People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

IBN KHALDOUN UNIVERSITY OF TIARET

Dissertation

Presented to:

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DÉPARTEMENT OF COMPUTER SCIENCE

Degree of MASTER

Specialty: software engineer

Presented by:

DJENANE Mouloud Amine
ZAHI Narimane

On the theme:

DEEP LEARNING FOR NONLINEAR REGRESSION

Defended publicly on 20 / 6 /2022 in Tiaret in front the jury composed of:

Mr BENDAOUD Mebarek	Prof	Tiaret University	Chairman
Mr AID Lahcene	MCA	Tiaret University	Supervisor
Mr MOSTEFAOUI Sid Ahmed	MCA	Tiaret University	Examiner

2021-2022

DEDICATION

This Dissertation is dedicated

to my parents

for their endless love, support and encouragement.

to my sisters

For being amazing sisters and best friends all at once

To the best partner I ever could get, Narimane

Thank you for your efforts and the hard work you put on this project, I'm grateful to have you as a partner.

To the closest friends to my heart

Hassane , Elhoucine, Mohammed, Sofiane , Raouf, Aziz, Riadh, Silia, Houda, Sara, Imen, Vanessa….

To everyone who has supported me and helped me get to where I am now,

thank you all.

(Mouloud)

DEDICATION

I must fully admit my gratitude to all the people who supported me during my path, who knew how to lift me up for reaching my goals, it is with a great love, respect and gratitude that I dedicate this modest work:

To my dearest mother K.C.Anissa ….There are times when we would like to bring someone back from heaven just to listen to his laughter and have the opportunity to tell him that we love him and that we really miss his presence... peace to your soul mom .

For my lovely father Zahi Ahcene, you are for me my source of joy and comfort, your prayers have never ceased and if I am at this stage of life, it is thanks to your encouragement and your words of support, words alone could not express all my love nor my esteem for you.

To my beloved brother and sister Lina, brotherhood is a small word to describe what unites us, you are this wonderful gift from the Lord and I wish you many blessings.

To my second mother djamila and all my familly Dalila, Samira, Nadia, zahra, Kahina, Mustapha, Mounir, Sofiane, kamel and Hamid no dedication could express all the love I have for you all.

To my best cousins Racha, katia, Emilia and the others. I can't put on paper the affection I have for you, I implore Allah to give you a better future.

To my partner Mouloud with whom I lived beautiful moments during this year and to his family.

For my best team of friends. Life is partly what we make it, and partly what it is made by the friends we choose and you make life more beautiful.

Narimane

Acknowledgements

First of all, we thank the Almighty God, for giving us the strength and patience to complete this modest work, our great greeting on the first educator our prophet Mohamed (Satisfaction and salvation of God on him)

We first express our deepest thanks to our supervisor Mr.***AID Lahcene*** we thank you for the honor you have given us by directing this project, without your corrections and your meticulous advice, this work could not have been completed. Your kindness, your competence, and especially your professional qualities inspire us, may this work be the testimony of the expression of our sincere thanks and our respectful gratitude with a great admiration and deep respect.

Our thanks also go to the members of the jury Mr. ***BENDAOUD Mebarek*** and Mr. ***MOSTEFAOUI Sid Ahmed***. It is a great honor and a privilege to have you as jury members to judge our modest work.

A special thanks goes to Mr. ***DEHBI Abdelkader*** for his valuable help, he gave us a strong hand in order to complete our project

We would also like to express our gratitude to all the people who have helped us from near or far, especially the staff of our faculty, for their advice and their help in the realization of our work.

Abstract

Artificial intelligence based on deep learning has shown to be useful in a wide range of applications and research areas because, in contrast to traditional machine learning algorithms, it can produce extremely high-level data representations from amounts of raw data. As a result, it has proven to be an excellent solution to a broad range of linear issues, and it seems to be highly for the non-linear ones

The aim of this work is to propose a methodology to be followed in the design of deep learning model for non-linear regression, then developing a CNN model that can deal with this scenario. Thereafter, apply it to solve real world issues such as predicting lifetime and mechanical performance degradation of multilayer greenhouse polyethylene films.

we created a hybrid deep learning model capable of handling the challenge in order to complete this assignment. Both the training and validation phases of the DL model were successful, demonstrating the feasibility of using 1D CNN for nonlinear regression.

Key words: Artificial intelligence, Machine learning, Deep learning, linear regression, nonlinear regression, CNN.

Résumé

L'intelligence artificielle basé sur l'apprentissage profond s'est révélée utile dans un large éventail d'applications et de domaines de recherche car contrairement aux algorithmes traditionnels d'apprentissage automatique, il peut produire des représentations de données de très haut niveau à partir de quantités de données brutes. Par conséquent, il s'est avérée être une excellente solution à un vaste éventail de problèmes linéaires et il semble qu'il est très efficace pour les non linéaires aussi.

L'objectif de ce travail est de proposer une méthodologie à suivre dans la conception d'un modèle d'apprentissage profond pour la régression non linéaire puis de développer un modèle CNN qui peut traiter ce scénario. Ensuite, nous l'appliquons pour résoudre des problèmes du monde réel tels que la prédiction de la durée de vie et de la dégradation des performances mécaniques des films polyéthylène multicouches en serre.

Pour mener à bien cette mission, nous avons créé un modèle hybride d'apprentissage profond capable de relever le défi. Les phases de formation et de validation du modèle DL ont été couronnées de succès, démontrant la faisabilité de l'utilisation de CNN 1D pour la régression non linéaire.

Mots clés : Intelligence artificielle, apprentissage automatique, apprentissage profond, régression linéaire, régression non linéaire, CNN.

ملخص

أثبت الذكاء الاصطناعي القائم على التعلم العميق فائدته في مدى واسع من التطبيقات ومجالات البحث لأنه على عكس خوارزميات التعلم الآلي التقليدية ، يمكنه إنتاج تمثيلات بيانية عالية المستوى للغاية من كميات البيانات الخام.لذلك ، فقد أثبت أنه حل رائع لمجموعة واسعة من المشكلات الخطية ويبدو أنه فعال جدا للمشاكل الغير خطية أيضا.

الهدف من هذا العمل هو اقتراح منهجية يجب اتباعها في تصميم نموذج التعلم العميق للانحدار الغير خطي ومن ثم تطوير نموذج CNN يمكنه معالجة هذا المشكل.ثم نطبقه لحل مشاكل العالم الحقيقي مثل التنبؤ بعمر وتدهور الأداء الميكانيكي لأفلام البولي إيثيلين متعددة الطبقات في البيوت البلاستيكية.

ولتنفيذ هذا العمل، أنشأنا نموذجا هجينا للتعلم العميق قادرا على مواجهة التحدي.وكانت مرحلتا التدريب والتحقق من صحة النموذج ناجحتين، مما يدل على جدوى استخدام CNN 1D للانحدار الغير خطي.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم الآلي، التعلم العميق، الانحدار الخطي، الانحدار الغير خطي CNN.

Table of contents

Table of Figures

List of tables

List Of Abbreviations

#

(1D) one Dimensional

(2D) two Dimensional.

(3D) three Dimensional.

A

(Adam) Adaptive Moment Estimation

(AE) Auto-Encoder

(AI) Artificial Intelligence

(ANN) Artificial Neural Network

(AUC) Area under the curve

B

(BNNs) Bayesian Neural Networks

C

(CAE) Auto Encoder Contractive

(CBR) Case-Based Reasoning

(CPU) Central Processing Unit

(CNN or ConvNet) Convolutional Neural Network

D

(DAE) Autoencoder Denoising

(DBN) Deep Belief Network

(DL) Deep Learning

(DRL) Deep Reinforcement Learning

(DTL) Deep Transfer Learning

E

(ENET) Elastic Net

F

(FNN) Feedforword Neural Network

G

(GAN) Generative Adversarial Network

(GPU)Graphics Processing Unit

(GRU) Recurrent Gated Units

K

(KDD) Knowledge Discovery From Data

(KNN) K-Nearest Neighbors

L

(LDPE) Low Density Polyethylene Film

(LDA) Linear Discriminant Analysis

(LR) Logistic Regression

(LSTM) Long Short-Term Memory

M

(MAE) Mean Absolute Error

(MDP) Markov Decision Process

(ML) Machine Learning

(MLP) Multi-Layer Perceptron

(MSE) Mean Squared Error

N

(NB) Naïve Bayes

(NLP) Natural Language Processing

(NLU) Natural Language Understanding

(NLG) Natural Language Generation

P

(PCA) Principal Component Analysis

(PCR) Principal Components Regression

(PE) Polyethylene

(PLSR) Partial Least Squares Regression

R

(RBM) Boltzmann Machine Restricted

(ReLU) Rectified Linear Unit

(RFE) Recursive feature Elimination:

(RL) Reinforcement Learning

(RMSE) Root-Mean Squared Error

(RNN) Recurrent Neural Network

S

(SAR) Sparse Autoencoder

(SOM) Self-Organizing Map

(SGD) Stochastic Gradient Descent

(SVM) Support Vector Machine

T

(TANH) Hyperbolic Tangent

V

(VAE) Variational Autoencoder

X

(XGBoost) Extreme Gradient Boosting

GENERALE INTRODUCTION

In recent years, artificial intelligence based on machine learning and deep learning has sparked tremendous global interest. It is considered as a leading technology of the current age of the fourth industrial revolution (Industry4.0 or 4IR), a new era that builds and extends the impact of digitization in new and unanticipated ways. Various types of AI such as analytical, functional, interactive, textual and visual can be applied to enhance the intelligence and capabilities of an application. In general, AI incorporates human behavior and intelligence to machines or systems based on potential techniques which are classified into ten categories : data mining , knowledge discovery and advanced analytics, rule-based modeling and decision-making ,fuzzy logic-based approach , knowledge representation , uncertainty reasoning and expert system modeling, case-based reasoning , text mining and natural language processing , visual analytics, computer vision and pattern recognition , hybridization, searching and optimization ,machine learning , neural networks and deep learning. Machine learning as a part of the AI area represents a method to learn from data or experiences. A successful machine learning model depends on both the data and the performance of the learning algorithms which are divided into several types including supervised, unsupervised, semi-supervised, and reinforcement learning. Besides the deep learning (DL), which is part of a broader family of machine learning methods that represents also a learning method from data where the computation is done through multi-layer neural networks and processing, i.e. it is originated from an artificial neural network (ANN). As DL models gain knowledge from data, there is an approach to multivariate data analysis known as regression allows to focus on the effects of predictors on the response. When the response variable is continuous, the simplest approximation is a linear approximation, which assumes an approximate linear relationship between the response and the predictors. Nonlinear regression, on the other hand, is typically based on the underlying mechanisms which generate the data. Thus, nonlinear models are usually closer to the true relationships than linear models, and nonlinear model predictions are more reliable than linear model. However, deriving nonlinear models for many practical problems may be difficult.

