0) \wedge B_i(y_0)$$

As a result, the individual rule outputs,

$$z_i^* = \alpha_i x_0 + b_i y_0$$

For individual control actions z_1, z_2 …, z_n if there are n rules, the global crisp control action is,

$$z_0 = \frac{\sum_{i=1}^{n} \alpha_i z_i}{\sum_{i=1}^{n} \alpha_i}$$

Larsen

Larsen's product operator is one of the most well-known fuzzy implications and defined by max operator. The firing level of n rules is donated as α_i , with i = 1,2, … , n

$$\alpha_i = A_i(x_0) \wedge B_i(y_0)$$

The membership function of the inference consequence C when there are n rules with firing level, α_i is computing by,

$$C(w) = \bigvee_{i=1}^{n} (\alpha_i C_i(w))$$

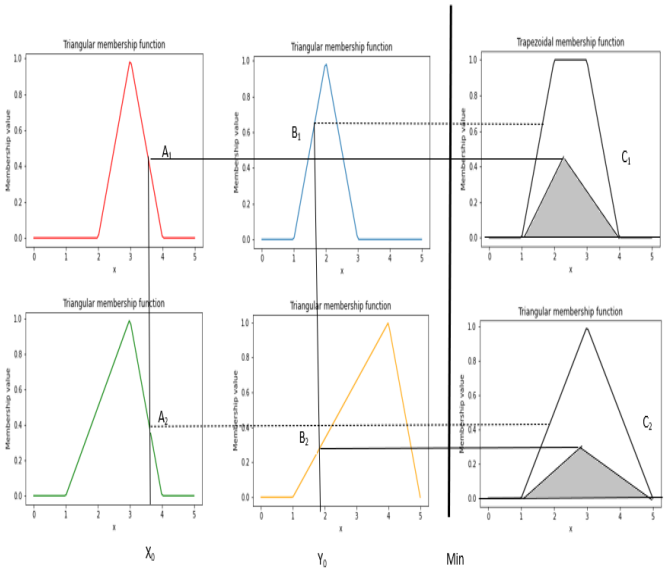


Table 4.7. Defuzzification operators (Fuller, 2000).

Center of Gravity. The defuzzified value of a fuzzy set A is defined as its fuzzy centroid for continuous membership function (μ_A):	$$z^* = \frac{\int z\mu_A(z)dz}{\int \mu_A(z)dz}$$
Center of Area. The center of gravity of the sub region with the largest area could be used to calculate the defuzzified value. Therefore, the defuzzified value is calculated, as:	$$z^* = \frac{\int z'\mu_{Am}(z)dz}{\int \mu_{Am}(z)dz}$$ where Am is the universe of discourse with the largest area, z' is the center gravity of this area.
Center of Sum Method. The main advantages of methods are the high speed and the lack of symmetric membership values constrains. This method could be represented as:	$$z^* = \frac{\int \underline{z} \sum_{k=1}^{n} C(z)dz}{\int \sum_{k=1}^{n} C(z)dz}$$ where $\underline{z}$ is the distance of the centroid from each membership function.
Height method. The objects with membership grades lower than a threshold α, in the universe of discourse A, are discounted. The application of the Center of Area method for remaining objects computes the defuzzified value z*	$$z^* = \frac{\int z\mu_A(z)dz}{\int \mu_A(z)dz}$$
First of Maxima. The smallest value of the object with maximized membership degree formed as:	$$z* = min(z \mid \mu_A(z) = max(\mu_A(X)))$$
Middle of Maxima. In a similar manner, the mean value of the objects with maximized membership degree in case of discrete fuzzy set A, is formed as:	$$z^* = \frac{\int_G zdz}{\int_G dz}$$ where z is a set of objects in the universe of discourse W, where maximum value of $\mu_A(z)$ is placed and G is a set that maximum object of A is located.
Last of Maxima. This concept is opposite to first of Maxima in terms of determining the higher value of the universe of discourse with maximized membership degree	$$z^* = \{z \mid \mu_A(z) = max(\mu_A(X))$$

Table 4.8. The steps of fuzzy AHP method (Aliyev & Temizkan, 2020).

Step 1. The fuzzy matrix $\tilde{C}$ should be constructed and decomposed into three matrices C_l, C_m, C_u which means matrices of lower, medium and upper values of triangular fuzzy numbers. These fuzzy matrices consist of triangular fuzzy numbers and the pairwise comparisons of the criteria or alternatives with respect to each other are illustrated.

Step 2. The three matrices from the previous step would be used to calculate the system of fuzzy linear homogeneous equations.

$$\tilde{C}_l w_l + \tilde{C}_m w_m + \tilde{C}_u w_u - \tilde{\lambda}_l w_l - \tilde{\lambda}_m w_m - \tilde{\lambda}_u w_u = 0$$

where

$$\tilde{C}_l = 2C_l + C_m,\ \tilde{C}_m = C_l + 4C_m + C_u,\ \tilde{C}_u = C_m + 2C_u$$

Step 3. The eigenvalues $\tilde{\lambda}_l$, $\tilde{\lambda}_m$, $\tilde{\lambda}_u$ of matrices $\tilde{C}_l$, $\tilde{C}_m$, $\tilde{C}_u$ and eigenvalues λ_l, λ_m, λ_u of matrices C_l, C_m, C_u are calculated by using these equations:

$$\tilde{\lambda}_l = 2\lambda_l + \lambda_m,\ \tilde{\lambda}_m = \lambda_l + 4\lambda_m + \lambda_u\ ,\ \tilde{\lambda}_u = \lambda_m + 2\lambda_u$$

Step 4. The eigenvectors w_l, w_m, w_u of matrices $\tilde{C}_l$, $\tilde{C}_m$, $\tilde{C}_u$ and $\overline{w}_l$, $\overline{w}_m$, $\overline{w}_u$ by using the formulas:

$$\overline{w}_l = \frac{w_l \lambda_l}{s_l \lambda_m},\ \overline{w}_m = \frac{w_m}{s_m},\ \overline{w}_u = \frac{w_u \lambda_u}{s_u \lambda_m},$$

$$s_l = \sum_{i=1}^{n} w_{i,l}\ ,\qquad s_m = \sum_{i=1}^{n} w_{i,m}\ ,\qquad s_u = \sum_{i=1}^{n} w_{i,u}$$

Step 5. The consistency index (CI) and consistency ratio (CR) of the matrix C_m are calculated. CR should be lower than 0.1 to claim that the comparison matrix is consistent and RI is the random index and depends on the size of the matrix.

$$CI = \frac{\lambda_{max} - n}{n-1},\ CR = \frac{CI}{RI}$$

Step 6. The priority fuzzy matrices P_l, P_m, P_u that contain normalized eigenvectors $\overline{w}_l, \overline{w}_m, \overline{w}_u$ of the alternatives with respect to each criterion are calculated.

Step 7. Vectors of global priorities g_l, g_m, g_u are calculated according to the following formulas

$$\overline{w}_l{}^T = \left[\overline{w}_{1,l}, \overline{w}_{2,l} \ldots \overline{w}_{n,l}\right]^T$$

$$\overline{w}_m{}^T = \left[\overline{w}_{1,m}, \overline{w}_{2,m} \ldots \overline{w}_{n,m}\right]^T$$

$$\overline{w}_u{}^T = \left[\overline{w}_{1,u}, \overline{w}_{2,u} \ldots \overline{w}_{n,u}\right]^T$$

$$g_l = \bar{P}_l \overline{w}_l = \left[g_{1,l}, g_{2,l} \ldots g_{m,l}\right]^T$$

$$g_m = \bar{P}_{1m}\bar{w}_m = \left[g_{1,m}, g_{2,m} \dots g_{m,m}\right]^T$$

$$g_u = \bar{P}_u\bar{w}_u = \left[g_{1,u}, g_{2,u} \dots g_{m,u}\right]^T$$

Step 8. In the last step, the expected value and standard deviation are calculated by the formulas:

$$g_{i,e} = \frac{g_{i,l} + 2g_{i,m} + g_{i,u}}{4}$$

$$\sigma_\iota = \left(\frac{1}{80}\left(3{g_{i,l}}^2 + 4{g_{i,m}}^2 + 3{g_{i,u}}^2 - 4g_{i,l}g_{i,m} - 2g_{i,u}g_{i,l} - 4g_{i,m}g_{i,u}\right)\right)^{1/2}$$

