Neural Networks, Fuzzy Systems and Pattern Recognition: A Comparative Study

Christopher Chao
School of Biomedical Engineering, Shanghai Jiao Tong University, Minhang District, Shanghai, China.
chaochao@160.com

Correspondence should be addressed to Christopher Chao : chaochao@160.com.

Article Info
Journal of Biomedical and Sustainable Healthcare Applications (http://anapub.co.ke/journals/jbsha/jbsha.html)
Doi: https://doi.org/10.53759/0088/JBSHA202303003
Received 10 August 2021; Revised from 14 March 2022; Accepted 23 April 2022.
Available online 05 January 2023.

Published by AnaPub Publications

Abstract – Artificial Intelligence (AI) and Machine Learning (ML) have been rapidly advancing in recent years, with many new techniques and models being developed. One area of AI and ML that has more focuses on Pattern Recognition (PR). PR is a subfield of ML that deals with the identification and classification of patterns in data. This field is closely related to other subfields of AI and ML, such as Neural Networks (NNs) and Neuro-Fuzzy Systems (NFS). NNs are a kind of artificial intelligence inspired by the way our brains work. This paper will provide a comparative research of three fields: Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR), highlighting their similarities and differences. NNs, NFS, and PR are three closely related fields of research in the field of AI and ML. The paper begins with a brief introduction to each of these fields, followed by a discussion of their similarities and differences. NNs are a type of AI that are modeled after the function and structure of the human brain system. They integrate a wide-range of interlinked processing nodes, known as neurons that are used to perform various tasks such as PR and control. NNs are particularly useful for tasks that involve large amounts of data, such as image and speech recognition.

Keywords – Artificial Intelligence (AI), Machine Learning (ML), Neural Networks (NNs), Neuro-Fuzzy Systems (NFS), Pattern Recognition (PR).

I. INTRODUCTION

Neural Networks (NNs) [1] have been widely used in image and speech recognition, as well as other areas such as natural language processing and bioinformatics. Neuro-Fuzzy Systems (NFS) [2] are a type of Artificial Intelligence (AI) [3] that combines the strengths of NNs and Fuzzy Logic (FL). FL is a form of mathematical logic that deals with reasoning under uncertainty. NFS are particularly useful for tasks that involve uncertainty and imprecision, such as decision making and control. They are able to handle both numerical and categorical data, making them more versatile than traditional fuzzy systems. NNs are a type of AI, which are modeled after the function and structure of the brain system. They integrate a larger number of neurons, which are interlinked processing nodes that are used to perform various tasks such as PR and control.

Neural Networks (NNs) have been widely used in image and speech recognition, as well as other areas such as natural language processing and bioinformatics. NNs architectures are feedforward, recurrent and Convolutional Neural Networks. The learning algorithms used in NNs are backpropagation and supervised learning. NFS are a type of AI that combines the strengths of NNs and FL. FL is a form of mathematical logic that deals with reasoning under uncertainty. NFS are particularly useful for tasks that involve uncertainty and imprecision, such as decision making and control. They are able to handle both numerical and categorical data, making them more versatile than traditional fuzzy systems. The learning algorithm used in NFS is adaptive network-based fuzzy inference system (ANFIS) [4].

Pattern Recognition (PR) is a subfield of ML that deals with the identification and classification of patterns in data. This field is closely related to both NNs and NFS, as they all integrate the usage of algorithms, which are meant to identify different patterns in data. They also share a common goal of improving the accuracy and efficiency of PR. In terms of differences, NNs are more specialized for tasks involving large amounts of data, while NFS are more specialized for tasks involving uncertainty and imprecision. Additionally, PR is a subfield of ML, while NNs and NFS are forms of AI.

In this paper, we aim to provide an in-depth study of these three fields of AI and ML: NNs, NFS, and PR. We will first introduce each of these fields in more detail, including their architectures, algorithms, and applications. We will then discuss their similarities and differences, and provide examples of how each of these fields has been applied in real-world problems. This study will serve as a comprehensive guide for researchers and practitioners working in these fields, providing a comprehension of the limitations and strengths of these approaches. In that regard, this paper is organized as follows: Section II provides a background analysis of this paper, which acts as a roadmap for the entire review. Section III focuses on providing a detailed literature review for Neural Networks, Neuro-Fuzzy Systems, and Pattern Recognition. Section IV provides detailed analysis of the architectures and application of these technologies, while Section V provides a review of their similarities and differences. Lastly, Section VI concludes the paper and recommends future research directions.

II. BACKGROUND ANALYSIS

To provide a comprehensive comparative study of Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR), we will use a combination of literature review and case studies. First, we will conduct a literature review of the relevant research in these fields. This will involve searching for and reviewing relevant academic papers, books, and other resources. We will focus on the most recent and relevant research, with a focus on papers published in the last five years. We will also consider the historical development of these fields, in order to understand their evolution and current state.

Next, we will provide an in-depth description of the architectures, and applications of each of these fields. For NNs, we will describe feedforward, recurrent and convolutional architectures, and the learning algorithms such as backpropagation and supervised learning. For NFS, we will describe the use of FL and the learning algorithm ANFIS. For PR, we will describe various algorithms used for feature extraction, dimensionality reduction, classification, and clustering.