for completing our project, we have organized our work as follows

CHAPTER I: Background on deep learning

CHAPTER II: Nonlinear Regression

CHAPTER III: Development of CNN model based on real dataset

CHAPTER I

Background on deep learning

PART I:

Artificial intelligence and Machine Learning

1.1 Introduction

Artificial intelligence is a key component of digitalization solutions that have received significant attention in the digital arena. It is regarded as a core technology of today's fourth industrial revolution, which represents a new era of innovation in technology that begins with data collection, followed by AI to interpret the data. There are several potential AI techniques in this area that can solve problems, including machine learning. In recent years, ML has expanded rapidly in the context of data analysis and computing, allowing applications to function intelligently based on machine learning algorithms that analyze data. Besides, deep learning technology, which is considered a core part of artificial intelligence, machine learning, and data science, DL has become a hot topic in the context of computing due to its learning capabilities from data and is widely applied in various application areas such as healthcare, visual recognition, text analytics, cybersecurity, and many more. However, due to the dynamic nature and variations in real-world problems and data, developing an appropriate DL model is a difficult task. Furthermore, a lack of core understanding turns DL methods into black-box machines that impede standard-level development. Overall, the goal of this chapter is to provide a high-level overview of AI-based modeling with a main contribution that explains the principles of various machine learning techniques and their applicability. then presents a structured and comprehensive view on DL techniques including a taxonomy considering various types of real-world tasks like supervised or unsupervised with real-world application areas where deep learning techniques can be used.

1.2 Artificial intelligence

Artificial intelligence (AI) is a broad branch of computer science and engineering that focuses on simulating a wide range of issues and functions in the field of human intellect. It’s primarily concerned with comprehending and carrying out intelligent tasks such as acquiring new abilities, thinking and adapting to new contexts and challenges. We explore various types of AI that include analytical, functional, interactive, textual, and visual as shown in figure 1-1.

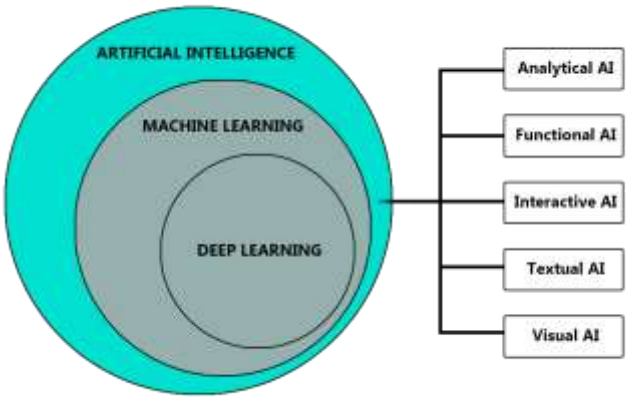


Figure 1-1 : various types of AI

in the following section, we specify the scope of each category in terms of computing and real-world services:

1.2.1 Analytical AI

Analytics typically refers to the process of identifying, interpreting, and communicating meaningful patterns of data. Thus, it aims to discover new insights, patterns, and relationships or dependencies in data and to assist in data-driven decision-making.

Therefore, in the domain of today's business intelligence, has becomes a core part of AI that can provide insights to an enterprise and generate suggestions or recommendations through its analytical processing capability. Various machine learning and deep learning techniques can be used to build an analytical AI model to solve a particular real-world problem. For instance, to assess business risk, a data-driven analytical model can be utilized.

1.2.2 Functional AI

Analytical AI and functional AI work in a similar way. because functional AI also explores massive quantities of data for patterns and dependencies. on the other hand, it executes actions rather than making recommendations. For instance, a functional AI model could be useful in robotics and IoT applications to take immediate actions.

1.2.3 Interactive AI

Interactive AI is commonly used to provide efficient and interactive communication automation, which is well-established in many facets of people's daily lives, particularly in the commercial sphere. For example, an interactive AI model could be valuable in the development of chatbots and smart personal assistants. While building an interactive AI

model, a variety of techniques such as machine learning, frequent pattern mining, reasoning, and AI heuristic search can be employed.

1.2.4 Textual AI

Textual AI typically covers textual analytics or natural language processing, through which businesses can enjoy text recognition, speech-to-text conversion, machine translation, as well as content generation capabilities. For instance, an enterprise may use textual AI to support an internal corporate knowledge repository to provide relevant services, e.g., answering consumers queries.

1.2.5 Visual AI

Visual AI can typically recognize, classify, and sort items, as well as convert images and videos into insights. Thus, it can be considered as a branch of computer science that trains machines to learn images and visual data in the same manner that humans do. In fields such as computer vision and augmented reality, this type of AI is often used.

1.2.6 Potential AI Techniques

A) Data mining, knowledge discovery and advanced analytics

Over the last decade, data mining has been a common word that is interchangeable with terms like knowledge mining from data, data or pattern analysis, knowledge extraction, knowledge discovery from data (KDD), etc. [1]. Data mining is described as the process of extracting useful patterns and knowledge from huge volumes of data [2], which is related to another popular term, **Data science**. Data science is commonly characterized as a concept that combines statistics, data analysis, and related approaches to study and investigate realities using data. Figure 1-2 explains how the knowledge discovery process works in general, according to Han et al.

Figure 1-2 : a general procedure for discovering knowledge [3]

Four types of analytics including descriptive, diagnostic, predictive and prescriptive are highlighted below, which can be used to develop data-driven models

Table 1-1: Types of analytical Methods

Analytical Methods	Definition	Data-driven model building	Examples
Descriptive analytics	It is the study of historical data in order to gain a better understanding of how a company has evolved.	Answer the question, What happened in the past ?	Summarizing past events, e.g., sales business data, social media usage, reporting general trends, etc.
Diagnostic analytics	It's a form of advanced analytics that looks at data or content to determine "why did it happen?". The purpose of diagnostic is to assist in the discovery of the problem's root cause.	Answer the question, whydidit happen?	Identify anomalies and casual relationships in order to estimate business loss, determine the influence of medications, etc.
Predictive Analytics	It explores data to answer a question,the primary purpose of predictive analytics is to identify and, in most cases, answer this question with a high degree of confidence.	Answer the question, what will happen in the future?	Predicting customer preferences, staff and resource needs. Identifying possible security breaches, recommending products, etc.
Prescriptive Analytics	This focuses on advising the optimal course of action based on data to maximize the Totaloutcomes and profitability.	Answer the question, What action should be taken?	Improving business management, maintenance, patient care and healthcare administration, determining optimal marketing strategies, etc.

B) Rule-Based modeling and decision-making

To analyze data in a meaningful way, a rule-based system is typically utilized to store and modify knowledge. A rule base is a knowledge base that contains a set of rules. In most cases, rules are written as IF-THEN statements of the form:

IF «antecedent » THEN « consequent ».

Such an IF-THEN rule-based expert system model can have the decision-making ability of a human expert in an intelligent system designed to solve complex problems and use knowledge-based reasoning [4]. The reason is that the rules in professional frameworks are easily understood by humans and are capable of representing relevant knowledge clearly and effectively. Furthermore, rule-based models may be quickly improved according to the demands by adding, deleting, or updating rules based on domain expert information, or recency, i.e. based on recent trends [5]

The term "rule-based system" was previously used to describe systems that used rule sets that were handcrafted or created by human. However, rule-based machine learning approaches could be more effective in terms of automation and intelligence, which include mainly classification and association rule learning techniques [4]. Several well-known classification techniques, such as decision trees, intrudTree, BehavDT, Ripple Down Rule Learner (RIDOR), Repeated Incremental Pruning to Produce Error Reduction, etc. exist with the capacity to generate rules. Based on support and confidence value, association rules are built by searching for frequent IF-THEN pattern data. Common association rule learning techniques such as AIS, Apriori, FP-Tree, RARM, Eclat, ABC-RUuleMiner, and others can be utilized to build a rule-based model using a given data set, e.g. provide a rule-based machine learning strategy for context-aware intelligent and adaptive mobile services. As a result, we can conclude that rule-based modeling can play an important role in the development of AI models as well as intelligent decision-making in a variety of applications to solve real-world issues.

C) Fuzzy Logic-Based approach

Fuzzy logic is a precise logic of imprecision and approximate reasoning [6]. This is a natural generation of conventional logic in which the degree of truth of a concept (known as membership value or degree of membership) can range from 0.0 to 1.0. Standard logic only applies to concepts that are either completely true (degree of truth 1.0) or completely false (0.0 degree of truth).It has been used to deal with the concept of partial truth, in which the truth may range from completely true to completely false (as 0.9 or 0.5). For instance, "if x is very large, do y, if x is not very large, do z". Here, the boundaries between very big and not too big may overlap. As a result, fuzzy logic-based model scanning recognizes, represents, manipulates, understands, and uses data and information that are vague and uncertain [6].

A fuzzy logic system's general architecture is depicted in Figure1-3. It usually consists of four sections:

- Fuzzification: it converts inputs, i.e. crisp numbers into fuzzy sets.
- Knowledge-base: it contains the set of rules and the IF-THEN conditions provided by the experts to govern the decision-making, based on linguistic information.