Table 4.9. The steps in calculating the TOPSIS method (Opricovic & Tzeng, 2004)

Step 1. An evaluation matrix with m choices and n criteria is constructed, where the intersection of each choice and criteria is denoted by x_{ij}. This results in the matrix $\left(x_{ij}\right)_{n\times m}$. Decision matrix should be normalized by the following formula:

$$r_{i,j} = \frac{x_{i,j}}{\sqrt{\sum_{i=1}^{m} x_{i,j}^2}}, i = 1,2,\dots,m, j = 1,2,\dots,n$$

Step 2. Weighted normalized decision matrix should be calculated. With the weight $w_j = (w_1, w_2, w_3,\dots, w_n)$, where w_j is the weight of the criteria for all j, $w_j = \frac{W_j}{\sum_{k=1}^{n} W_k}, j = 1,2,\dots,n$ and $\sum_{i=1}^{n} w_i = 1$, $w_j = 1$, The normalization of weight matrix V, is $v_{ij} = w_j * r_{ij}$ i = 1, 2,…., m, j = 1, 2, …., n.

Step 3. Best (A^+) and worst (A^-) matrices of ideal solutions are determined by this formula:

$$A^+ = \left\{\left(max\, v_{ij} \mid j \in J\right), \left(min\, v_{ij} \mid j \in J'\right)\right\}, i = 1,2,..,m = \{V_1^+, V_2^+, \dots, V_n^+\}$$

$$A^- = \left\{\left(min\, v_{ij} \mid j \in J\right), \left(max\, v_{ij} \mid j \in J'\right)\right\}, i = 1,2,..,m = \{V_1^-, V_2^-, \dots, V_n^-\}$$

Step 4. Distance between the target alternative i and the best condition A^+ is represented by symbol (S^+) and formed by the following equation:

$$S_i^+ = \sqrt{\sum_{j=1}^{m} \left(v_{ij} - v_j^+\right)^2}, \qquad i = 1, 2, \ldots, m$$

Contrary to above, (S^-) is the distance between the alternative i from the worst ideal solution and formed as:

$$S_i^- = \sqrt{\sum_{j=1}^{m} \left(v_{ij} - v_j^-\right)^2}, \qquad i = 1, 2, \ldots, m$$

Step 5. Similarity to the best condition should be calculated, as:

$$C_i^+ = \frac{S_i^-}{(S_i^- + S_i^+)}, 0 \leq C_i^- \leq 1, i = 1, 2, \ldots, m$$

Remark. $C_i^+ = 1$, if and only if the alternative ideal solution is the best and $C_i^+ = 0$, the worst one.

Step 6. Alternative C^+should be sorted descended and its largest value is the best ideal solution.

C. The most representative Neuro-Fuzzy model in literature is the Adaptive Neuro-Fuzzy Inference System (ANFIS) and its architecture in Takagi-Sugeno, Tsukamoto, and Mamdani Fuzzy model is described in more detail.

According to ANFIS architecture, the Takagi-Sugeno Fuzzy Model is represented, while for the purpose of simplicity, it is assumed that the under-examined fuzzy inference system has two inputs, x and y, and one output, z. A typical rule set with two fuzzy if-then rules for a first-order Takagi-Sugeno fuzzy model is as follows:

$$Rule\ 1: If\ x\ is\ A_1\ and\ y\ is\ B_1, then\ f_1 = p_{1x} + q_{1y} + r_1;$$

$$Rule\ 2: If\ x\ is\ A_2\ and\ y\ is\ B_2, then\ f_2 = p_{2x} + q_{2y} + r_2;$$

The Takagi-Sugeno model's reasoning process is depicted in Figure 5.5, and the equivalent ANFIS architecture is presented in Figure 5.6, where nodes of the same layer have comparable roles as would be discussed later.

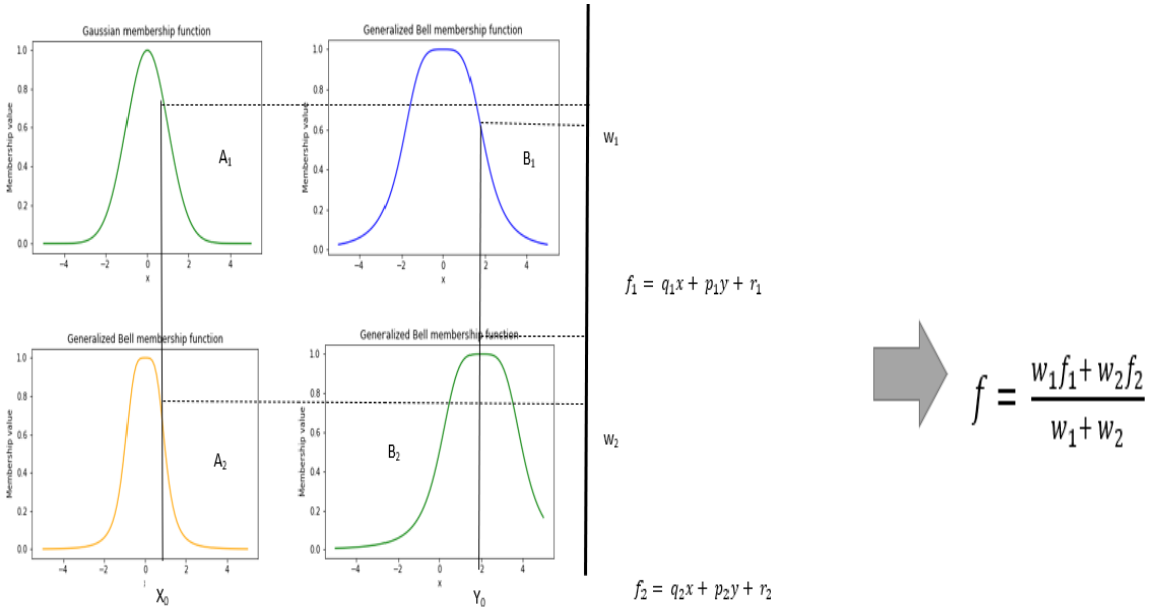


Figure 5.5. Two inputs first order Takagi-Sugeno fuzzy model with two rules is illustrated.

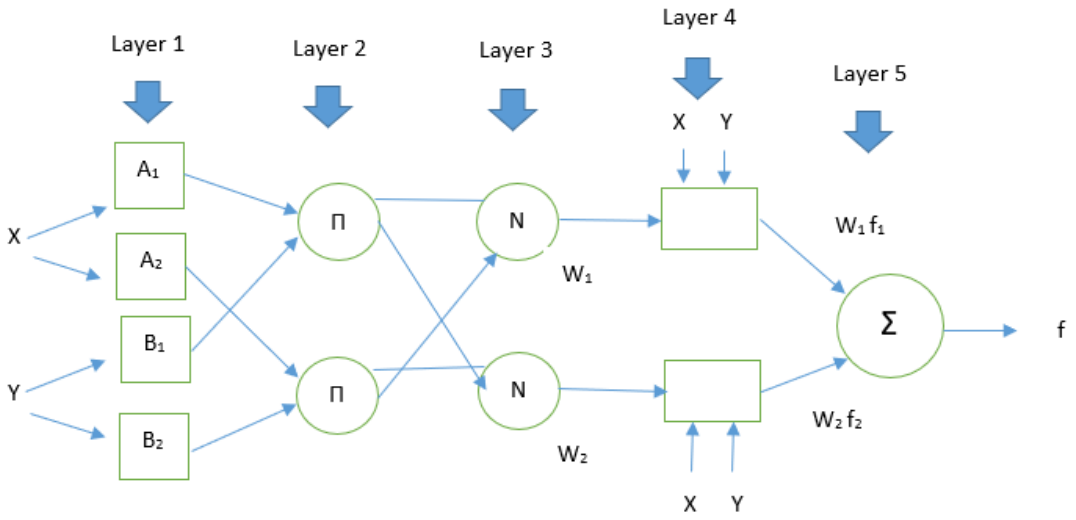


Figure 5.6. The ANFIS architecture is depicted.