We will also provide examples of how each of these fields has been applied in real-world problems. This will include case studies/examples of successful applications in areas such as speech, and image recognition, bioinformatics, and natural language processing etc. These examples will serve to demonstrate the practical usefulness of these techniques and provide insight into their limitations. Finally, we will discuss the similarities and differences between these three fields. We will compare their architectures, algorithms, and applications, highlighting their strengths and weaknesses. We will also discuss the current state of research in these fields, including recent advances and future directions.

III. LITERATURE REVIEW

Neural Networks

Neural Networks (NNs) also referred to as Artificial Neural Networks (ANNs), are an AI type, which are modeled after the function and structure of the human brain. The earliest work on NNs can be traced back to the early 1940s, with the pioneering work of Walter Pitts and Warren McCulloch. They proposed a mathematical model of neurons, which laid the foundation for the development of NNs. In the 1960s and 1970s, the perceptron algorithm, developed by Frank Rosenblatt, was introduced as a simple model of a single-layer feedforward neural network [5]. However, it was later discovered that the perceptron algorithm could only solve linearly separable problems and did not have the ability to learn.

In the [6], Chen, Li, and Yang introduced the backpropagation algorithm, which allowed NNs to learn from errors and improve their performance. This renewed focus on NNs spawned new network topologies including RBFNs, MLPs, and Hopfields. There have been substantial developments in neural network topologies like RNNs and CNNs in recent years (CNNs). It has been shown that convolutional neural networks (CNNs) excel in image recognition tasks, whereas recurrent neural networks (RNNs) excel at NLP.

Neuro-Fuzzy Systems

Neuro-Fuzzy Systems (NFS), also known as Adaptive-Network-Based Fuzzy Inference Systems (ANFIS), are a type of AI that combines the strengths of NNs and FL. FL is a form of mathematical logic that deals with reasoning under uncertainty. In [7], Sundaramurthy, Sugumaran, Thangavelu, and Sekaran proposed the concept of NFS, which integrate NNs and FL to improve the performance of fuzzy systems. NFS have been applied to various tasks such as control, prediction, and classification. They have been found to be particularly useful in tasks involving uncertainty and imprecision, such as decision making and control. The key advantage of NFS is that they can handle both numerical and categorical data, making them more versatile than traditional fuzzy systems.

Pattern Recognition

Pattern Recognition (PR) is a subfield of ML that deals with the identification and classification of patterns in data. It has its roots in the field of statistics and has been applied to different fields, e.g., bioinformatics, natural language processing, and computer vision. The main task in PR is to extract features from the data, which are then used to classify the data into different categories. There are different approaches for feature extraction, e.g., principal component analysis (PCA) [8], and linear discriminant analysis (LDA) [9]. After the features have been retrieved, the data may be categorised using techniques like support vector machines (SVMs) and k-nearest neighbors (k-NN). Some of the most recent developments in PR include the use of large data, which has led to the creation of scalable algorithms like random forest and gradient boosting, and machine learning, which has been used in NLP and computer perception.

In conclusion, NNs, NFS, and PR are three closely related fields of study in the field of AI and ML. Each of these fields has its own history, architectures, algorithms, and applications. They have been widely used in various fields and have been found to be successful in solving problems. However, they all have their own strengths and weaknesses and the choice of which method to use depends on.

IV. ARCHITECTURES AND APPLICATIONS

Architecture

Neural Networks

As mentioned earlier, NNs signify a ML algorithm, which is modeled after the functionality and structure of the human brain. These algorithms have been constructed to identify patterns, and potentially make predictions according to input data. The infrastructure of NNs refers to the layout and organization of its layers, nodes, and connections. A neural network normally integrates input layers, output layers, and one or more hidden layers. The input layer obtains raw input data, e.g., text or images, and transfers it on the first hidden layer, whereby each one processes information and transfers it on the next layer, until it scopes the output layer. This layer then generates the final classification or prediction based on processed dataset.

The nodes in a neural network, also known as neurons, are the basic building blocks of the network [10]]. Each neuron receives input from other neurons in the previous layer, performs a calculation, and passes the result to the next layer. The calculation performed by a neuron is typically a simple mathematical function, such as a dot product or sigmoid function. The connections between neurons are represented by weight values, which represent the strength of the connection. During training, the weights are adjusted to optimize the performance of the system. The process of transforming weights is typically known as backpropagation, which involves calculating the errors between the forecasted outputs and actual outputs, and then updating the weights to reduce the error.

There are several types of neural network architectures, each with their own strengths and weaknesses. Some popular architecture includes feedforward networks, Convolutional Neural Networks, and recurrent NNs.