- Inference engine: it determines the matching degree of the current fuzzy input concerning each rule and decides which rules are to be fired according to the input field. Then, the fired rules are combined to form the control actions.
- Defuzzification: transforms the fuzzy sets obtained by the inference engine into a crisp value.

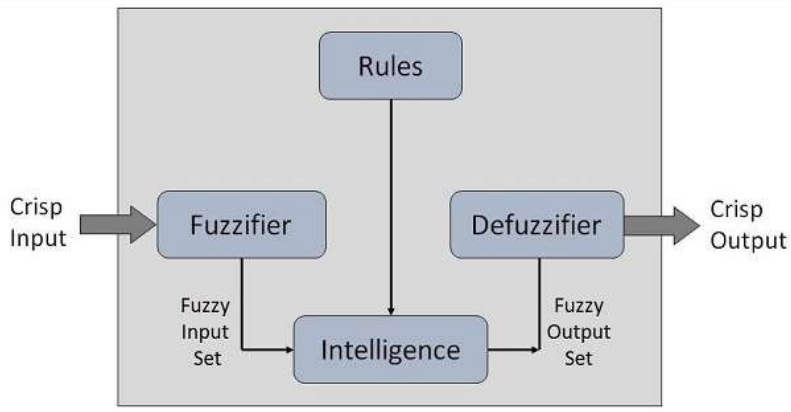


Figure 1-3 : a general architecture of a fuzzy logic system

Although machine learning models may discriminate between two (or more) object classes based on their ability to learn from data, fuzzy logic is favored when distinguishing features are vaguely defined and rely on human expertise and knowledge. Thus, the system may work with any type of input data, including imprecise, distorted, or noisy data, as well as with limited data. It is a suitable strategy to use in scenarios with real, continuous-valued elements because it uses data acquired in surroundings with such properties [7]. Overall, we can infer that fuzzy logic can reach reasonable results in a world of imprecision, uncertainty, and partial data, and that it may be useful in such settings when developing a model.

D) Knowledge representation, uncertainty reasoning and expert system modeling

Knowledge representation is the study of how an intelligent agent's beliefs, intents, and judgements may be expressed appropriately for automated reasoning. Reasoning is the process of using existing knowledge to conclude, make predictions or construct explanations. Many types of knowledge can be used in various application domains include descriptive, structural, procedural, meta and heuristic knowledge. As a result, an effective form of knowledge representation is necessary in designing an intelligent system. Several knowledgerepresentation approaches exist in the fields that can be utilized to construct a knowledge-based conceptual model including logical, semantic network, frame and production rules [8].

Potential knowledge representation strategies are summarized in the following:

- Ontology-based: in general, ontology is an explicit specification of conceptualization and a formal way to define the semantics of knowledge and data. [9]
- Rule-base: it typically consists of pairs of the condition, and corresponding action, which means, IF « condition » THEN « action » [4].
- Uncertainty and probabilistic reasoning: Probabilistic reasoning is a method of knowledge representation in which the concept of probability is used to signify the uncertainty in knowledge, and where probability theory and logic are combined to address the uncertainty [10].

Thus, knowledge representation and modeling are important for developing AI models as well as intelligent decision-making in a variety of application.

E) Case-basedreasoning (CBR)

Case-based reasoning is a cognitive science and artificial intelligence paradigm that depicts reasoning as primarily memory-based.CBR is concerned with the smart reuse of information from previously solved problems, as well as its adaptation to new and unsolved problems. The inference is a problem-solving approach based on the similarities between the current condition and previously solved problems in a repository. Its idea is that the more the two issues are similar, the more similar their solutions will be. Thus, case-based reasoners deal with new problems by obtaining previously stored cases that describe similar previous problem-solving experiences and modifying their answers to match new requirements. CBR becomes increasingly intelligent as the number of saved examples grows, and hence may be useful in such scenarios while creating a model. However, the system's efficiency will decline, as the time required to find and process relevant cases increases. A general design of case-based reasoning is shown in Figure 1-4.

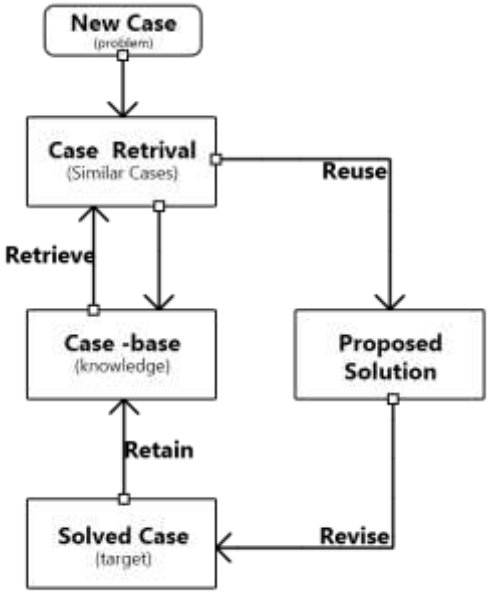


Figure 1-4 : A general case-based reasoning design

F) Text mining and Natural Language processing

Text mining, also known as text data mining, comparable to text analytics, is the technique of extracting meaningful information from a variety of text or written resources. Information retrieval, lexical analysis to investigate word frequency distributions, information extraction and data mining techniques are all part of text analysis which text mining accomplishes them through the use of a various analysis tools as natural language processing (NLP), natural language understanding (NLU) and natural language generation(NLG)…etc. while NLP is a text analysis technique that allows machines to interpret human speech, NLU is a subset of NLP that helps the machine to understand the data by understanding the context, semantic, syntax, intent and sentiment of the text. However, NLG is also a subset of NLP, it is a domain within Artificial intelligence that seeks to produce intelligible text based on three major transformer models, namely GPT, BERT and XLNet [11]. Overall, we may deduce that by combining learning techniques with natural language processing, computers can effectively evaluate, understand, and infer meaning from human speech or text, which could be valuable for developing textual AI models.

G) Visual analytics, Computer vision and pattern recognition

Computer vision is a subset of AI that enables computers and systems to extract meaningful information from digital photos, videos, and other visual inputs and act or make recommendations based on that data. Its goal is to understand and automate the activities that the human visual system can perform. As a result, this is concerned with the automated extraction, analysis and comprehension of relevant information from a single or a series of images. In terms of technology, it comprises the establishment of a theoretical and algorithmic foundation for achieving autonomous visual understanding by processing an image at the pixel level. Common tasks in the realm of visual analytics and

computer vision include object recognition or classification, detection, tracking, picture restoration, feature matching, image segmentation, scene reconstruction, video motion analysis, and so on. The basis for today's computer vision algorithms is Patterns recognition which is the automated recognition of patterns and regularities in data, it often involves the categorization (supervised learning) and grouping (unsupervised learning) of patterns [11]. Typically, learning techniques rather than static analysis is more effective in terms of automation and intelligence in such visual analytics. As a result, it's critical to develop effective visual AI models in a variety of application domains in order to solve real-world problems.

H) Hybridapproach, Searching and optimization

A hybrid approach is a blend of multiple approaches or systems to design a new and superior model. Thus, a hybrid strategy integrates the necessary techniques of ai depending on the demands. Machine learning and deep learning techniques and their hybridization can be used to tackle a wide range of real-world problems in a variety of fields including business, finance, healthcare, smart cities, cybersecurity and so on. As a result, combining multiple techniques could play a key role to build an effective AI model in the area.

1.3 Machine Learning

Machine learning is predominantly an area of artificial intelligence which has been a key component of digitalization solutions that has caught major attention in the digital arena, which is the study of computer algorithms that automate analytical model building, i.e. it usually provides systems with the ability to learn and enhance from experience automatically without being specifically programmed. In the current age, machine learning becomes popular in various application areas such as predictive analytics and intelligent decision-making, cybersecurity and threat intelligence, internet of things and smart cities, E-commerce and product recommendations, sustainable agriculture and many more because of its learning capabilities from the past and making intelligent decisions. In general, the effectiveness and the efficiency of a machine learning solution depend on the nature and characteristics of data and the performance of the learning algorithms. Its algorithms can be categorized into four major types as it is shown in figure 1-5.

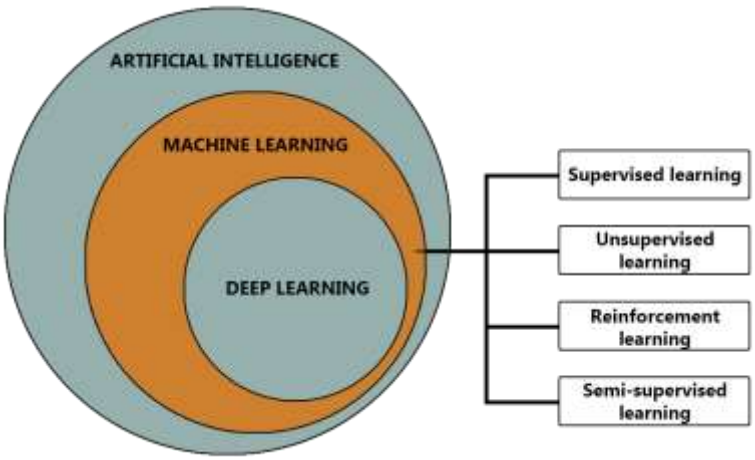


Figure 1-5 : Various types of machine learning techniques

The next section briefly explains each sort of learning techniques with the scope of their applicability to solve real-world problems.

1.3.1 Supervised learning

is typically the task of machine learning to learn a function that maps an input to an output based on sample input-output pairs [2], i.e. a task-driven approach, It infers a function from labeled training data and a collection of training examples.it is used when certain goals are determined to be achieved from a specific set of inputs. The most typical supervised tasks are classification that separates the data, and regression that fits the data. The popular techniques that can be used in supervised learning to solve various problems include Navies Bayes, K-nearest neighbors, Support vector machines, Decision Trees - ID3, C4.5, CART, BehavDT, IntrudTree, Ensemble learning, Random Forest, Linear regression, Support vector regression, etc.