Layer 1: Every node I in layer one has a node function and is an adaptive node.

$$O_{1,i} = \mu_{Ai}(x), \quad for\ i = 1,2,$$

$$or$$

$$O_{1,i} = \mu_{Bi-2}(y), \quad for\ i = 3,4,$$

where A_i (or B_{i-2}) is a linguistic label (such as "little" or "big") associated with this node and x (or y) is the input to node i. In other words, $O_{1,i}$ is the degree to which the input x (or y) satisfies the quantifier A and is the membership grade of a fuzzy set A (=A_1, A_2, B_1 or B_2).

$$\mu_A(x) = \frac{1}{1 + \left|\frac{x - c_i}{\alpha_i}\right|^{2b}},$$

the parameter set being "a_i, b_i, c_i." These parameters' values affect how the bell-shaped function behaves, which displays different membership functions for fuzzy set A. Premise parameters are what are included in this layer.

Layer 2: This layer's nodes are all fixed nodes with the label "anfis," and its output is the sum of all incoming signals:

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y), \quad i = 1,2.$$

The strength of a rule's firing is represented by each node output. In general, the node function in this layer could be any other T-norm operator that does fuzzy AND.

Layer 3: This layer's nodes are all fixed nodes with the designation N. The i^{th} node determines the ratio of the firing strength of the i^{th} rule to the total firing strength of all rules:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2.$$

The outputs of this layer are referred to as normalized firing strengths for convenience.

Layer 4: Each node I in this layer has a node function and is adaptive:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_{ix} + q_{iy} + r_i)$$

where p_i, q_i, and r_i represent this node's parameter set and anfis is a normalized firing strength from layer 3. Consequential parameters are the name given to the parameters in this layer.

Layer 5 A fixed node with the name "anfis" is the only node in this layer, and it computes the output as the sum of all input signals:

$$overall\ output = O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

As a result, an adaptive network that is functionally comparable to a Sugeno fuzzy model has been created (Jang, 1993).

Regarding ANFIS architecture of the Tsukamoto Fuzzy Model, in Figures 5.7 and 5.8 the defuzzification process for the Tsukamoto FIS is represented.

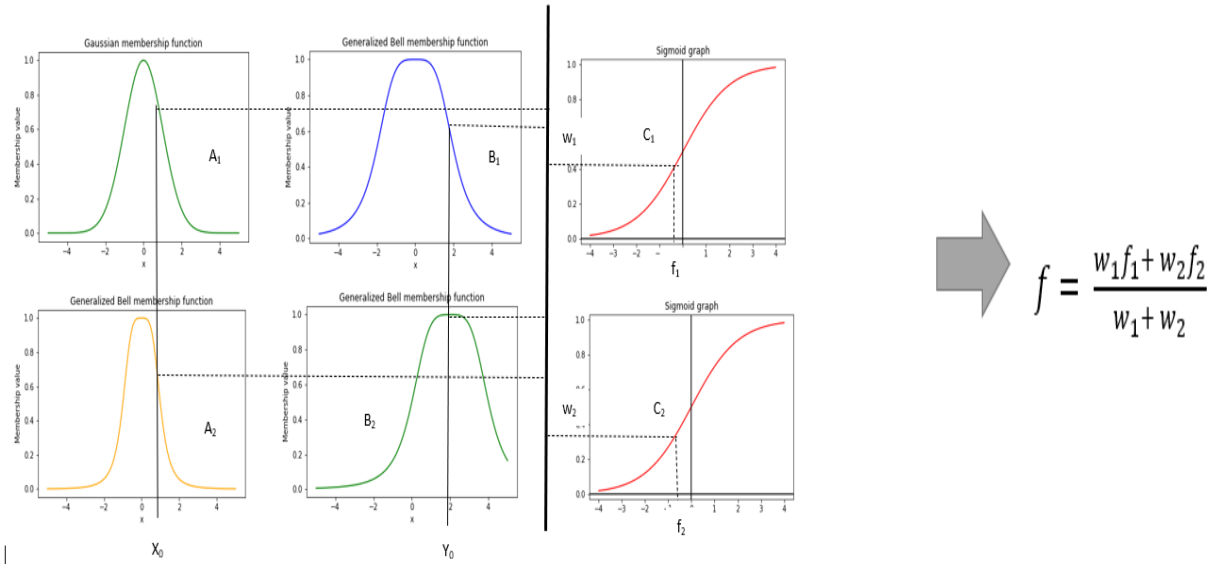


Figure 5.7. Tsukamoto defuzzification.

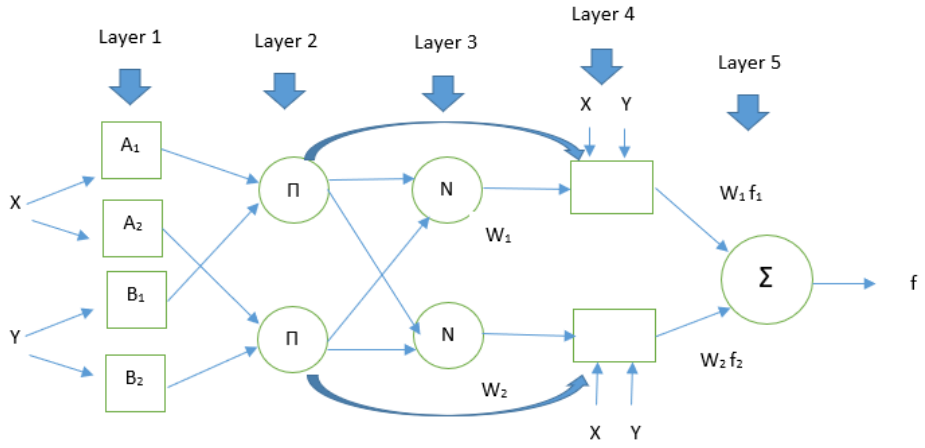


Figure 5.8. ANFIS based on Tsukamoto.

As shown in Figure 5.8, the system is comparable to a Sugeno-based ANFIS, with the exception that a weighted membership function is utilized rather than a linear membership function. As a result, all the layers are identical to those in the Sugeno-based example, with the exception of the last layer, where the defuzzification equation is changed (Singh & Lone, 2020).

As ANFIS architecture of the Mamdani Fuzzy Model is concerned, should be noticed that if the integrals in the centroid defuzzification approach shown here are replaced with discrete approximations, a corresponding ANFIS for the Mamdani fuzzy inference system with max-min composition might be built. The resultant ANFIS, however, is a lot more intricate than either the TS or Tsukamoto ANFIS. Mamdani ANFIS with max-min composition has a more complicated structure and processing, but this does not always translate to superior learning ability or approximation capacity. A Mamdani fuzzy model could be readily defuzzed using sum-product composition and centroid defuzzification, and the associated ANFIS could be created straight from the theorem without any approximation at all (Jang, 1993).

D. Computational experiment and results of the proposed Neuro-Fuzzy model are illustrated in the following tables. As a consequence, tables contain variables of datasets, fuzzy scale, pairwise comparison matrices, fuzzy weights, metrics output, and structures of Neural Network models in three datasets, and finally TOPSIS results for model selection based on fuzzy weights in three datasets.

Table 6.1. Australian credit approval dataset.
(All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data).