Feedforward Networks

Feedforward networks, also known as multi-layer perceptrons, are the simplest form of NNS and are used for a variety of tasks such as image classification and text generation [11]. They consist of an output layer, one or more hidden layer, and an input layer, with no interconnections between layers.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of neural network that is predominantly well suited for image and video processing tasks. According to Jiang et al. [12], they are based on the idea of convolution, which is a mathematical operation that extracts features from images. In order to analyze and predict picture input, CNNs employ a series of layers, including convolutional layers, fully linked layers, and pooling layers.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are NNs, which are designed to generate and process sequential information, e.g., natural language or time series. They use a type of node called a recurrent node, which allows the network to maintain a state and process data in a sequential manner. RNNs are used for tasks such as speech recognition and language translation. The design of a neural network refers to the layout and organization of its layers, nodes, and connections [13]. Different types of neural network architectures are well suited for different tasks and applications. The most important aspect of neural network architecture is its capacity to learn from the information and make potential predictions with high accuracy. Another important aspect of neural network architecture is the activation function, which is a mathematical function employed to the output of every neuron. The activation function allows the network to introduce non-linearity into the model, which is essential for solving complex problems. Some common activation functions include sigmoid, ReLU, and tanh as illustrated in **Table 1**.

Table 1. Activation functions for the neural network

Sigmoid function	Sigmoid function represents a smooth functionality, which maps input values to a value between 0 and 1. It is typically employed in output layers of NNs for binary classification issues. However, sigmoid function suffers from the vanishing gradient problem, which makes it difficult to train deep networks.
Rectified Linear Unit function	Popular activation feature ReLU (Rectified Linear Unit) reverses the sign of input values less than 0 and translates input values equal or greater to 0 to itself. ReLU is computationally efficient and helps to alleviate the vanishing gradient problem.

Hyperbolic tangent function	Similar to sigmoid function is the Tanh (hyperbolic tangent) function; however, it maps input values to a varying a range between -1 and 1. It is commonly employed in the hidden layer of NNs.

The quantity of layers or synapses in a neural network is also crucial to its functioning. The number of layers and the neurons is known as the network's capacity, which determines the complexity of the model. The more layers and neurons a network contains, the more complex patterns it can learn from the input, but also increases the risk of overfitting. A network with a low capacity has fewer layers and neurons and is less likely to overfit, but may not be able to learn the underlying patterns in the data as well. In addition, Dropout is a technique used to prevent overfitting in NNs. It works by randomly dropping out (disabling) a certain percentage of neurons during each training iteration. This forces the network to learn multiple independent representations of the data, making the model more robust to unseen data.

Transfer Learning is a method used to improve deep neural network efficiency, which is an approach where a framework trained on a single task is employed as a beginning point for a framework on a second task. This can greatly improve performance and reduce the amount of data and computation required to train the model. In summary, the architecture of NNs is a crucial aspect of their design and performance. It includes the activation functions, the number of layers and neurons, and techniques such as dropout and transfer learning. The objective of neural network architecture is to identity a good balance between model complexity and performance, which requires a combination of knowledge, experimentation and fine-tuning.

Neuro-Fuzzy Systems

Neuro-Fuzzy Systems (NFS), also known as adaptive network-based fuzzy systems (ANFIS), are a type of hybrid system that combines the strengths of both NNs and FL. They are designed to solve complex problems that involve both continuous and discrete data, and have been employed to a large number of areas, including control systems, PR, and forecasting. The architecture of a NFS typically integrates five layers: the output layer, the normalization layer, the rule layer, the fuzzy layer, and the input layer.

Input Layer

The input layer in a NFS is responsible for receiving and processing input data. The input data is typically a set of numerical or symbolic values, which are then transformed and processed by the input layer to produce a set of features or attributes that are used by the system's fuzzy inference engine. The input layer typically includes a number of processing elements, such as fuzzy membership functions, which are utilize to map input dataset to the appropriate set of features or attributes. Additionally, the input layer may include a number of pre-processing or normalization techniques to ensure that the input data is in the appropriate format for the fuzzy inference engine.

The input layer in a NFS is a crucial component as it's crucial to the model's functionality as a whole. It is the responsibility of this layer to glean the pertinent details from the raw data and transforming it into a format that can be easily understood by the fuzzy inference engine. This allows the system to make accurate and reliable decisions based on the input data. The input layer in a NFS can be designed to handle a wide range of input data types and formats. For example, it can process numerical data, symbolic data, or even image or audio data. This makes NFS highly versatile and applicable to different issues and applications.

In addition to processing input data, the input layer in a NFS may also include a number of adaptive learning algorithms that allow the system to adjust its parameters and behavior based on the input data. This allows the system to improve its performance over time, and adapt to changing conditions. Overall, the input layer in a NFS is a crucial component that plays a key obligation in the versatility and performance of the model. It is responsible for extracting the relevant information from the input data and transforming it into a format that can be easily understood by the fuzzy inference engine, allowing for accurate and reliable decisions to be made.

The fuzzy layer

The fuzzy layer in a NFS is responsible for applying FL to the input data. It is the core of the system where the fuzzy inference takes place. The fuzzy layer is composed of a fuzzy inference engine and set of fuzzy rules. The fuzzy rules are a set of if-then statements that describe the interconnection between the output data and the input. They are designed to capture the human expert's knowledge and decision-making process. The fuzzy inference engine takes the input data and the fuzzy rules and applies them to produce the output. The fuzzy layer uses a number of fuzzy membership functions to map the input dataset to the appropriate fuzzy sets. These membership functions are used to determine the membership degree of a particular input value to a specific fuzzy set. This allows the system to make decisions based on the degree of similarity between the input data and the fuzzy sets.