1.3.2 Unsupervised learning

it analyzes unlabeled datasets without the need for human interference, i.e., data-driven process [2]. This is widely used for extracting generative features, identifying important trends and structures, groupings in results and exploratory purposes. The most prevalent unsupervised learning tasks are clustering, density estimation, feature learning, dimensionality reduction, finding association rules, anomaly detection, etc. The popular techniques for solving unsupervised learning tasks are clustering algorithms such as K-means, K-Mediods, CLARA, DBSCAN, hierarchical clustering, single linkage or complete linkage, BOTS, association learning algorithms such as AIS, Apriori, Apriori-TID and Apriori-Hybrid, FP-Tree, and RARM, Eclat, ABC-Rule Miner as well as feature selection and extracting techniques like Pearson Correlation, principal component analysis
, etc. that can be used to solve various unsupervised learning-related tasks, in accordance with the data's nature.

1.3.3 Semi-supervised learning

it can be defined as a hybridization of the supervised and unsupervised methods, as it operates on both labeled and unlabeled data. Thus, it falls between learning without supervision and learning with supervision .in the real world, labeled data could be rare in several contexts, and unlabeled data are numerous, where semi-supervised is useful. The ultimate goal of semi-supervised learning model is to deliver a better prediction result than that obtained from the model's labeled data alone. Some application areas where semi-supervised learning is used include machine translation, labeling data, fraud detection and text classification.

1.3.4 Reinforcement learning

is a type of machine learning algorithm that enables software agents and machines to automatically evaluate the optimal behavior in a particular context or environment to improve its efficiency [13]. In general, it is capable of perceiving and interpreting its surroundings, taking actions, and learning through trial and error, i.e., an environment-driven approach, in which the environment is typically modeled as a markov decision process and decisions are made using a reward function. The most prevalent reinforcement learning algorithms are Monte Carlo learning, Q-learning, and Deep Q Networks. Reinforce learning could be applied in autonomous driving tasks such as trajectory optimization, motion planning, dynamic pathing, and scenario-based learning rules for highways.

The following table summarized various types of machine learning with examples.
Table 1-2: various types of machine learning with examples.

Learning type	Model building	Examples
Supervised	Algorithms or models learn from labeled data (task-driven approach)	Classification, regression
Unsupervised	Algorithms or models learn from unlabeled data(Data-Driven approach)	Clustering, associations, dimensionalityreduction
Semi-supervised	Models are built using combined data(labeled+unlabeled)	Classification, clustering
Reinforcement	Models are based on reward or penalty(environment-driven approach)	Classification, control

1.3.5 Machine Learning Tasks and Algorithms

To efficiently develop data-driven systems, machine learning algorithms such as Classification analysis, regression, data clustering, feature engineering and dimensionality reduction, association rule learning, or reinforcement learning exist as shown in figure 1-6.

Thus, choosing a proper learning algorithm that is suitable for the target application in specific domain is challenging. The reason is that the purpose of learning algorithms is different, even within the same category, the results of different learning algorithms can differ based on the data features.

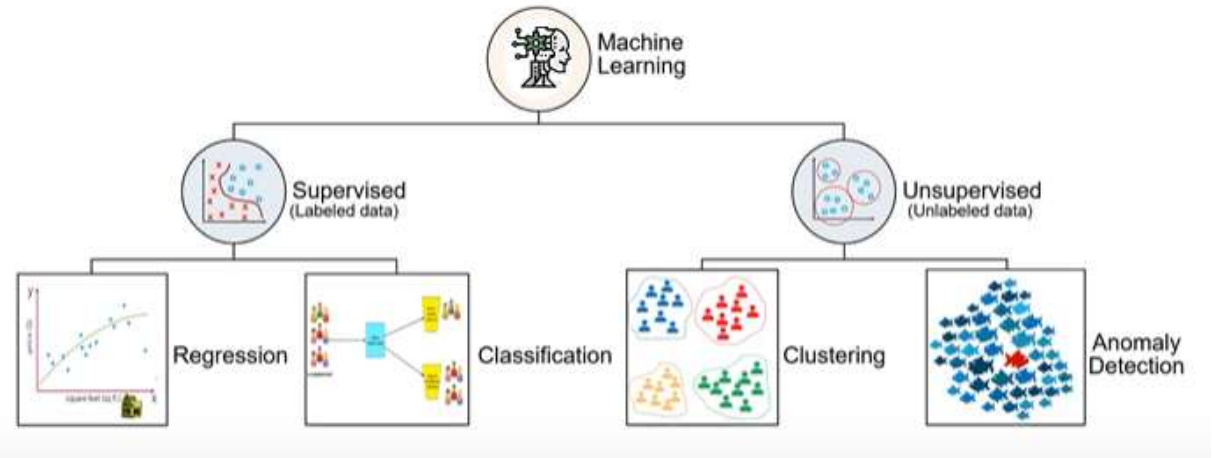


Figure 1-6 : Algorithms and Tasks of Machine Learning

A) Classification Analysis

In machine learning, classification is a supervised learning method, referring to a problem of predictive modeling as well, where a class label is predicted for a given example. Mathematically, it maps a function(F) from input variables(X) to output variables(Y) as target, label or categories. The common classification problems as it is shown in figure 1-7, include binary classification which refers to tasks having two class labels, Multiclass classification that refers to those classification tasks having more than two class, and Multi-label classification which is an important consideration where an example is associated with several classes or labels. Thus, it is a generalization of multiclass classification, where the problem's classes are hierarchically constructed, and each example may simultaneously belong to more than one class in each hierarchical level.

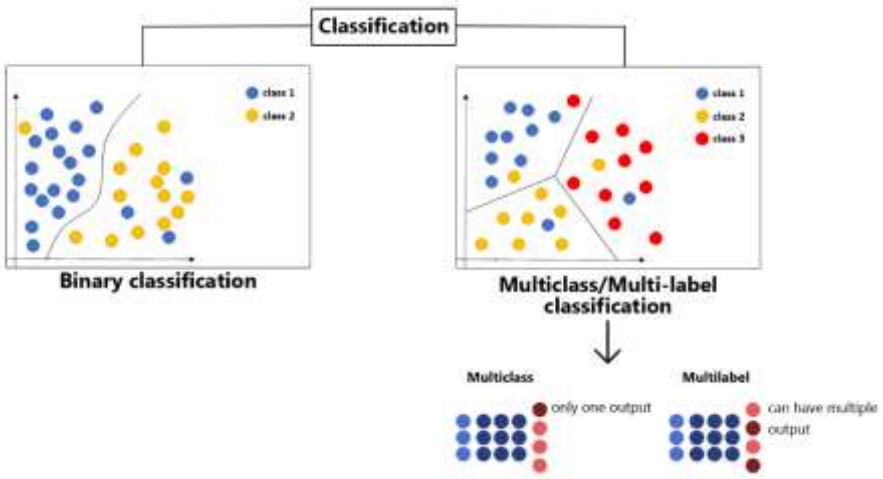


Figure 1-7 : Common classification issues

B) Regression Analysis

Regression analysis encompasses several machine learning methods for predicting a continuous (y) result variable based on the value of one or more (x) predictor variables. The most important distinction between classification and regression is that classification predicts distinct class labels, while regression predicts a continuous quantity as shown in figure 1-8. Some overlaps are frequently found between the two types of machine learning algorithms

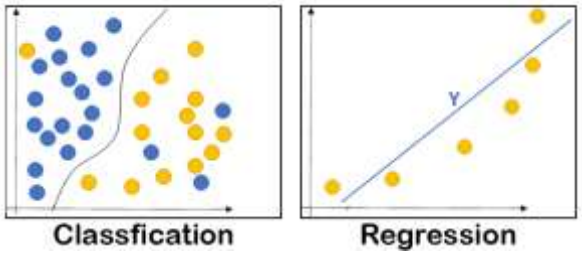


Figure 1-8 : Classification vs. Regression

C) Cluster Analysis

Also known as clustering, is an unsupervised machine learning technique for identifying and grouping related data points in large datasets without concern for the specific outcome, it groups a collection of objects in such a way that objects in the same category, called a cluster, are more similar to each other than objects in other groups [2]. It is frequently used as a data analysis technique to find interesting trends or patterns in data, such as groups of consumers based on their behavior.

The following is a brief discussion and summary of various types of clustering methods.

Partitioning methods

This clustering approach categorizes the data into multiple groups or clusters based on the features and similarities in the data. Depending on the nature of the target applications, data scientists or analysts typically determine the number of clusters to produce for clustering methods either dynamically or statically. The most common clustering algorithms based on partitioning methods are K-means, K-Mediods, CLARA etc.

Density-based methods

It uses the concept that a cluster in the data space is a contiguous region of high point density separated from other such clusters by contiguous regions of low point density to identify distinct groups or clusters and points that are not part of a cluster are considered as noise. The typical clustering algorithms based on density are DBSCAN, OPTICS, etc. Density-based methods frequently struggle with data clusters of similar density and high dimensionality.

Clustering algorithms

Many clustering algorithms with the ability to group data have been proposed in the machine learning and data science literature. The popular methods that are widely used in various application areas are:

-K-means clustering **-GMM clustering**

-Mean-shift clustering **-DBSCAN**

-Agglomerative hierarchical clustering

D) Reinforcement Learning

Reinforcement learning is a machine learning technique that allows an agent to learn through trial and error in an interactive environment by utilizing input from its actions and experiences. The RL method, in contrast to supervised learning, is based on interacting with the environment. The problem to be solved in reinforcement learning is defined as a Markov Decision Process (MDP), which is all about making decisions sequentially. An RL problem typically consists of four components: Agent, Environment, Rewards, and Policy. Model-based and Model-free techniques are the two broad categories of RL. Model-based RL is the process of deducing optimal behavior from an environment model by performing actions and observing the outcomes, which include the next state and the immediate reward [14]. Model-based approaches include AlphaZero and AlphaGo. A model-free approach, on the other hand, does not use the distribution of the transition probability and the reward function associated with MDP. Model-free algorithms include Q-learning, Deep Q Network, Monte Carlo Control, SARSA (State–Action–Reward–State–Action), and others. The key distinction between model-free and model-based learning is the policy network, which is required for model-based RL but not for model-free learning.

popular RL algorithms are discussed further below.