Features	Comments
A1	Categorical (0,1)
A2	Continuous
A3	Continuous
A4	Categorical (1,2,3)
A5	Categorical (1 to 14)
A6	Categorical (1, 2,3, 4,5,6,7,8,9)
A7	continuous.
A8	Categorical (1, 0)
A9	Categorical (1, 0)
A10	Continuous
A11	Categorical (1, 0)
A12	Categorical (1, 2, 3)
A13	Continuous
A14	Continuous
A15	Categorical (1,2)

Table 6.2. Real-world dataset

Features	Type
actv_days_sinc_lst_momo_acvty	Continuous
actv_mnths_sinc_lst_momo_acvty	Continuous
momo_cnt_cashout_p2p_1m	Continuous
momo_cnt_cashout_p2p_3m	Continuous
momo_cnt_cashout_p2p_6m	Continuous
momo_cnt_airtime_1m	Continuous
momo_cnt_cashin_deposit_1m	Continuous
momo_cnt_cashin_p2p_1m	Continuous
momo_cnt_cashout_widthrw_1m	Continuous
momo_amt_bundle_1m	Continuous
momo_amt_cashout_p2p_1m	Continuous
momo_amt_cashout_p2p_3m	Continuous
momo_amt_cashout_p2p_6m	Continuous
momo_amt_airtime_1m	Continuous
momo_amt_cashin_b2p_1m	Continuous
momo_amt_cashin_deposit_1m	Continuous
momo_amt_cashin_deposit_3m	Continuous
momo_amt_cashin_deposit_6m	Continuous
momo_amt_cashin_p2p_1m	Continuous
momo_amt_cashin_p2p_3m	Continuous
momo_amt_cashout_widthrw_1m	Continuous

momo_amt_cashout_widthrw_3m	Continuous
momo_amt_cashout_p2b_1m	Continuous
momo_amt_cashout_p2b_3m	Continuous
momo_amt_cashout_p2b_6m	Continuous
momo_fee_cashout_p2p_1m	Continuous
momo_fee_cashout_p2p_3m	Continuous
momo_fee_cashout_p2p_6m	Continuous
momo_fee_cashout_widthrw_1m	Continuous
momo_fee_cashout_p2b_1m	Continuous
momo_fee_cashout_p2b_3m	Continuous
momo_acc_bal_bundle_1m	Continuous
momo_acc_bal_bundle_3m	Continuous
momo_acc_bal_bundle_6m	Continuous
momo_acc_bal_cashout_p2p_1m	Continuous
momo_acc_bal_cashout_p2p_3m	Continuous
momo_acc_bal_airtime_1m	Continuous
momo_acc_bal_cashin_deposit_1m	Continuous
momo_acc_bal_cashin_deposit_3m	Continuous
momo_acc_bal_cashin_p2p_1m	Continuous
momo_acc_bal_cashin_p2p_3m	Continuous
momo_acc_bal_cashin_p2p_6m	Continuous
momo_acc_bal_cashout_widthrw_1m	Continuous
momo_acc_bal_cashout_widthrw_3m	Continuous
momo_acc_bal_cashout_p2b_1m	Continuous
momo_acc_bal_cashout_p2b_3m	Continuous
momo_cnt_tot_1m	Continuous
momo_cnt_tot_3m	Continuous
momo_cnt_tot_6m	Continuous
momo_amt_tot_1m	Continuous
momo_amt_tot_3m	Continuous
momo_amt_tot_6m	Continuous
momo_fee_tot_1m	Continuous
momo_fee_tot_3m	Continuous
momo_fee_tot_6m	Continuous
default_flag	Categorical

Table 6.3. Credit risk dataset of Kaggle.

Features names	Description	Type
person_age	Age	Continuous
person_income	Annual Income	Continuous
person_home_ownership	Home ownership	Categorical (0,1,2,3)
person_emp_length	Employment length (in years)	Continuous
loan_intent	Loan intent	Categorical (0,1,2,3,4,5)
loan_grade	Loan grade	Categorical (0,1,2,3,4,5,6)
loan_amnt	Loan amount	Continuous
loan_int_rate	Interest rate	Continuous
loan_percent_income	Percent income	Continuous
cb_person_default_on_file	Historical default	Categorical (0,1)

cb_preson_cred_hist_length	Credit history length	Continuous
loan_status	Loan status	Categorical (0,1)

Table 6.4. Linguistic terms and the corresponding triangular fuzzy numbers (Harker & Vargas, 1987; Ayhan, 2013).

Saaty scale	Definition	Fuzzy Triangular Scale
1	Equally important (Eq. Imp.)	(1,1,1)
3	Weakly important (W. Imp.)	(2,3,4)
5	Fairly important (F. Imp.)	(4,5,6)
7	Strongly important (S. Imp.)	(6,7,8)
9	Absolutely important (A. Imp.)	(9,9,9)
2	The intermittent values between two adjacent scales	(1,2,3)
4		(3,4,5)
6		(5,6,7)
8		(7,8,9)

6.5. Pair-wise comparison matrix of AUC decision maker supporter.

Evaluation metrics	AUC	Accuracy	F1-score	H-measure
AUC	[1, 1, 1]	[1, 2, 3]	[2, 3, 4]	[3, 4, 5]
Accuracy	[0.33, 0.5, 1.0]	[1, 1, 1]	[1, 2, 3]	[3, 4, 5]
F1-score	[0.25, 0.33, 0.5]	[0.33, 0.5, 1.0]	[1, 1, 1]	[1, 2, 3]
H-measure	[0.2, 0.25, 0.33]	[0.2, 0.25, 0.33]	[0.33, 0.5, 1.0]	[1, 1, 1]

6.6. Pair-wise comparison matrix of Accuracy decision maker supporter.

Evaluation metrics	AUC	Accuracy	F1-score	H-measure
AUC	[1, 1, 1]	[0.2, 0.25, 0.33]	[1, 2, 3]	[2, 3, 4]
Accuracy	[3, 4, 5]	[1, 1, 1]	[2, 3, 4]	[4, 5, 6]
F1-score	[0.33, 0.5, 1.0]	[0.25, 0.33, 0.5]	[1, 1, 1]	[1, 2, 3]
H-measure	[0.25, 0.33, 0.5]	[0.17, 0.2, 0.25]	[0.33, 0.5, 1.0]	[1, 1, 1]

6.7. Pair-wise comparison matrix of F1-score decision maker supporter.

Evaluation metrics	AUC	Accuracy	F1-score	H-measure
AUC	[1, 1, 1]	[0.33, 0.5, 1.0]	[0.33, 0.5, 1.0]	[2, 3, 4]
Accuracy	[1, 2, 3]	[1, 1, 1]	[0.33, 0.5, 1.0]	[3, 4, 5]
F1-score	[1, 2, 3]	[1, 2, 3]	[1, 1, 1]	[3, 4, 5]
H-measure	[0.25, 0.33, 0.5]	[0.2, 0.25, 0.33]	[0.2, 0.25, 0.33]	[1, 1, 1]

6.8. Pair-wise comparison matrix of H-measure decision maker supporter.

Evaluation metrics	AUC	Accuracy	F1-score	H-measure
AUC	[1, 1, 1]	[1, 2, 3]	[1, 2, 3]	[0.2, 0.25, 0.33]
Accuracy	[0.33, 0.5, 1.0]	[1, 1, 1]	[0.33, 0.5, 1.0]	[0.17, 0.2, 0.25]
F1-score	[0.33, 0.5, 1.0]	[1, 2, 3]	[1, 1, 1]	[0.17, 0.2, 0.25]
H-measure	[3, 4, 5]	[4, 5, 6]	[4, 5, 6]	[1, 1, 1]

Table 6.9. Evaluation metrics of 1-layer NN in Australian credit approval.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.901515	0.909091	0.901515	0.909091
Precision	0.823529	0.894737	0.879310	0.857143
Recall	0.982456	0.894737	0.894737	0.947368
F1-score	0.896000	0.894737	0.886957	0.900000
AUC	0.911228	0.907368	0.900702	0.913684
H-measure	0.714584	0.713569	0.692092	0.725220

Table 6.10. Evaluation metrics of 2-layer NN in Australian credit approval.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.901515	0.901515	0.893939	0.901515
Precision	0.879310	0.879310	0.877193	0.866667
Recall	0.894737	0.894737	0.877193	0.912281
F1-score	0.886957	0.886957	0.877193	0.888889
AUC	0.900702	0.900702	0.891930	0.902807
H-measure	0.692092	0.692092	0.667468	0.695758