The fuzzy layer also includes a defuzzification process that transforms the fuzzy output into crisp values. This process allows the system to produce a single output value that represents the final decision made by the system. Overall, the fuzzy layer in a NFS is responsible for applying FL to the input data and making decisions based on the fuzzy rules and membership functions. It is the core of the system where the fuzzy inference takes place and it is able to model the human decision-making process.

In addition to its core functions, the fuzzy layer in a NFS can also include a number of additional features that enhance its performance and capabilities. For example, it may include adaptive learning algorithms that allow the system to adjust its fuzzy rules and membership functions based on the input data. This allows the system to improve its performance over time and adapt to changing conditions. The fuzzy layer may also include a number of techniques for handling uncertainty and imprecision in the input data. For example, it may use FL operators such as "and" or "or" to handle multiple conflicting inputs, or use possibility theory to handle incomplete or uncertain input data.

Another important aspect of the fuzzy layer is its interpretability, as the fuzzy rules are designed to be interpretable by human experts; it makes it easy to understand the reasoning behind the system's decisions. This interpretability helps to build trust in the system and permits for the easier detection of areas or errors for improvement. In summary, the fuzzy layer in a NFS is a crucial component that applies FL to the input data and makes decisions according to different fuzzy rules and membership functions. It includes a number of additional features such as adaptive learning, handling uncertainty and imprecision, and interpretability. All of these features make NFS robust, versatile, and able to model the human decision-making process.

The Rule Layer

The rule layer is where the fuzzy rules are applied to the input data. For each rule, there is an antecedent (if-part) and a consequent (then-part) that is employed to make classification or predictions based on input data [14]. The antecedent is a combination of the fuzzy sets that the input data belongs to, while the consequent is a set of parameters that are used to make the final prediction. The rule layer in a NFS is responsible for representing the knowledge and decision-making process of the model in the form of the fuzzy rule, which are a set of if-then statements that describe the interconnection between relationships output data and the input. They are designed to capture the human expert's knowledge and process of decision-making and are employed by the fuzzy inference engine in the fuzzy layer to make decisions.

According to Bai, Zhu, Wen, Zhang, and Zhang [15], the rule layer includes a collection of fuzzy rules that are defined based on the input data and the desired output. These rules are typically represented in a human-readable format, making them easy to understand and interpret by human experts. The number of rules as well as their complexity may vary depending on the application and the amount of knowledge that needs to be represented. The rule layer in a NFS can also include a number of techniques for generating and optimizing the fuzzy rules. For example, it may use a genetic algorithm or a neural network to generate the rules, or use optimization techniques to adjust the parameters of the existing rules to improve their performance. This allows the system to adapt and improve its performance over time.

The rule layer also includes a process of simplification of the rule base. The simplification process aims at reducing the number of rules and making them more comprehensible. This can be achieved by applying techniques such as rule merging, pruning, or clustering. The rule layer in a NFS is responsible for representing the knowledge and decision-making process of the system in the form of rules, which are easier to understand and interpret by human experts and they are used by the fuzzy inference engine to make decisions. The rule layer can include a number of techniques for generating, optimizing and simplifying the fuzzy rules, allowing the system to adapt and improve its performance over time.

The Normalization Layer

The normalization layer in a NFS is responsible for pre-processing and normalizing the input data before it is passed to the input layer for further processing [16]. The goal of the normalization layer is to ensure that the input data is in the appropriate format for the fuzzy inference engine and to enhance the general functionality of the model. The normalization layer includes a number of techniques for pre-processing and normalizing the input data. These techniques can include data scaling, data transformation, data cleaning, and data imputation. Data scaling, for example, is used to adjust the range of the input data to a specific range. Data transformation, on the other hand, is used to change the format of the input data, such as converting it from numerical to symbolic format.

Data cleaning is used to remove any irrelevant or missing data, while data imputation is used to fill in missing data. These techniques are important to ensure that the input data is in the appropriate format and to improve the overall operation of the system. It is also worth mentioning that the normalization layer can include techniques for dealing with outliers, which are values that lie far from the expected range of values. These techniques can include using statistical methods such as the median or the interquartile range to identify and remove outliers.

It includes a number of techniques such as data scaling, data transformation, data cleaning, and data imputation to ensure that the input data is in the appropriate format and to improve the overall performance of the system. It also can include techniques for dealing with outliers to ensure that the input data is free of errors and in a format that the system can process. The normalization layer is a critical component of a NFS as it helps to ensure that the input data is in the appropriate format for the fuzzy inference engine. Without proper normalization, the system may not be able to make accurate and reliable decisions based on the input data.

Another important function of the normalization layer is to aid in the enhancement of the generalization operation of the system. By normalizing the input data, it can help the system to be more robust and less sensitive to variations in the input data. This can help the system to perform well in different scenarios, and make accurate predictions even when dealing with new or unseen data. Moreover, the normalization layer can also be used to improve the interpretability of the system by providing a common scale for the input data. This can make it easier for human operations to comprehend the cognition behind the system's decisions, which can help to build trust in the system. It is also worth noting that depending on the type and format of the input data, the normalization layer may need to use different techniques to pre-process and normalize the data. For example, normalizing image data may require different techniques than normalizing text data.