Monte Carlo methods

Monte Carlo techniques, also known as Monte Carlo experiments, are a broad category of computational algorithms that rely on repeated random sampling to produce numerical results [13]. The underlying idea is to use randomness to solve problems that are deterministic in theory. The three problem classes where Monte Carlo techniques are most commonly used are optimization, numerical integration, and drawing from a probability distribution.

Q-learning

Q-learning is a model-free reinforcement learning algorithm that instructs an agent on what action to take under what conditions [13]. It does not require an environment model (hence the term **model-free**), and it can deal with stochastic transitions and rewards without requiring adaptations. The 'Q' in Q-learning usually stands for quality, as the algorithm calculates the maximum expected rewards for a given behavior in a given state.

Deep Q-learning

The basic working step in Deep Q-Learning is that the initial state is fed into the neural network, which returns the Q-value of all possible actions as an output. Nonetheless, Q-learning works well when we have a relatively simple setting to overcome. Deep learning, on the other hand, can be used as a function approximator as the number of states and actions increases.

One of the fundamental machine learning paradigms, along with supervised and unsupervised learning, is reinforcement learning. Many real-world problems can be solved using RL, including game theory, control theory, operations analysis, information theory, simulation-based optimization, manufacturing, supply chain logistics, multiagent systems, swarm intelligence, aircraft control, robot motion control, and many others.

E) Artificial Neural Network

artificial neural networks are a subset of machine learning and are at the heart of deep learning algorithms. A typical neural network is mainly composed of many simple, connected processing elements or processors called neurons, each of which generates a series of real-valued activations for the target outcome.

Figure 1-9 shows a schematic representation of the mathematical model of an artificial neuron, i.e., processing element, highlighting input (Xi*)*, weight (*w*), bias (*b*), summation function (Σ), activation function (*f*) and corresponding output signal (*y*).

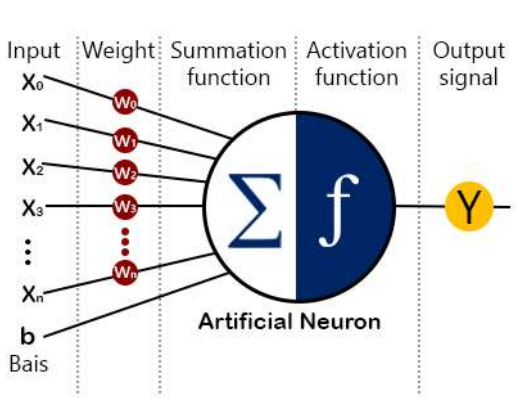


Figure 1-9 : A schematic representation of an artificial neuron's mathematical model

The Multi-layer Perceptron (MLP)

a supervised learning approach [15], is a type of feedforward artificial neural, since 1986 used algorithm "Backpropagation" [2], which is also known as the most basic building block of a neural network. Various optimization approaches, such as Stochastic Gradient Descent (SGD), Limited Memory BFGS (L-BFGS), and Adaptive Moment Estimation (Adam), are used during the training process. MLP necessitates the adjustment of several hyperparameters, including the number of hidden layers, neurons, and iterations, which

can make solving a complex model computationally expensive, on the other hand, has the advantage of learning non-linear models in real-time or online via partial fit [15].

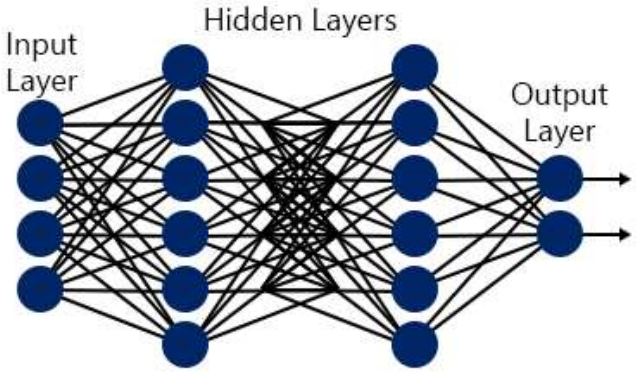


Figure 1-10 : A structure of an artificial neural network modeling with multiple processing layers

1.3.6 Machine learning algorithms

A) Support Vector Machine (SVM)

It's a widely used learning algorithm and one of the most powerful supervised classification techniques [16,17], originally introduced by Vapnik[18,19] and successively extended by a number of other academic. SVM can be trained to generate a model using training data{x1,…..xn}, where $xi \in N$ and class labels {y1, …y n} where $y_i \in \{-1,1\}$, The class of new testing is predicted using this model.

In high or infinite-dimensional space, a support vector machine constructs a hyper-plane or set of hyper-planes as shown in figure 1-11. Intuitively, the hyper-plane, which has the largest margin from the nearest training data points in any class, achieves a strong separation since. Margin means the maximal width of the slab width of the slab parallel to the hyperplane that has no interior data points. This helps to reduce the generalization error for classifying a new data point, i.e. the greater the margin, the lower the classifier's generalization error.

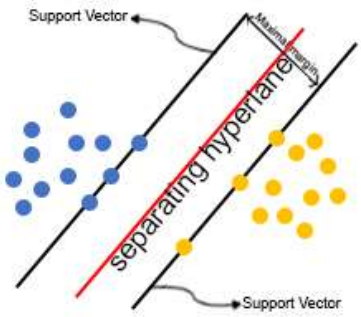


Figure 1-11 : SVM and the separating hyperplane

To separate data in the n-dimensional input space, Linear SVM uses the decision hyperplane defined as:

$$f(x) = w^T x + b = 0 \qquad (1)$$

Where$w$ is the hyperplane normal vector, $w \in \mathbb{R}^n$, and b/‖ w‖ is the perpendicular distance between the hyperplane and the origin (‖. ‖ is the 2-norm), b∈ℝ. This hyperplane is positioned such that the distance between the closet vectors of the opposite classes to the hyperplane is maximal. For two linearly separable classes (as already mentioned, with the class labels $y_i \in \{+1, -1\}$, the training data must satisfy the following conditions:

$$w^T x + b \leq 1 \qquad y_i = -1.$$

$$w^T x + b \geq 1 \qquad y_i = +1,$$

These two hyperplanes are called canonical hyperplanes. The width of the band formed by the canonical hyperplanes is 2/‖w||. To find the maximum margin separator, it is sufficient to search among the separators checking for all the examples $y_i \times f(x_i) \geq 1$, the separator for which ‖ω ‖ is minimal.

In cases where data points are clustered so that linear separation is not possible, the data points can be mapped into feature space (higher dimensional space) where a linear separation is possible. This hyperplane which is linear in feature space will be nonlinear in its corresponding input space [20]. Different kernel functions, including linear kernel ($\mathbf{x}^T\mathbf{x}'$), polynomial kernel (($\mathbf{x}^T\mathbf{x}' + 1)^d$), **RBF** kernel (exp(-$\gamma$||$\mathbf{xx}'$||2)) and sigmoidal kernel(tanh($\gamma\mathbf{xx}' + C$)) can be used in SVM for the nonlinear problem[21].

B) Naïve Bayes (NB)

The naive Bayes algorithm is based on the bayes theorem with the assumption of independence between each pair of features, in simple terms, a Naive bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.it works well and can be used for both binary and multi-class categories in many situations. To effectively classify the noisy instances in the data and to construct a robust prediction model, the NB classifier can be used [22]. Compared to more sophisticated approaches, the key benefit is that it needs a small amount of training data to estimate the necessary parameters and quickly. However, because of its strong assumptions on features independence, its performance may be harmed. The most frequent NB classifier are Gaussian, Multinomial, Complement, Bernoulli, and Categorical.

C) Linear Discriminant Analysis (LDA)

It is a linear decision boundary classifier created by fitting class conditional densities to data and applying baye's rule. Known as a generalization of fisher's linear discriminant which projects a given dataset into a lower-dimensional space, i.e. a reduction of

dimensionality that minimizes the complexity of the model or reduces the computational costs of the model. The typical LDA model frequently fits each class with a Gaussian density. Assuming that all classes share the same covariance matrix [15]. LDA is strongly connected to ANOVA (analysis of variance) and regression analysis, which seek to express one dependent variable as a linear combination of other features or measurements.

D) Logistic regression (LR)

Logistic Regression is another common probabilistic based statistical model used to solve classification problems in machine learning. LR typically uses a logistic function to estimate the probabilities, which is also referred to as the mathematically defined sigmoid function in eq 2. It is capable of overfitting high-dimensional datasets and performs well when the dataset can be separated linearly. In such cases, the regularization (L1 and L2) techniques can be used to avoid over-fitting. The assumption of linearity between the dependent and independent variables is regarded as a major drawback of Logistic Regression. It can be applied to both classification and regression problems, but it is most commonly applied to classification.

$$g(x) = \frac{1}{1 + exp^{-x}} \qquad (2)$$

E) K-nearest neighbors (KNN)

KNN is an instance-based learning or non-generalizing learning, also known as a lazy learning algorithm. it does not concentrate on developing a general internal model, instead, it stores all instances corresponding to training data in n-dimensional space. KNN analyzes data and classifies new data points based on similarity measures (e.g., Euclidean distance function) [15]. Classification is determined by a simple majority vote of the K nearest neighbors of each point.it is quite robust to noisy training data, and accuracy depends on data quality's, most difficult aspect of KNN is to choose the optimal number of neighbors to be considered. KNN can be used both for classification as well as regression.

F) Random forest

A random forest classifier is a well-known as ensemble classification technique used in machine learning and data science in a variety of application areas. This method uses parallel ensembling which parallelizes the fitting of several decision tree classifiers on various data set sub-samples and relies on majority voting or averages to determine the outcome or final result. It thus minimizes the over-fitting problem and increases the prediction accuracy and control [15]. As a result, the RF learning model with multiple decision trees is typically more accurate than a model based on a single decision tree. It combines bootstrap aggregation (bagging) and random feature selection to create a series of decision trees with controlled variation. RF is applicable to classification and regression problems, and it works well with both categorical and continuous values.