Table 6.11. Evaluation metrics of 3-layer NN in Australian credit approval.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.568182	0.568182	0.916667	0.909091
Precision	0.000000	0.000000	0.859375	0.881356
Recall	0.000000	0.000000	0.964912	0.912281
F1-score	0.000000	0.000000	0.909091	0.896552
AUC	0.500000	0.500000	0.922456	0.909474
H-measure	0.000000	0.000000	0.750781	0.717028

Table 6.12. The best hyperparameters of 1-Layer NN in Australian credit approval dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units	39	40	30	39
Alpha of LeakyRelu	0.023783	0.056486	0.069636	0.087344
Dropout	0.465079	0.321456	0.418900	0.476562
Learning Rate	0.000291	0.000326	0.001384	0.000204

Batch Size	16	16	16	16
Kernel Regularizers	0.000160	0.000178	0.001046	0.000437
Bias Regularizers	0.007385	0.001872	0.000015	0.001286
Time (seconds)	35.12	1209.99	1342.69	1078.418

Table 6.13. The best hyperparameters of 2-Layer NN in Australian credit approval dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	27	7	31	23
Units (2nd layer)	33	28	34	18
Alpha of LeakyRelu (1st layer)	0.085426	0.071215	0.063457	0.075391
Alpha of LeakyRelu (2nd layer)	0.085948	0.066952	0.080776	0.071172
Dropout (1st layer)	0.20192	0.495251	0.316344	0.460156
Dropout (2nd layer)	0.219817	0.472364	0.489855	0.267969
Kernel Regularizers (1st layer)	0.001011	0.007377	0.000129	0.000217
Kernel Regularizers (2nd layer)	0.000001	0.000432	0.001115	0.000053
Bias Regularizers (1st layer)	0.000331	0.000082	0.000015	0.000018
Bias Regularizers (2nd layer)	0.001770	0.000026	0.000927	0.000012
Learning Rate	0.000189	0.001613	0.000212	0.000625
Batch Size	16	256	32	16
Time (seconds)	26.77	1314.15	1148.92	4589.11

Table 6.14. The best hyperparameters of 3-Layer NN in Australian credit approval dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	9	33	27	11
Units (2nd layer)	29	11	31	31
Units (3rd layer)	35	36	30	31
Alpha of LeakyRelu (1st layer)	0.057629	0.077506	0.088733	0.077500
Alpha of LeakyRelu (2nd layer)	0.071648	0.050823	0.030004	0.032500
Alpha of LeakyRelu (3rd layer)	0.070544	0.017658	0.024256	0.032500
Dropout (1st layer)	0.487013	0.379613	0.406850	0.425000
Dropout (2nd layer)	0.407752	0.226256	0.295919	0.275000
Dropout (3rd layer)	0.371759	0.426354	0.393775	0.275000
Kernel Regularizers (1st layer)	0.006568	0.000027	0.001351	0.000056

Kernel Regularizers (2nd layer)	0.000489	0.000031	0.002079	0.000056
Kernel Regularizers (3rd layer)	0.000010	0.002230	0.000012	0.000056
Bias Regularizers (1st layer)	0.000039	0.000047	0.002242	0.001778
Bias Regularizers (2nd layer)	0.000778	0.000160	0.000020	0.001778
Bias Regularizers (3rd layer)	0.003376	0.000321	0.005832	0.000056
Learning Rate	0.000058	0.000316	0.000272	0.002659
Batch Size	64	32	32	256
Time (seconds)	28.85	1531.23	4568.51	1553.07

Table 6.15. Evaluation metrics of 1-layer NN in Credit risk dataset of Kaggle.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0. 864930	0.864415	0.866472	0.864415
Precision	0.774862	0.764391	0.767411	0.775453
Recall	0.473019	0.481450	0.492411	0.468803
F1-score	0.587435	0.590792	0.599897	0.584341
AUC	0.718975	0.721792	0.727165	0.717082
H-measure	0.334284	0.337359	0.347785	0.330960

Table 6.16. Evaluation metrics of 2-layer NN in Credit risk dataset of Kaggle.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.895955	0.900583	0.904354	0.907439
Precision	0.826381	0.915068	0.931319	0.887290
Recall	0.618044	0.563238	0.571669	0.623946
F1-score	0.707188	0.697286	0.708464	0.732673
AUC	0.792456	0.774949	0.780456	0.801861
H-measure	0.487065	0.466846	0.480633	0.518255

Table 6.17. Evaluation metrics of 3-layer NN in Credit risk dataset of Kaggle.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.912239	0.903668	0.906754	0.904011
Precision	0.907005	0.896947	0.873256	0.938375
Recall	0.633221	0.594435	0.633221	0.564924
F1-score	0.745780	0.715010	0.734115	0.705263
AUC	0.808327	0.788504	0.804885	0.777729
H-measure	0.535375	0.491948	0.522100	0.476017

Table 6.18. The best hyperparameters of 1-Layer NN in Credit risk dataset of Kaggle.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units	30	1213	10	14
Alpha of LeakyRelu	0.021675	0.026858	0.036926	0.026875
Dropout	0.436936	0.277678	0.224593	0.293750
Learning Rate	0.000143	0.000281	0.000414	0.000985
Batch Size	256	256	256	512
Kernel Regularizers	0.006809	0.000802	0.000743	0.006494
Bias Regularizers	0.000258	0.000777	0.002538	0.000205

Table 6.19. The best hyperparameters of 2-Layer NN in Credit risk dataset of Kaggle.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	37	37	34	40
Units (2nd layer)	25	23	21	40
Alpha of LeakyRelu (1st layer)	0.052106	0.033495	0.040039	0.058516
Alpha of LeakyRelu (2nd layer)	0.021953	0.011575	0.062107	0.020547
Dropout (1st layer)	0.407296	0.211454	0.293840	0.216406
Dropout (2nd layer)	0.253895	0.454297	0.215003	0.249219
Kernel Regularizers (1st layer)	0.000093	0.000012	0.000144	0.001877
Kernel Regularizers (2nd layer)	0.000333	0.000534	0.000030	0.000461
Bias Regularizers (1st layer)	0.000018	0.000418	0.000067	0.000882
Bias Regularizers (2nd layer)	0.000404	0.003113	0.005658	0.000018
Learning Rate	0.001683	0.004086	0.006154	0.000870
Batch Size	32	512	32	32

Table 6.20. The best hyperparameters of 3-Layer NN in Credit risk dataset of Kaggle.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	39	39	24	32
Units (2nd layer)	27	37	24	32
Units (3rd layer)	21	36	22	27
Alpha of LeakyRelu (1st layer)	0.034490	0.024722	0.046519	0.074688
Alpha of LeakyRelu (2nd layer)	0.060273	0.042851	0.044781	0.029688
Alpha of LeakyRelu (3rd layer)	0.031091	0.041160	0.092553	0.012813

Dropout (1st layer)	0.232611	0.234990	0.252576	0.228125
Dropout (2nd layer)	0.295774	0.306068	0.257878	0.284375
Dropout (3rd layer)	0.341761	0.309852	0.332012	0.471875
Kernel Regularizers (1st layer)	0.001831	0.000405	0.000021	0.000107
Kernel Regularizers (2nd layer)	0.000027	0.000023	0.000048	0.002207
Kernel Regularizers (3rd layer)	0.000460	0.005663	0.002540	0.000604
Bias Regularizers (1st layer)	0.000268	0.000070	0.000027	0.001433
Bias Regularizers (2nd layer)	0.000017	0.002519	0.000186	0.000255
Bias Regularizers (3rd layer)	0.000136	0.000123	0.001747	0.000165
Learning Rate	0.002194	0.001205	0.006756	0.001162
Batch Size	128	256	256	16
Time (seconds)	83.18	101.11	108.67	99.1

Table 6.21. Evaluation metrics of 1-layer NN in Real-world dataset.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.618030	0.611686	0.607346	0.614802
Precision	0.610110	0.599228	0.588026	0.603204
Recall	0.689429	0.713723	0.760998	0.708689
F1-score	0.647349	0.651483	0.663423	0.651706
AUC	0.616793	0.609919	0.604684	0.613176
H-measure	0.066120	0.059947	0.057617	0.063094

Table 6.22. Evaluation metrics of 2-layer NN in Real-world dataset.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.683695	0.679688	0.673344	0.680690
Precision	0.726462	0.723382	0.693878	0.753883
Recall	0.606260	0.599256	0.639965	0.552418
F1-score	0.660940	0.655494	0.665832	0.637615
AUC	0.685036	0.681082	0.673923	0.682912
H-measure	0.163509	0.157039	0.142419	0.166114

Table 6.23. Evaluation metrics of 3-layer NN in Real-world dataset.