In summary, the normalization layer in a NFS is a critical component that is responsible for pre-processing and normalizing the input data before it is passed to the input layer for further processing. It includes a number of techniques such as data scaling, data transformation, data cleaning, and data imputation to ensure that the input data is in the appropriate format and to enhance the general operation of the model by making it more robust and less sensitive to variations in the input data, and can improve the interpretability of the system by providing a common scale for the input data. The choice of techniques used in the normalization layer may vary depending on the type and format of the input dataset.

The Output Layer

The output layer in a NFS is responsible for producing the final decision or output based on the input data and the fuzzy rules. It takes the output from the fuzzy layer, which is usually in the form of fuzzy sets, and converts it into a crisp value using a defuzzification process. The output layer is the last phase in the process of decision-making of a NFS. The output layer typically includes a number of defuzzification methods such as the Centroid approach, the Weighted Average (WA) method, and the Mean of Maximum (MOM) method. These methods convert the fuzzy output into the crisp values by taking into account the membership degree of the input data to each fuzzy set. The output layer can also include a number of post-processing techniques to further refine the output. For example, it may include techniques for handling uncertainty, such as returning a range of possible outputs or a probability distribution. Additionally, the output layer may include a number of visualization techniques to help human experts understand the output and the reasoning behind the system's decisions.

One of the main advantages of NFS is their ability to handle uncertainty and imprecision in the input data. FL allows the system to work with vague or incomplete information, while NNs provide the capacity to learn from information and data, and potentially make predictions based on patterns in the data. Another advantage of NFS is their ability to learn from both supervised and unsupervised data. They can be trained using a variety of techniques such as backpropagation, gradient descent and least squares method. In addition, NFS can be used purposely to generalize function modeling, from simple linear functions to highly non-linear functions. This makes them suitable for many different uses, e.g., control systems, PR, and forecasting. Nonetheless, there are also some limitations to NFS. One limitation is the difficulty of interpreting the fuzzy rules, which can make it difficult to understand how the system is making its predictions. Additionally, NFS can be sensitive to the choice of fuzzy sets and rule complexity, which may have an effect on how well the system works.

Neuro-Fuzzy Systems are a powerful type of hybrid system that combines the strengths of both NNs and FL. They are designed to solve complex problems that involve both continuous and discrete data, and have been applied to different. NFS can handle uncertainty and imprecision in the input data, and can be used to model a wide range of functions. However, their interpretability and sensitivity to the choice of fuzzy sets and the number of rules are some of the limitations of NFS.

Pattern Recognition

Pattern Recognition (PR) is a field of study within computer science and AI, which concentrate on the advancement of different models and algorithms that can automatically identify patterns within data. The architecture of a PR system is critical to its ability to accurately and efficiently identify patterns within data. There are several different types of PR architecture, each with its own strengths and weaknesses. One of the most common types are artificial neural network (ANN) architecture, which are modeled after the brain structure and are formulated to learn from data through a process of training and testing. They are particularly well-suited for complex, non-linear patterns and can be employed for different applications, e.g., as recognition of visual content, recognizing spoken language, and processing of natural language.

Another type of PR architecture is the decision tree architecture. Decision trees are a type of algorithm that can be used to classify data based on a series of decisions made at each node of the tree. They are particularly well-suited for tasks that involve multiple decision points, such as medical diagnosis or fraud detection. Another type of PR architecture is the Bayesian network architecture. Bayesian networks are a type of probabilistic graphical model, which could be employed in modeling complex systems and make predictions about future events. They are particularly well-suited for tasks that involve uncertain or incomplete information, such as weather forecasting or speech recognition. There is also a more recent architecture called Hybrid architecture, which combines the strengths of multiple architectures to improve performance. A common example of hybrid architecture is combining a decision tree with a neural network.

The architecture of a PR system is a crucial factor in determining its ability to accurately and efficiently identify patterns within data. Different types of architecture have different strengths and weaknesses and are well-suited for different types of tasks. Therefore, choosing the appropriate architecture represents a major advancement of a successful PR system. **Table 2** presents an overview of the different types of architectures of PR.

Table 2. Types of architectures of PR	
Artificial Neural Network (ANN) Architecture	ANN integrates layers of interlinked nodes, identified as artificial neurons, which are trained to recognize patterns within data. ANNs are particularly well-suited for complex, non-linear patterns, and can be employed for different applications such as processing in natural languages, recognizing spoken language, and identifying visual content. One of the major advantages of ANNs is their capacity to comprehend and learn from data through the testing and training process. They are also able to generalize their knowledge to new situations, making them well-suited for tasks that involve a large amount of data or changing conditions.
Decision Tree Architecture	Decision trees are a type of algorithm that can be used to classify data based on a series of decisions made at each node of the tree. They are particularly well-suited for tasks that involve multiple decision points, such as medical diagnosis or fraud detection. Decision trees are simple to understand and interpret, making them easy to use for non-experts. Additionally, decision trees are easy to construct and can be used with both categorical and numerical data. However, decision trees can be prone to overfitting, which occurs when the tree becomes too complex and starts to fit the noise in the data rather than the underlying pattern.
Bayesian Network Architecture	Bayesian networks are particularly well-suited for tasks that involve uncertain or incomplete information, such as weather forecasting or speech recognition. Bayesian networks are based on the principles of Bayesian statistics, which provide a way to update beliefs about the state of the world based on new evidence. One of the main advantages of Bayesian networks is their ability to handle uncertainty and missing data. However, they can be computationally intensive and may require a significant amount of data to produce accurate results.
Hybrid Architecture	Hybrid architectures combine the strengths of multiple architectures to improve performance. They are often used when the task at hand is too complex for a single architecture to handle. A common example of hybrid architecture is combining a decision tree with a neural network. The decision tree is used to choose a feature subset to be passed to the neural network, which then performs the classification. This can lead to improved performance, as the decision tree is able to select the most informative features, while the neural network is capable of modeling the complex relationships between the features.