G) Decision tree

a well-known non-parametric supervised learning method, it is a graph that represents choices and their outcomes in form of a tree. The graph 's nodes represent an event or choice, and the graph's edges represent the decision rules or conditions. its learning methods are used for both classification and regression tasks. For DT algorithms, ID3, C4.5 and CART are well known. Furthermore, recently proposed BEhavDT and intrudTree are effective in relevant application domains such as user behavior analytics and cybersecurity analytics, respectively. By sorting down the tree from the root to some leaf nodes, DT classifies the instances. Instances are classified by checking the attribute defined by that node, starting at the root node of the tree, then moving down the tree branch corresponding to the attribute as shown in figure 1-12.

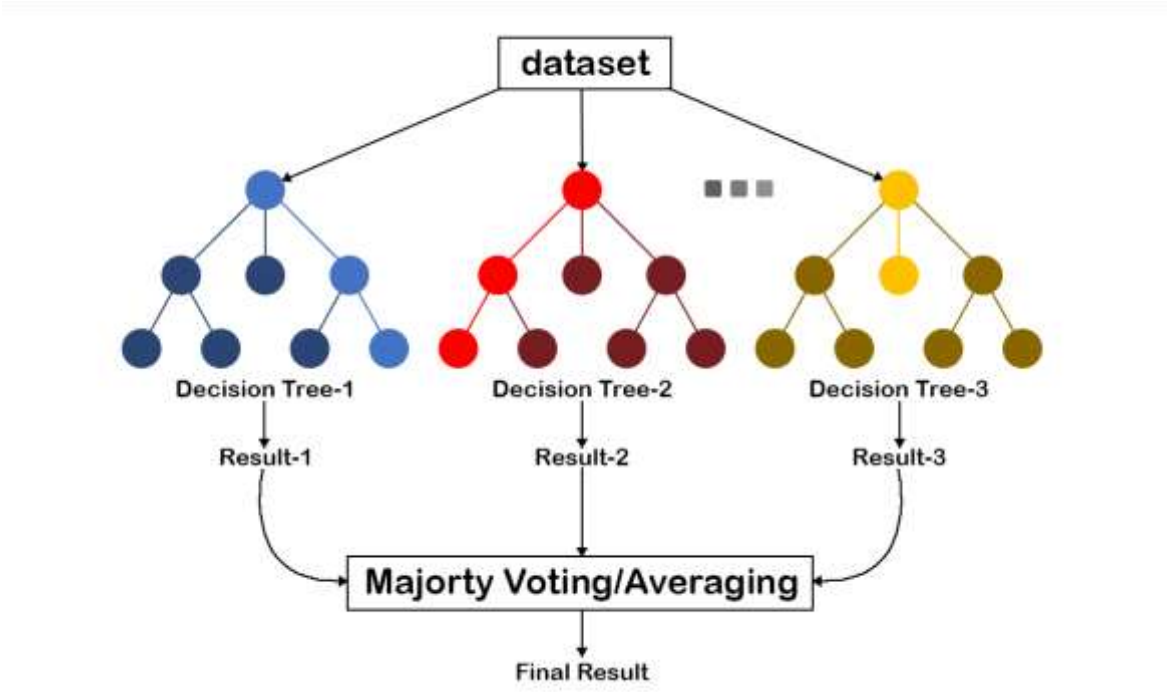


Figure 1-12 : An example of a random forest structure considering multiple

H) Adaptive Boosting

AdaBoost is an ensemble learning process that uses an iterative approach to improve poor classifiers by learning from their errors. This is created by Yoav Freund et al.and also known as metalearning. Unlike the random forest, which employs parallel ensembling, Adaboost uses sequential ensembling. It builds a powerful classifier by combining many low-performing classifiers to produce a high-accuracy classifier. AdaBoost is referred to as an adaptive classifier because it significantly improves the efficiency of the classifier, but in some instances, it can trigger overfits. AdaBoost is best used to improve the performance of decision trees and the base estimator [15] on binary classification problems, however, is sensitive to noisy data and outliers.

I) Extreme gradient boosting (XGBoost)

Like Random Forests, is an ensemble learning algorithm that creates a final model based on a collection of individual models, typically decision trees. Similarly, to how neural networks use gradient descent to optimize weights, the gradient is used to minimize the loss function. Extreme Gradient Boosting is a type of gradient boosting that considers more detailed approximations when determining the best model. To minimize loss and advance regularization, it computes second-order gradients of the loss function (L1 and L2) which reduces over-fitting and improves model generalization and performance. XGBoost is quick to interpret and can handle large-sized datasets well.

J) Rule-based classification

The term rule-based classification can be used to refer to any classification scheme that makes use of IF-THEN rules for class prediction. Several classification algorithms with rule generation capabilities exist, including Zero-R, One-R, decision trees, DTNB, Ripple Down Rule learner(RIDOR) and Repeated incremental Pruning to Produce Error Reduction(RIPPER).Among these techniques, the decision tree is one of the most common rule-based classification algorithms because it has several advantages such as being easier to interpret , the ability to handle high-dimensional data , simplicity and speed , good accuracy , and the ability to produce rules for human clear and understandable classification

For splitting, the most popular used criteria are gini for the Gini impurity and entropy for information gain that can be expressed mathematically as:

$$Entropy: \ H(x) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i) \qquad (3)$$

$$Gini(E) = 1 - \sum_{i=1}^{c} Pi^2 \qquad (4)$$

1.3.7 Dimensionality Reduction and Feature Learning

high-dimensional data processing is a challenging task for both researchers and application developers in machine learning and data science. Thus, dimensionality reduction, an unsupervised learning technique, is important because it improves human interpretations, reduces computational costs, and avoids overfitting and redundancy by simplifying models. Dimensionality reduction can be accomplished through both the feature selection and feature extraction processes. The primary distinction between feature selection and feature extraction is that **feature selection** retains a subset of the original features [23], whereas **feature extraction** creates entirely new ones [24]. These techniques are briefly discussed below.

A) Feature selection

The process of selecting a subset of unique features (variables, predictors) to use in building machine learning and data science models is known as feature selection. It reduces the complexity of a model by removing irrelevant or less important features,

allowing for faster training of machine learning algorithms. A correct and optimal subset of the selected features in a problem domain can reduce overfitting by simplifying and generalizing the model and increasing the model's accuracy [23]. Thus, feature selection is regarded as one of the fundamental concepts in machine learning that has a significant impact on the effectiveness and efficiency of the target machine learning model. Some popular techniques for feature selection include the Chi-squared test, the Analysis of variance (ANOVA) test, Pearson's correlation coefficient, and recursive feature elimination.

B) Feature extraction:

Feature extraction techniques in a machine learning-based model or system typically provide a better understanding of the data, a way to improve prediction accuracy, and a way to reduce computational cost or training time. The goal of feature extraction [25, 26] is to reduce the number of features in a dataset by creating new ones from old ones and then discarding the old ones. The majority of the information in the original set of features can be summarized using this new reduced set of features. Principal components analysis (PCA), for example, is frequently used as a dimensionality-reduction technique to extract a lower dimensional space and create new brand components from existing features in a dataset.

Many algorithms for reducing data dimensions have been proposed in the machine learning and data science literature. The popular methods that are widely used in various application areas are summarized below.

C) Variance threshold

The variance threshold is a simple basic approach to feature selection. This excludes all low variance features, i.e., all features whose variance does not exceed the threshold. By default, it eliminates all zero-variance characteristics, i.e., characteristics that have the same value in all samples. This feature selection algorithm considers only the (X) features, not the required (Y) outputs, and can thus be used for unsupervised learning.

D) Pearson correlation

Pearson's correlation is another method for understanding a feature's relationship to the response variable and can be used to select features. This method is also used to determine the relationship between features in a dataset. The resulting value is [-1, 1], where -1 indicates perfect negative correlation, +1 indicates perfect positive correlation, and 0 indicates that there is no linear correlation between the two variables. If X and Y are represented by two random variables, then the correlation coefficient between X and Y is defined as

$$r(X,Y) = \frac{\sum_{i=1}^{n}(X_I-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\overline{Y})^2}} \qquad (5)$$

E) ANOVA

ANOVA is a statistical tool used to compare the mean values of two or more groups that differ significantly from each other. ANOVA assumes a linear relationship between the variables and the target, as well as a normal distribution for the variables. The ANOVA method employs F tests to statistically test the equality of means. The results of this test's 'ANOVA F value' can be used for feature selection, where certain features independent of the goal variable can be omitted.

F) Chi square

The chi-square statistic estimates the difference between the observed and expected effects of a series of events or variables. The magnitude of the difference between the real and observed values, degrees of freedom, and sample size are all proportional to χ^2. For testing relationships between categorical variables, the chi-square is commonly used.

G) Recursive feature elimination

RFE is a brute force method for selecting features. RFE [15] fits the model and removes the weakest feature before it reaches the number of features specified. The model's coefficients or feature significance are used to rank features. RFE aims to remove model dependencies and collinearity by removing a small number of features recursively per iteration.

H) Model-based selection

Linear models penalized with the L1 regularization can be used to reduce the dimensionality of the data. Least absolute shrinkage and selection operator (Lasso) regression is a type of linear regression in which some of the coefficients are reduced to zero [15]. As a result, that feature can be eliminated from the model. Thus, the penalized lasso regression method is frequently used in machine learning to select a subset of variables.

I) Principal component analysis

In the field of machine learning and data science, principal component analysis is a well-known unsupervised learning approach. PCA is a mathematical technique that converts a set of correlated variables into a set of uncorrelated variables known as principal components [27,28]. Figure 1-13 shows the effect of PCA on various dimensions of space, where Fig. 8a depicts the original features in 3D space and Fig. 8b depicts the created principal components PC1 and PC2 onto a 2D plane and 1D line, respectively, with the principal component PC1.As a result, PCA can be used as a feature extraction technique to reduce dataset dimensionality and to build an effective machine learning model [24].