Metric	T-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Accuracy	0.680022	0.678019	0.671675	0.668447
Precision	0.702052	0.714542	0.744932	0.771516
Recall	0.644123	0.610856	0.538849	0.494419
F1-score	0.671841	0.658643	0.625349	0.602641

AUC	0.680644	0.679182	0.673976	0.671462
H-measure	0.153476	0.152768	0.151431	0.154668

Table 6.24. The best hyperparameters of 1-Layer NN in Real-world dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units	38	40	4	19
Alpha of LeakyRelu	0.095879	0.086744	0.091024	0.096484
Dropout	0.364806	0.366596	0.221464	0.464844
Learning Rate	0.001949	0.000072	0.002296	0.000413
Batch Size	128	1024	32	256
Kernel Regularizers	0.000030	0.009885	0.000202	0.000241
Bias Regularizers	0.000102	0.000389	0.000038	0.000141

Table 6.25. The best hyperparameters of 2-Layer NN in Real-world dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	40.000000	39.000000	36.000000	35.000000
Units (2nd layer)	40.000000	36.000000	27.000000	39.000000
Alpha of LeakyRelu (1st layer)	0.070235	0.010615	0.049384	0.045156
Alpha of LeakyRelu (2nd layer)	0.054595	0.076800	0.027080	0.064844
Dropout (1st layer)	0.200386	0.252838	0.281791	0.298438
Dropout (2nd layer)	0.460749	0.486565	0.352922	0.232813
Kernel Regularizers (1st layer)	0.000279	0.000026	0.000105	0.000120
Kernel Regularizers (2nd layer)	0.000017	0.000831	0.000174	0.000050
Bias Regularizers (1st layer)	0.000184	0.000233	0.003863	0.000184
Bias Regularizers (2nd layer)	0.000457	0.007148	0.000104	0.000284
Learning Rate	0.000733	0.002072	0.000580	0.002074
Batch Size	256.000000	32.000000	64.000000	1024.000000
Time (sec.)	123.000000	145.820000	156.160000	119.370000

Table 6.26. The best hyperparameters of 3-Layer NN in Real-world dataset.

Hyperparameters	Tree-str Parzen Est.	Random Sampler	Genetic Algorithm	Quasi Monte Carlo
Units (1st layer)	35	27	26	39
Units (2nd layer)	40	28	17	35
Units (3rd layer)	26	14	16	37

Alpha of LeakyRelu (1st layer)	0.016079	0.013365	0.057078	0.033906
Alpha of LeakyRelu (2nd layer)	0.085886	0.095177	0.096936	0.059219
Alpha of LeakyRelu (3rd layer)	0.053844	0.042875	0.015241	0.073281
Dropout (1st layer)	0.207566	0.293543	0.444847	0.392188
Dropout (2nd layer)	0.344614	0.319715	0.202576	0.223438
Dropout (3rd layer)	0.431114	0.237113	0.216964	0.467187
Kernel Regularizers (1st layer)	0.000015	0.000038	0.000011	0.000011
Kernel Regularizers (2nd layer)	0.000191	0.000047	0.000030	0.000021
Kernel Regularizers (3rd layer)	0.000104	0.000409	0.007039	0.000284
Bias Regularizers (1st layer)	0.000198	0.000648	0.000017	0.000050
Bias Regularizers (2nd layer)	0.000014	0.000460	0.000133	0.000229
Bias Regularizers (3rd layer)	0.006699	0. 007598	0.000090	0.000063
Learning Rate	0.000566	0.001790	0.001566	0.000552
Batch Size	32	512	128	32

6.27. Results of FAHP based on 1st expert (AUC).

	Lower Bound	Medium Bound	Upper Bound	Expected Value	Standard Deviation
AUC	0.338767	0.455194	0.619706	0.467216	0.044744
Accuracy	0.220523	0.296097	0.417592	0.307577	0.031579
F1 score	0.115728	0.159414	0.232229	0.166696	0.018706
H-measure	0.070381	0.089295	0.123837	0.093202	0.008631

6.28. Results of FAHP based on 2nd expert (Accuracy).

	Lower Bound	Medium Bound	Upper Bound	Expected Value	Standard Deviation
AUC	0.159202	0.217721	0.302435	0. 224270	0.022836
Accuracy	0.433319	0.547381	0.708841	0.559231	0.043885
F1 score	0.109038	0.150256	0.217175	0.156681	0.017338
H-measure	0.066339	0.084641	0.118327	0.088487	0.008398

6.29. Results of FAHP based on 3rd expert (F1-score).

	Lower Bound	Medium Bound	Upper Bound	Expected Value	Standard Deviation
AUC	0.149870	0.201420	0.292367	0.211269	0.022957
Accuracy	0.224864	0.301002	0.420900	0.311942	0.031380
F1 score	0.301868	0.417805	0. 575832	0.428327	0.043572

H-measure	0.065436	0.079773	0.104156	0.082285	0.006224

6.30. Results of FAHP based on 4th expert (H-measure).

	Lower Bound	Medium Bound	Upper Bound	Expected Value	Standard Deviation
AUC	0.130892	0.188050	0.271122	0.194528	0.022361
Accuracy	0.070076	0.093385	0.137957	0.098701	0.010993
F1 score	0.093523	0.129608	0.189467	0.135551	0.015401
H-measure	0.468858	0.588958	0.765456	0.603058	0.047318

Table 6.31. Australian credit approval results of TOPSIS with priority in AUC.

	C*	S-	S*
1 Hidden Layer MPL	0. 576346	0. 004995	0. 003672
2 Hidden Layer MPL	0.000000	0.000000	0. 008650
3 Hidden Layer MPL	1.000000	0. 008650	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.32. Australian credit approval results of TOPSIS with priority in Accuracy.

	C*	S^-	S*
1 Hidden Layer MPL	0.540528	0.004223	0.003590
2 Hidden Layer MPL	0.000000	0.000000	0.007788
3 Hidden Layer MPL	1.000000	0.007788	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.33. Australian credit approval results of TOPSIS with priority in F1-score.

	C*	S^-	S*
1 Hidden Layer MPL	0. 573024	0. 004784	0. 003564
2 Hidden Layer MPL	0.000000	0.000000	0. 008333
3 Hidden Layer MPL	1.000000	0. 008333	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.34. Australian credit approval results of TOPSIS with priority in H-measure.

	C*	S^-	S*
1 Hidden Layer MPL	0.56479	0.016078	0.012390
2 Hidden Layer MPL	0.00000	0.000000	0.028467

3 Hidden Layer MPL	1.00000	0.028467	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.35. Credit risk of Kaggle results of TOPSIS with priority in AUC.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.041585
2 Hidden Layer MPL	0.913466	0.037997	0.003600
3 Hidden Layer MPL	1.000000	0.041585	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.36. Credit risk of Kaggle results of TOPSIS with priority in Accuracy.

	C*	S^-	S*
1 Hidden Layer MPL	0.00000	0.000000	0.034940
2 Hidden Layer MPL	0.90755	0.031722	0.003231
3 Hidden Layer MPL	1.00000	0.034940	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.37. Credit risk of Kaggle results of TOPSIS with priority in F1-score.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.057320
2 Hidden Layer MPL	0.910062	0.052168	0.005156
3 Hidden Layer MPL	1.000000	0.057320	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.38. Credit risk of Kaggle results of TOPSIS with priority in H-measure.