In conclusion, there are several different types of PR architecture, each with its own strengths and weaknesses. The choice of architecture will rely on a particular task, data type, and the resources available. Hybrid architectures can be a good option when the task is too complex for a single architecture to handle. It's important to evaluate the specific needs of the task and the available data before choosing architecture.

Applications

Neural Networks

Neural Networks (NNs) are currently revolutionizing many aspects of daily life and business by bringing new levels of AI to various industries. They are being developed for a number of reasons, including increasing the efficiency of ML and enhancing technology's ability to solve every-day human challenges and tasks. NNs are designed to mimic the way the human brain works. Once trained using a set of inputs, they can produce the desired results. **Table 3** shows some of the example of real-life NNs applications.

Table 3: Real-world applications of NNs	
Speech and Voice Recognition:	Speech recognition technology is now widely used in various applications. It can be found in home automation systems, hands-free computing, video games, and virtual assistants like Siri and Alexa. These virtual assistants are made possible by sophisticated NNs. Additionally, platforms such as Google and YouTube have also integrated a voice search feature to assist users.
Fraud Prevention:	In the field of finance, NNs are utilized to detect fraudulent transactions. Some NNs are designed to analyze past account transactions and their frequency to identify potential fraud. Additionally, NNs can also detect fraudulent transactions by analyzing the transaction size and the type of retailer involved.
eCommerce:	In the eCommerce industry, NNs are primarily used to personalize the customer experience. Large eCommerce platforms employ AI to display related and recommended products that a customer might be interested in buying. These recommendations are based on consumer behavior and past purchases.
Cybersecurity:	NNs are also commonly used to safeguard computers from viruses and malware. They can protect a computer from cyber-attacks by identifying whether a USB device is faulty or contains harmful software. They also help to exploit zero-day vulnerabilities.
Text Classification	Text classification is primarily utilized in web search, information filtering, and language

And Categorization:	identification. This method can also be used to organize website content and documents on a system, such as grouping them by topic or categorizing them based on priority or urgency.
Stock Market Prediction:	There are various factors that can impact the performance of the stock market. NNs can analyze these factors and predict prices to assist traders in making investment decisions. However, the development of these applications is still in the early stages, as the networks need to study a large volume of historical and market data in order to make accurate predictions.
Marketing:	NNs are also commonly employed in target marketing. Marketers use market segmentation methods to divide potential customers into different groups based on characteristics such as consumer behavior, age, location, and demographics. NNs can then be programmed to interact with each segment in a personalized and appropriate manner.

NNs can be incredibly valuable for a wide range of industries, businesses, and individuals. In particular, they can be useful for marketers looking to personalize their strategies and target repeat customers. They can also aid stock traders in making investment decisions. Additionally, banks have been utilizing NNs for the detection of fraud and to improve cybersecurity.

Neuro-Fuzzy Systems

As fuzzy systems gained popularity in industrial applications, it became clear that creating a fuzzy model with better performance was not a simplified task. Finding a better membership function and rule typically integrate a challenging process of trials and errors. In order to address this, the ideology of integrating learning approaches into fuzzy models was recommended. NNs, which have effective learning algorithms, were considered as a solution to digitalize or assist in the tuning and development of fuzzy models. The initial applications of these methods were primarily in process control, but they have since been applied to a wide range of fields. Gradually, its application has expanded to encompass various areas of knowledge, as illustrated in **Table 4** below.

Table 4. Areas of knowledge in neuro-fuzzy systems	
Control systems	NFS are commonly used in control systems to improve the performance of machines and processes. They can be used to control the temperature of a furnace, the speed of a motor, or the flow rate of a liquid.
Robotics	NFS are also used in robotics to improve the decision-making capabilities of robots. They can be used to control the movement of a robot or to help it navigate through an unknown environment.
Predictive maintenance	NFS can be used to predict when a machine or system is likely to fail, allowing for preventative maintenance to be performed before a failure occurs. This can save companies a significant amount of money by reducing downtime and avoiding costly repairs.
Medical diagnosis	NFS can be used to assist doctors in making diagnoses. They can be trained on a dataset of patient data to identify patterns and make predictions about the likelihood of a particular diagnosis.
Image and signal processing	NFS can be used to improve the accuracy of image and signal processing. They can be used to identify patterns and features in images, such as objects or faces, or to filter noise from signals.
Natural Language Processing	NFS can be used in natural language processing to improve the accuracy of language translation, sentiment analysis, and text generation.