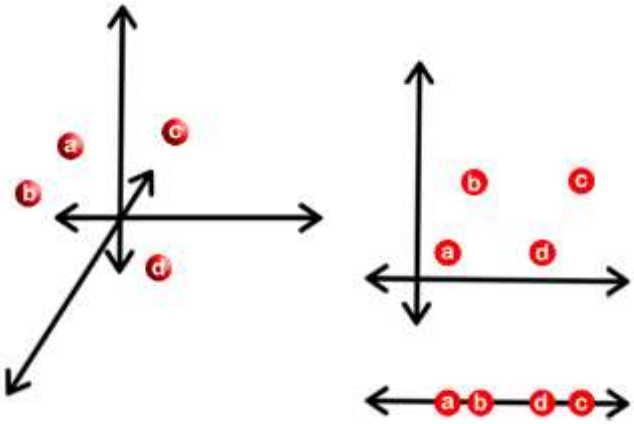


Figure 1-13 : An example of a principal component analysis (PCA) and created principal components

1.3.8 Real-World Applications of AI and ML

throughout the last several years, AI approaches have been effectively applied to a variety of issues in a variety of application areas such as cybersecurity, healthcare, business and social media, virtual reality and assistance, robotics, etc. The following table appears several AI tasks and techniques that are utilized nowadays.

Table 1-3: Real-World Applications of AI and ML

AI techniques	Application areas	Tasks	References
Machine learning	Healthcare	COVID-19 aid	Blumenstock et al. [29]
	Cybersecurity	Anomaly and Attack Detection	Sarker et al. [30] Sarker et al. [31]
	Smartcity	Smart parking pricing system	Saharan et al. [32]
	Recommendationsystems	Hotelrecommendation	Ramzan et al.[33]
	Virtual Assistant	An intelligent chatbot	Dhyani et al. [34]
	Visual Recognition	Facial expression analysis	Li et al. [35]
Data mining, knowledge discovery and advanced	Education	Decision support systems	Hamed et al. [36]
	Business	Maximisingcompetitiveadvantage	Alazab et al. [37]

analytics	Cybersecurity	Human-centred data mining	Afzaliseresht et al.[38]
	Diagnostic analytics	To mature gasfields	Poort et al. [39]
	Prescriptive analytics	Optimizingoutpatientappointment	Srinivas et al. [40]
Rule-based modeling and decisionmaking	Intelligent systems	Mining contextualrules	Sarker et. Al[41]
	Healthcare	Identifyingriskfactors	Borah et al. [42]
	Recommendation system	Web page recommendation	Bhavithra et al. [43]
	Smart systems	Risk prediction	Xu et al. [44]
Fuzzylogic-basedapproach	Healthcare	Heartdiseasediagnosis	Reddy et al. [45]
	Agriculture	Smart irrigation	Krishnan et al.[46]
	Cybersecurity	Network anomalydetection system	Hamamoto et al.[47]
	Business	Customer satisfaction	Kang et al. [48]
Knowledge representation, Uncertainty reasoning and Expert system modeling	Smart systems	Smart traffic monitoring	Goel et al. [49]
	cloudcomputing	Ontology data access control	Kiran et al.[50]
	cybersecurity	Vulnerability management	Syed et al. [51]
	Mobile expert system	Personalizeddecision-making	Sarker et al. [52]
Case-basedreasoning	Healthcare	Breast cancer management	Lamy et al. [53]
	Smart cities	Energy management	Gonzalez et al. [54]
	Smart Industry	Faultdetection system	Khosravani et al. [55]
	RecommendationSystems	Classification and regressiontasks	Corrales et al. [56]
Text mining and natural language processing	Sentiment analysis	Sentiment analysis of tweets	Phan et al. [57]
	Business	Product reviews sentiment	Onan et al. [58]
	Cybersecurity	Estimatingsecurity of events	Subramaniyaswamy et al. [59]

	Healthcare	Effectiveness of social media	Nawaz et al. [60]
Visual analytics, computer vision and pattern recognition	Healthcare	Cervical cancer diagnostics	Elakkiya et al. [61]
	Computer vision	Human falldetection	Arrou et al. [62]
	Visual Analytics	Navigation mark classification	Pan et al. [63]
Hybrid approach, searching and optimization	Mobile application	Personalizeddecision-making	Sarker et al. [52]
	Recommendationsystems	Personalizedhotelrecommendation	Ramzan et al. [33]
	Sentiment analysis	Tweet sentiment accuracyanalysis	phan et al. [57]
	Business	Customer satisfaction	Kang et al. [48]
	Cybersecurity	Optimum featureselection	Onah et al. [64]

PART II:

Deep learning

1.4 Deep Learning

DL is considered as a subset of ML and AI. It can be viewed as an AI function that mimics the human brain's data processing and a part of a wider family of machine learning approaches that use artificial neural networks (ANN). It became well-known as a result of Alexnet's outstanding performance on Imagenet 2012 [68].

Deep learning provides a computational architecture for learning from data by combining several processing layers, such as input, hidden, and output layers [65]. The main advantage of deep learning over traditional machine learning methods is that it performs better in several situations, particularly learning from large datasets. Figure 1-14 depicts the general performance of deep learning over machine learning when the amount of data increases. However, it may vary depending on the data characteristics and experimental set up.

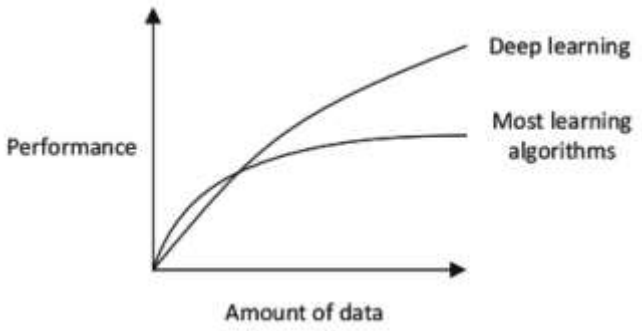


Figure 1-14: Machine learning and deep learning performance in general [65]

1.4.1 Various Forms of Data

As DL models learn from data, an in-depth understanding and representation of data is required to build a data-driven intelligent system in a specific application area. In the real world, data can take various forms, which are typically represented as below for deep learning modeling

A) Sequential data

is any type of data in which the order is important, i.e. a set of sequences. When building the model, it must explicitly account for the sequential nature of the input data. Sequential data includes text streams, audio fragments, video clips, and time-series data.

B) Image or 2D Data

A digital image is composed of a matrix, which is a rectangular array of numbers, symbols, or expressions arranged in rows and columns in a 2D array of numbers. A digital

image's four essential characteristics or fundamental parameters are matrix, pixels, voxels, and bit depth.

C) Tabular Data

A tabular dataset is made up of rows and columns. Tabular datasets like database tables, contain data in a columnar format. Each column (field) must be named, and each column can only contain data of the specified type. Overall, it is a logical and systematic arrangement of data in rows and columns based on data properties or features. Deep learning models can efficiently learn from tabular data and allow to create data-driven intelligent systems.

1.4.2 DL Properties and Dependencies

A DL model typically goes through the same stages of processing as machine learning modeling. Figure 1-15 depicts a deep learning workflow for solving real-world problems, which includes three processing steps: data understanding and preprocessing, DL model building and training, and validation and interpretation.

However, unlike ML modeling [66,67], feature extraction in the DL model is automated rather than manual. Machine learning techniques commonly used in various application areas include K-nearest neighbor, support vector machines, decision trees, random forests, naive Bayes, linear regression, association rules, and k-means clustering [12]. The DL model, on the other hand, includes convolution neural network, recurrent neural network, autoencoders, deep belief network and many more. The following section discuss the key properties and dependencies of DL techniques that must be considered before beginning work on DL modeling for real-world applications.

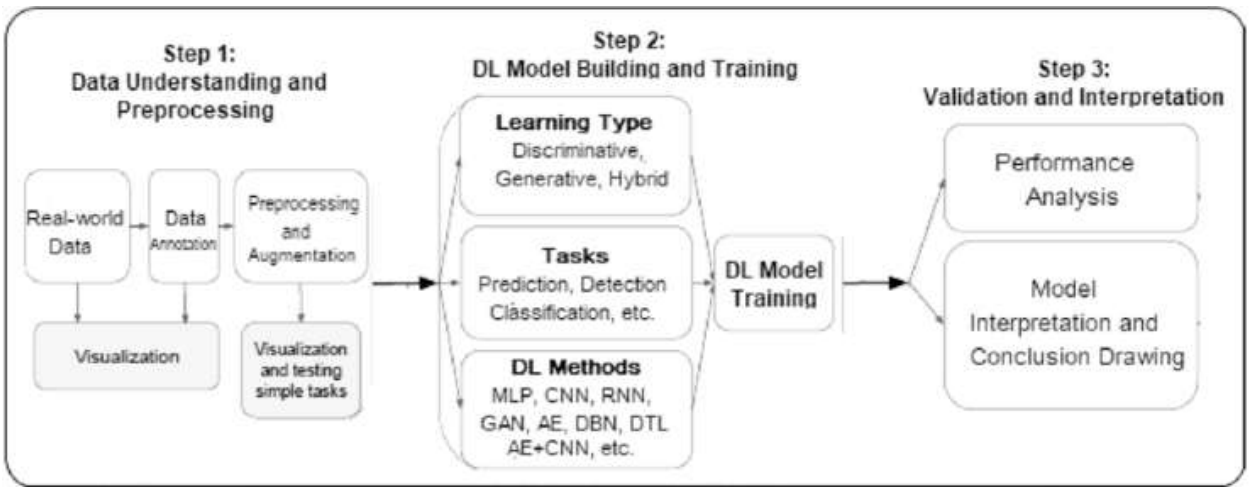


Figure 1-15 : A typical DL workflow to solve real-world problems[65]

A) Data Dependencies

Deep learning is typically dependent on a large amount of data to build a data-driven model for a specific problem domain. Deep learning algorithms frequently perform poorly when the data volume is small [68]. However, if the specified rules are used, the performance of standard machine-learning algorithms will be improved [68, 69].