	C*	S^-	S*
1 Hidden Layer MPL	0.00000	0.000000	0.139072
2 Hidden Layer MPL	0.90883	0.126394	0.012679
3 Hidden Layer MPL	1.00000	0.139072	0.000000

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.39. Real-world results of TOPSIS with priority in AUC.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.051041
2 Hidden Layer MPL	0.969753	0.050955	0.001589
3 Hidden Layer MPL	0.912240	0.046695	0.004492

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.40. Real-world results of TOPSIS with priority in Accuracy.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.050757
2 Hidden Layer MPL	0.971368	0.050680	0.001494
3 Hidden Layer MPL	0.915850	0.046655	0.004287

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.41. Real-world results of TOPSIS with priority in F1-score.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.041679
2 Hidden Layer MPL	0.909367	0.040974	0.004084
3 Hidden Layer MPL	0.910248	0.038119	0.003759

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

Table 6.42. Real-world results of TOPSIS with priority in H-measure.

	C*	S^-	S*
1 Hidden Layer MPL	0.000000	0.000000	0.251551
2 Hidden Layer MPL	0.994888	0.251540	0.001292
3 Hidden Layer MPL	0.897067	0.225665	0.025894

Where C*: Similarity to the best condition, S*: Distance between the target alternative i and the best condition and S^-: Distance between the alternative i from the worst ideal solution

References

Abraham, A. (2001). Neuro Fuzzy Systems: State-of-the-Art Modeling Techniques. Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, 269–276. https://doi.org/10.1007/3-540-45720-8_30

Akkoç, S. (2019). Exploring the nature of credit scoring: a neuro fuzzy approach. *Fuzzy Economic Review, 24(1),* 3-24. DOI:10.25102/fer.2019.01.01

Aliyev, R., Temizkan, H., & Aliyev, R. (2020). Fuzzy Analytic Hierarchy Process-Based Multi-Criteria Decision Making for Universities Ranking. *Symmetry, 12(8),* 1351. https://doi.org/10.3390/sym12081351

Audi, R. (1999). *The Cambridge Dictionary of Philosophy (2nd ed.).* Cambridge University Press.

Ayhan, M. B. (2013). A Fuzzy Ahp Approach For Supplier Selection Problem: A Case Study In A Gearmotor Company. *International Journal of Managing Value and Supply Chains, 4(3),* 11–23. https://doi.org/10.5121/ijmvsc.2013.4302

Bendre, N., Ebadi, N., Prevost, J. J., & Najafirad, P. (2020). Human Action Performance Using Deep Neuro-Fuzzy Recurrent Attention Model. *IEEE Access, 8*, 57749–57761. https://doi.org/10.1109/access.2020.2982364

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., & Lin, Z. (2016). *Towards Biologically Plausible Deep Learning.* ArXiv:1502.04156 [Cs]. https://arxiv.org/abs/1502.04156

Berenji, H. R., & Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks, 3(5),* 724–740. https://doi.org/10.1109/72.159061

Bergstra, J., Yamins, D., & Cox, D. (2013, February 13). *Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures.* Proceedings.mlr.press; PMLR. https://proceedings.mlr.press/v28/bergstra13.html

Bessis, J. (2015). *Risk management in banking*. John Wiley & Sons.

Blalock, D., Ortiz, J. J. G., Frankle, J., & Guttag, J. (2020). *What is the State of Neural Network Pruning?* ArXiv:2003.03033 [Cs, Stat]. https://arxiv.org/abs/2003.03033

Bobzien, S. (2002). The Development of Modus Ponens in Antiquity: From Aristotle to the 2nd Century AD. *Phronesis, 47(4),* 359–394. https://doi.org/10.1163/156852802321016541

Burkov, A. (2019). *The Hundred- Page Machine Learning Book*. http://ema.cri-info.cm/wp-content/uploads/2019/07/2019BurkovTheHundred-page_Machine Learning.pdf

Carter, J., Chiclana, F., Khuman, A. S., & Chen, T. (Eds.) (2021). *Fuzzy Logic: Recent Applications and Developments. (1 ed.)* Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-66474-9

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16(16),* 321–357. https://doi.org/10.1613/jair.953

Chen, M.-C., & Huang, S.-H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24(4),* 433–441. https://doi.org/10.1016/s0957-4174(02)00191-4

Chia-Feng Juang, & Chin-Teng Lin. (1998). An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems, 6(1),* 12–32. https://doi.org/10.1109/91.660805

Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

Crouhy, M., Galai, D., & Mark, R. (2005). *The Essentials of Risk Management (2nd ed.).* McGraw Hill Education Europe.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. https://doi.org/10.1109/4235.996017

Ding, J., & Schmidt, D. (2005). Rainbow, a New Multivariable Polynomial Signature Scheme. *Applied Cryptography and Network Security,* 164–175. https://doi.org/10.1007/11496137_12

Finlay, S. (2012). *Credit Scoring, Response Modeling, and Insurance Rating: A Practical Guide to Forecasting Consumer Behavior 2nd ed*. Palgrave Macmillan.

Freeman, M., & Smith, F. (2013). *Law and Language: Current Legal Issues*. Oxford University Press.

Fullér, R. (2000). Introduction to Neuro-Fuzzy Systems. *Physica-Verlag HD.* 49-50. https://doi.org/10.1007/978-3-7908-1852-9

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.

Hand, D. J., & Anagnostopoulos, C. (2022). Notes on the H-measure of classifier performance. *Advances in Data Analysis and Classification.* https://doi.org/10.1007/s11634-021-00490-3

Harker, P. T., & Vargas, L. G. (1987). The Theory of Ratio Scale Estimation: Saaty's Analytic Hierarchy Process. *Management Science, 33(11),* 1383–1403.

Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1-2), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Horn, P. S., & Kafadar, K. (2006). Trimming and Winsorization. *Encyclopedia of Environmetrics*. https://doi.org/10.1002/9780470057339.vat027

Hull, J. (2018). *Risk management and financial institutions (5th ed.).* John Wiley & Sons, Inc.

Hwang, C.-L., & Yoon, K. (1981). *Multiple Attribute Decision Making*. In Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-48318-9

Irvine, A. D., & Deutsch, H. (2016). *Russell's Paradox* (Stanford Encyclopedia of Philosophy). Stanford.edu. https://plato.stanford.edu/entries/russell-paradox/

Jang, R. (1992). Neuro-Fuzzy Modeling: Architectures, Analyses and Applications *(Doctoral dissertation). University of California, Berkeley*.

Jang, J.-S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, *Man, and Cybernetics, 23(3),* 665–685. https://doi.org/10.1109/21.256541

Jang, J.-S. R., & Chuen-Tsai Sun. (1995). Neuro-fuzzy modeling and control. Proceedings of the IEEE, 83(3), 378–406. https://doi.org/10.1109/5.364486

Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review]. *IEEE Transactions on Automatic Control, 42(10),* 1482–1484. https://doi.org/10.1109/tac.1997.633847

Kabir, G., & Hasin, M. A. A. (2012). Comparative analysis of TOPSIS and Fuzzy TOPSIS for the evaluation of travel website service quality. *International Journal of Quality Research, 6(3),* 169-185.

Kruse, R., & Nauck, D. (1998). Neuro-Fuzzy Systems. Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, 230–259. https://doi.org/10.1007/978-3-642-58930-0_12

Łatuszyńska, A. (2014). Multiple-Criteria Decision Analysis Using Topsis Method For Interval Data In Research Into The Level Of Information Society Development. *Folia Oeconomica Stetin, 13(2),* 63–76.

Lin, C. T., & Lee, C. S. G. (1991). Neural network based fuzzy logic control and decision system. *IEEE Transactions on Computers, 40(12),* 1320-1336. Doi 10.1109/12.106218

Lin, C.-T., Lin, C.-J., & Lee, C. (1995). MItY sets and systems Fuzzy adaptive learning control network with on-line neural learning. *Fuzzy Sets and Systems, 71,* 25–45. https://doi.org/10.1016/0165-0114(94)00131-Q

Łukasiewicz, J. (1951). *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic.* Clarendon Press.