Neuro-Fuzzy Systems (NFS) have a wide range of applications and they have proven to be a powerful tool in various fields such as control systems, robotics, predictive maintenance, medical diagnosis, image and signal processing, and natural language processing [17]. Their ability to learn and make decisions based on uncertain or imprecise information makes them a valuable asset in many different industries.

Pattern Recognition

Pattern Recognition (PR) is a technique that uses algorithms to identity regularities and patterns of data. It is a fundamental aspect of ML and is used in a wide variety of applications. Some of the most common applications of PR include (**Table 5**).

Table 5. Pattern recognition application	
Biometric recognition	PR is used to identify individuals based on their unique physical or behavioral characteristics, such as fingerprints, facial features, or iris patterns. This technology is used in applications such as security systems, access control, and personal identification.
Natural language processing	PR is used to identify patterns in natural language data, such as text and speech. It is used in applications such as text classification, sentiment analysis, and machine translation.
Financial analysis	PR algorithms can be used to analyze financial data, such as stock prices and trading volumes, to identify patterns and make predictions about future market movements.
Medical diagnosis	PR can be used to help doctors make diagnoses by evaluating different clinical images, e.g., MRI's, CT scans, and X-rays. It is also employed in analyzing patient data to identity different patterns, and forecasts about the patients' health conditions.
Speech recognition	PR algorithms are used to convert speech into text, enabling computers to understand spoken language. This technology is used in applications such as voice assistants, dictation software, and telephone call routing systems.

Image and video analysis	PR is used to analyze images and videos to identify objects, faces, and other features. It can be used in applications such as facial recognition, image classification and object detection.

Pattern Recognition (PR) is a powerful technique that is used in different applications such as video or image analysis, biometric recognition, speech recognition, financial analysis, natural language processing, and medical diagnosis. The ability of PR algorithms to identify patterns and regularities in data makes them a valuable tool in many different industries and research fields.

V. SIMILARITIES AND DIFFERENCES

Similarities

Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR) are all related fields that aim to model and analyze complex data. NNs are ML models, which are also based on the functionality and structure of the human brain. These models integrate layers of interlinked artificial neurons that have the capacity to transmit and process data via the network. In addition, NNs can be employed to perform different tasks, include natural language process, image recognition, and prediction. NFS are a combination of NNs and FL. FL is a mathematical framework that deals with imprecise or uncertain information, and it is often used to model human reasoning and decision-making. NFS combine the ability of NNs to learn from data with the ability of FL to handle imprecise or uncertain information.

Pattern Recognition (PR) is a field of study that deals with the automatic recognition of patterns in data. It is a subfield of ML, and it is closely related to both NNs and NFS. NNs and NFS can be used as PR algorithms, and PR techniques can be used to improve the performance of NNs and NFS.

All three fields are related in the sense that they are based on mathematical models that can learn from data and can be used for recognizing patterns or making decisions in complex problems. NNs and NFS are specific types of models that are used for this purpose, while PR is a broader field that encompasses these models and other techniques.

Differences

Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR) are all related fields that aim to model and analyze complex data, however, they have some key differences. Neuro-Fuzzy Systems (NFS) are a combination of NNs and FL. FL is a mathematical framework that deals with imprecise or uncertain information, and it is often used to model human reasoning and decision-making. NFS combine the ability of NNs to learn from data with the ability of FL to handle imprecise or uncertain information. This means NFS are more robust to handle uncertain or ambiguous data compare to neural network.

Pattern Recognition (PR) is a field of study that deals with the automatic recognition of patterns in data. It is a subfield of ML, and it is closely related to both NNs and NFS. However, PR is more general and broader field that encompasses a wide range of techniques, including statistical, geometrical, and structural methods, whereas NNs and NFS are specific types of models used for PR and decision making.

VI. CONCLUSION AND FUTURE RESEARCH

This paper provided a comparative research of Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR), highlighting their similarities and differences. The use of literature review and real-life applications will provide a comprehensive understanding of the various aspects of these fields. The results of this study will be useful for researchers and practitioners in the field of AI and ML, providing insight into the strengths and limitations of these techniques. These three fields are closely related and share a common goal of improving the accuracy and efficiency of PR. Each of these fields has different strengths and weaknesses, and has been applied in various real-world problems. This study can be used as a reference for researchers and practitioners working in these fields, to comprehend the limitations and strengths of these approaches.

There are several potential future research directions for Neural Networks (NNs), Neuro-Fuzzy Systems (NFS) and Pattern Recognition (PR).

- Neural Networks: One of the main areas of research is making NNs more efficient and able to learn from less data. This includes developing techniques for transfer learning, which allows a neural network trained on one task to be fine-tuned for another, and one-shot learning, which allows a neural network to learn from a single example. Another area of research is making NNs more interpretable, so that it is easier to understand why they are making certain predictions.
- Neuro-Fuzzy Systems: Research in NFS is focused on developing methods for combining the advantages of NNs and fuzzy systems. This includes developing methods for learning the parameters of fuzzy systems using NNs, and methods for incorporating FL into NNs to make them more interpretable.
- Pattern Recognition: Research in PR is focused on developing methods for automatically recognizing patterns in data, such as images, speech, and text. This includes developing deep learning methods for image and speech recognition, and natural language processing techniques for analyzing text data. Another area of research is developing methods for robust PR, which can handle noisy or incomplete data.

Overall, the future research in these fields will be driven by the need for more accurate and efficient methods for processing and analyzing large amounts of data, as well as the need for methods that can adapt to changing environments and can explain their decisions. With advancements in technology, there will be more focus on developing more robust and interpretable models, as well as methods for incorporating domain knowledge and prior information into the training process. Additionally, there will be a growing emphasis on developing AI systems that can provide interpretable and transparent explanations of their decisions.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding was received to assist with the preparation of this manuscript.

Ethics Approval and Consent to Participate
Not applicable.

Competing Interests
There are no competing interests.

References

[1]. Z. Wu, X. Nie, and B. Cao, "Coexistence and local stability of multiple equilibrium points for fractional-order state-dependent switched competitive neural networks with time-varying delays," Neural Netw., vol. 160, pp. 132–147, 2023.

[2]. S. S. Soomro, A. H. Jalbani, M. I. Channa, S. Lakho, and I. A. Memon, "An evaluation of smart learning approach using bloom taxonomy based neuro-fuzzy system," J. Intell. Fuzzy Syst., vol. 43, no. 2, pp. 1995–2004, 2022.

[3]. I. Aattouri, H. Mouncif, and M. Rida, "Modeling of an artificial intelligence based enterprise callbot with natural language processing and machine learning algorithms," IAES Int. J. Artif. Intell. (IJ-AI), vol. 12, no. 2, p. 943, 2023.

[4]. C. Wang, "A surrogate approach to estimate the intrinsic multifractality in financial returns using adaptive network-based fuzzy inference system (ANFIS)," J. Intell. Fuzzy Syst., vol. 40, no. 4, pp. 8541–8547, 2021.

[5]. "Neural Networks - History," Stanford.edu. [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html. [Accessed: 15-Jan-2023].

[6]. Y. Chen, C. Li, and J. Yang, "Design and application of Nagar-Bardini structure-based interval type-2 fuzzy logic systems optimized with the combination of backpropagation algorithms and recursive least square algorithms," Expert Syst. Appl., vol. 211, no. 118596, p. 118596, 2023.

[7]. S. Sundaramurthy, V. Sugumaran, A. Thangavelu, and K. Sekaran, "Predicting rheumatoid arthritis from the biomarkers of clinical trials using improved harmony search optimization with adaptive neuro-fuzzy inference system," J. Intell. Fuzzy Syst., vol. 44, no. 1, pp. 125–137, 2023.

[8]. X. Ma, J. Zhao, Y. Wang, C. Shang, and F. Jiang, "Robust factored principal component analysis for matrix-valued outlier accommodation and detection," Comput. Stat. Data Anal., vol. 179, no. 107657, p. 107657, 2023.

[9]. L. A. Prendergast and J. A. Smith, "Influence functions for linear discriminant analysis: Sensitivity analysis and efficient influence diagnostics," J. Multivar. Anal., vol. 190, no. 104993, p. 104993, 2022.

[10]. R. Holota, "Colour image recognition based on single-layer neural networks of min/max nodes," Neural Netw. World, vol. 22, no. 4, pp. 395–405, 2012.

[11]. K. Satish Kumar, P. AnandRaj, K. Sreelatha, and V. Sridhar, "Reconstruction of GRACE terrestrial water storage anomalies using Multi-Layer Perceptrons for South Indian River basins," Sci. Total Environ., vol. 857, no. Pt 2, p. 159289, 2023.

[12]. X. Jiang et al., "Characterizing functional brain networks via Spatio-Temporal Attention 4D Convolutional Neural Networks (STA-4DCNNs)," Neural Netw., vol. 158, pp. 99–110, 2023.

[13]. P. S. Geidarov, "Clearly defined architectures of neural networks and multilayer perceptron," Opt. Mem. Neural Netw., vol. 26, no. 1, pp. 62–76, 2017.

[14]. M.-G. Joo, "Automatic learning of fuzzy rules for the equivalent 2 layered hierarchical fuzzy system," J. Korean Inst. Intell. Syst., vol. 17, no. 5, pp. 598–603, 2007.

[15]. K. Bai, X. Zhu, S. Wen, R. Zhang, and W. Zhang, "Broad learning based dynamic fuzzy inference system with adaptive structure and interpretable fuzzy rules," IEEE Trans. Fuzzy Syst., vol. 30, no. 8, pp. 3270–3283, 2022.

[16]. A. Ziaee and E. Çano, "Batch Layer Normalization, A new normalization layer for CNNs and RNN," arXiv [cs.LG], 2022.

[17]. G. S. Ng, F. Liu, T. F. Loh, and C. Quek, "A novel brain-inspired neuro-fuzzy hybrid system for artificial ventilation modeling," Expert Syst. Appl., vol. 39, no. 15, pp. 11808–11817, 2012.