Manevitz, L., Yousef, M., Cristianini, N., Shawe-Taylor, J., & Williamson, B. (2001). One-Class SVMs for Document Classification. *Journal of Machine Learning Research, 2*, 139–154.

Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an Integration of Deep *Learning and Neuroscience. Frontiers*. https://dspace.mit.edu/handle/1721.1/107008

Menabrea, L. (1842). *Sketch of the Analytical Engine Invented by Charles Babbage*. https://johnrhudson.me.uk/computing/Menabrea_Sketch.pdf

Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks, 11(3),* 748–768. https://doi.org/10.1109/72.846746

Moline, J. (1969). *Aristotle, Eubulides and the sorites. Mind*, 78, 393–407.

Mor-Yosef, S., Samueloff, A., Modan, B., Navot, D., & Schenker, J. G. (1990). Ranking the risk factors for cesarean: logistic regression analysis of a nationwide study. *Obstetrics and Gynecology, 75(6),* 944–947. https://pubmed.ncbi.nlm.nih.gov/2342742/

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists.* O'Reilly Media, Inc.

Nauck, D., & Kruse, R. (1993). *A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation*. Retrieved April 13, 2023, from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6b853229347c25182119597511a5a8d358a63343

Nauck, D., Klawonn, F., & Kruse, R. (1997). *Foundations of Neuro-Fuzzy Systems*. Chichester, UK: John Wiley & Sons.

Neal, B. (2019). On the bias-variance tradeoff: Textbooks need an update. arXiv preprint arXiv:1912.08286. https://doi.org/10.48550/arXiv.1912.08286

Nebot, À., & Mugica, F. (2021). Forest Fire Forecasting Using Fuzzy Logic Models. *Forests, 12(8),* 1005. https://doi.org/10.3390/f12081005

Opricovic, S., & Tzeng, G.-H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research, 156(2),* 445–455. https://doi.org/10.1016/s0377-2217(03)00020-1

Owen, A. B. (2003). Quasi-monte carlo sampling. Monte Carlo Ray Tracing: Siggraph, 1, 69-88. https://artowen.su.domains/reports/siggraph03.pdf

Ozkan, I. A. (2020). A Novel Basketball Result Prediction Model Using a Concurrent Neuro-Fuzzy System. *Applied Artificial Intelligence, 34(13),* 1038–1054. https://doi.org/10.1080/08839514.2020.1804229

Pabuçcu, H., & Ayan, T. Y. (2016). The development of an alternative method for the sovereign credit rating system based on adaptive neuro-fuzzy inference system. *Americould Journal of Operations Research, 7(1),* 41-55.

Pasila, F. (2019). Credit Scoring Modeling of Indonesian Micro, Small and Medium Enterprises using Neuro-Fuzzy Algorithm. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* https://doi.org/10.1109/fuzz-ieee.2019.8858841

Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. O'Reilly Media, Inc.

Rana, R. S., & Prasad, (2010). *Exploration of Hybrid Neuro Fuzzy Systems*.

Resti, A., & Sironi, A. (Eds.). (2012). *Risk Management and Shareholders' Value in Banking.* John Wiley & Sons, Inc. https://doi.org/10.1002/9781118371886

Romeijn, H. E. (2008). Random Search Methods. *Encyclopedia of Optimization*, 3245–3251. https://doi.org/10.1007/978-0-387-74759-0_556

Rutkowska, D. (2001). *Neuro-fuzzy architectures and hybrid learning (Vol. 85).* Springer Science & Business Media.

Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling, 9(3-5),* 161–176. https://doi.org/10.1016/0270-0255(87)90473-8

Saaty, T. L. (2008). The analytic hierarchy and analytic network measurement processes: Applications to decisions under risk. *European Journal of Pure and Applied Mathematics, 1(1),* 122-196. DOI: https://doi.org/10.29020/nybg.ejpam.v1i1.6

Shihabudheen, K. V., & Pillai, G. N. (2018). Recent advances in neuro-fuzzy system: A survey. *Knowledge-Based Systems, 152,* 136–162. https://doi.org/10.1016/j.knosys.2018.04.014

Singh, H., & Lone, Y. A. (2020). *Deep neuro-fuzzy systems with python*. Berkeley: Apress.

Sorensen, R. A. (2009). Sorites arguments. In *J. Kim, E. Sosa, & G. S. Rosenkrantz (Eds.), A Companion to Metaphysics (pp. 565)*. John Wiley & Sons.

Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. TELKOMNIKA *(Telecommunication Computing Electronics and Control), 14(4),* 1502. https://doi.org/10.12928/telkomnika.v14i4.3956

Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning, 5(3),* 191–212. https://doi.org/10.1016/0888-613x(91)90008-a

Takagi, H. (1997). Introduction to Fuzzy Systems, Neural Networks, and Genetic Algorithms. *Intelligent Hybrid Systems, 3–33*. https://doi.org/10.1007/978-1-4615-6191-0_1

Talpur, N., Abdulkadir, S. J., Alhussian, H., Hasan, M. H., Aziz, N., & Bamhdi, A. (2022). Deep Neuro-Fuzzy System application trends, challenges, and future

perspectives: a systematic survey. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-022-10188-3

Tano, S., Oyama, T., & Arnould, T. (1996). Deep combination of fuzzy inference and neural network in fuzzy inference software — FINEST. *Fuzzy Sets and Systems, 82(2),* 151–160. https://doi.org/10.1016/0165-0114(95)00251-0

Te, P. (2016). *Risk Management in Banking Principles and Framework*. Oxford Fajar Sdn. Bhd.

Theodoridis, S. (2020). *Machine learning: a Bayesian and optimization perspective*. Elsevier: Academic Press.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley Pub. Co.

Trillas, E., & Valverde, L. (1985). On implication and indistinguishability in the setting of fuzzy logic. *In J. Kacprzyk & R. R. Yager (Eds.), Management decision support systems using fuzzy sets and possibility theory* (pp. 198-212). Verlag TUV Rheinland.

Watanabe, S., & Hutter, F. (2022). c-TPE: Generalizing Tree-structured Parzen Estimator with Inequality Constraints for Continuous and Categorical Hyperparameter Optimization. ArXiv:2211.14411 [Cs]. https://arxiv.org/abs/2211.14411

Whitehead, A. N., & Russell, B. (1962). *Principia Mathematica to *56 (2nd ed.).* The Syndics of the Cambridge University Press.

Yager, R. R. (1986). *A characterization of the extension principle.* Fuzzy Sets and Systems, 18(3), 205–217. https://doi.org/10.1016/0165-0114(86)90002-3

Yea, B., Konishi, R., Osaki, T., & Sugahara, K. (1994). The discrimination of many kinds of odor species using fuzzy reasoning and neural networks. Sensors and Actuators A: *Physical, 45(2),* 159–165. https://doi.org/10.1016/0924-4247(94)00831-0

Yousofi Tezerjan, M., Safi Samghabadi, A., & Memariani, A. (2021). ARF: A hybrid model for credit scoring in complex systems. *Expert Systems with Applications, 185*, 115634. https://doi.org/10.1016/j.eswa.2021.115634

Zadeh, L. A. (1965). Fuzzy sets. *Information and control, 8(3),* 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, & Cybernetics, 3(1),* 28–44. https://doi.org/10.1109/TSMC.1973.5408575

Zadeh, L. A. (1975). *Fuzzy logic and approximate reasoning: In memory of grigore moisil. Synthese,* 30, 407-428.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-III. *Information Sciences, 9(1),* 43–80. https://doi.org/10.1016/0020-0255(75)90017-1

Zadeh, L. A. (1979). A theory of approximate reasoning. In *J. E. Hayes, D. Michie, & L. I. Mikulich (Eds.), Machine intelligence (9), 149-194.* Elsevier.

Zanakis, S. H., Solomon, A., Wishart, N., & Dublish, S. (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research, 107(3),* 507–529. https://doi.org/10.1016/s0377-2217(97)00147-1

Ziemba, P. (2019). Inter-criteria dependencies-based decision support in the sustainable wind energy management. *Energies, 12(4),* 749. https://doi.org/10.3390/en12040749