B) Hardware Dependencies

While training a model with large datasets, DL algorithms necessitate large computational operations. As the larger the computations, the more the advantage of a GPU over a CPU, the GPU is mostly used to optimize the operations efficiently. Thus, to work properly with the deep learning training, GPU hardware is necessary. Therefore, DL relies more on high-performance machines with GPUs than standard machine learning methods [70,71].

C) Feature Engineering Process

is the process of extracting features (characteristics, properties, and attributes) from raw data using domain knowledge, A fundamental distinction between DL and other machine learning techniques is the attempt to extract high-level characteristics directly from data [72, 12]. Thus, DL decreases the time and effort required to construct a feature extractor for each problem**.**

D) Model Training and Execution time

In general, training a deep learning algorithm takes a long time due to the large number of parameters in the DL algorithm, thus, the model training process takes longer. For example, DL models can take more than a week to complete a training session, whereas ML algorithms take only seconds to hours [69, 71]. When compared to other machine learning methods, deep learning algorithms run extremely quickly during testing [71].

1.4.3 Deep Learning Techniques and Applications

in this section, various types of deep neural network techniques are discussed, which typically consider several layers of information-processing stages in hierarchical structures to learn. However, before diving into the specifics of the DL techniques, it's a good idea to review the different types of learning tasks, such as (i) Supervised: a task-driven approach that uses labeled training data, (ii) Unsupervised: a data-driven process that analyzes unlabeled datasets, (iii) Semi-supervised: a hybridization of both the supervised and unsupervised methods, and (iv) Reinforcement: an environment driven approach. Thus, DL techniques broadly divided into three major categories (i) deep networks for supervised or discriminative learning, (ii) deep networks for unsupervised or generative learning, and (iii) deep networks for hybrid learning combing both and relevant others, as shown in Figure 1-16.

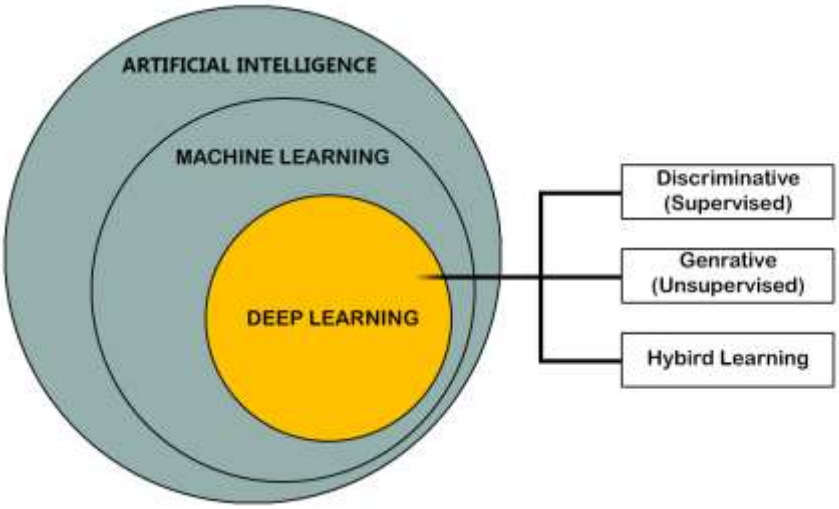


Figure 1-16: major categories of DL techniques

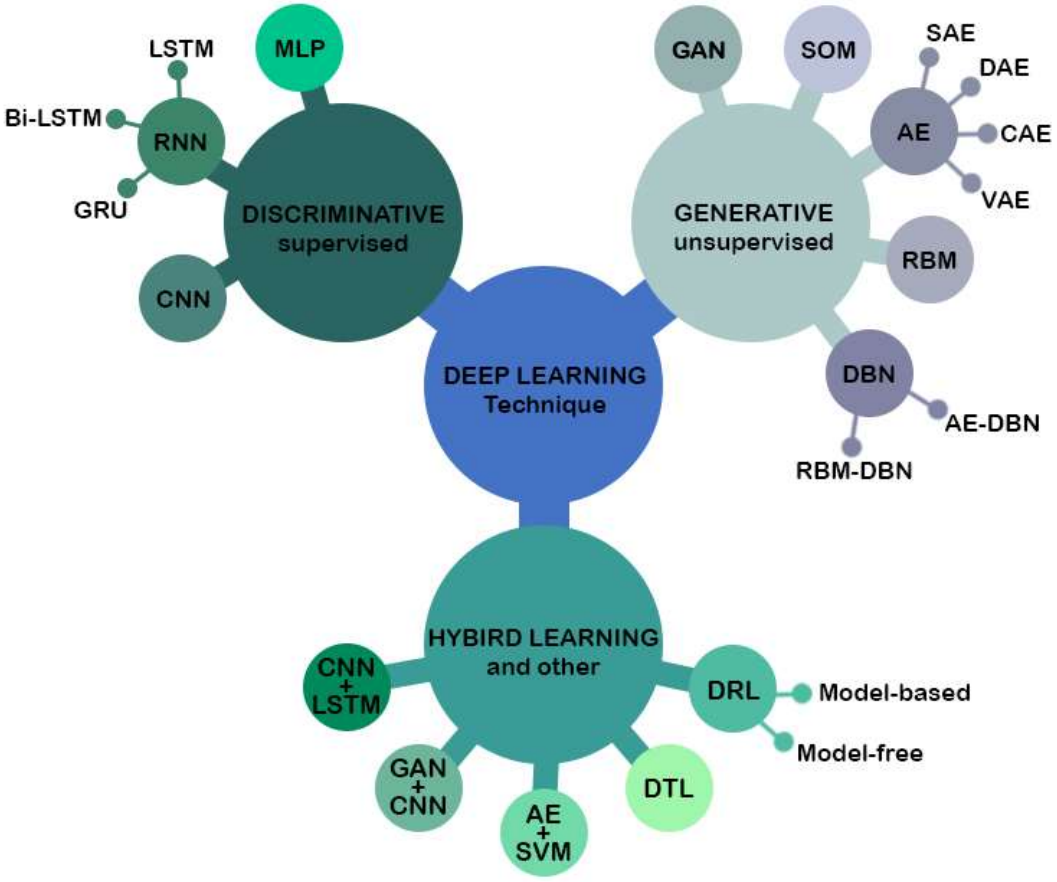


Figure 1-17: DL techniques

In the following, we briefly discuss each of these techniques shown in figure 1-17 according to their learning capabilities.

A) Deep Networks for supervised or discriminative Learning

This DL technique category is used to provide a discriminative function in supervised or classification applications. Typically, discriminative deep architectures are designed to provide discriminative power for pattern classification by describing the posterior distributions of classes conditioned on visible data [73]. Discriminative architectures mainly include Convolutional Neural Networks (CNN or ConvNet), Recurrent Neural Networks (RNN), Bayesian Neural Network along with their variants.

Convolutional Neural Network (CNN or ConvNet)

Convolutional neural networks (CNNs) have emerged as essential state-of-the-art deep learning algorithms in recent years. CNN is a feedforward neural network with a deep structure and one or more convolutional layers, modeled after the organization of animal visual cortex. The network employs a mathematical operation known as convolution, hence the name. Extraction. This is equivalent to a matrix multiplication. In contrast to the standard completely linked, CNN neurons have local connectivity (sparse interactions), therefore they do not have to be connected to all the outputs from the previous layer of neurons.

CNNs are intended to process data in the form of numerous arrays. 3D CNNs are capable of processing 3D data such as video or volumetric images. 2D CNNs work with 2D data such as images or audio spectrograms, whereas 1D CNNs work with 1D data such as signals and sequences.

The 1D CNNs used in this work are recent versions of the well-known 2D CNNs, having been introduced only a few years ago. 1D CNNs have quickly achieved state-of-the-art performance levels in a variety of applications, including cardiac arrhythmia classification, electrical motor fault detection, wind prediction, and acoustic waste sorting.

A typical CNN, as shown in figure 1-18, is generally a convolution layer adjacent to a pooling layer, alternating in turn and finally output by a fully connected layer.

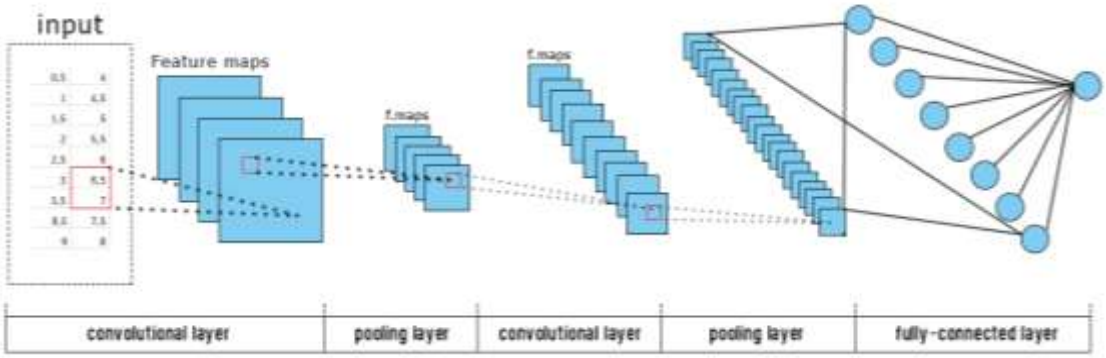


Figure 1-18: An example of a convolutional neural network

i) Convolution layer

The convolution operation is the core of the CNN, in which a small square matrix of numbers, known as a kernel (filter), is applied across the input, which is a matrix of numbers known as a tensor. As shown in figure 1-19, a hadamard product between the kernel and the tensor is calculated and summed at each position of the input to obtain the output value in the corresponding position of the target tensor. For example, if we utilize two-dimensional data I of pxq elements as input, we will most likely also want to employ a two-dimensional kernel K of nxn elements, the discrete convolution, often represented with an asterisk, is determined by the following formula:

$$V(i, j) = (I*K)(i, j) = \sum x^{n}{}_{=0} \quad \sum^{n}{}_{y=0} \qquad I(i + x, j + y)K(x, y).(6)$$

Where V(i,j) is the output value in the corresponding position of the target tensor.

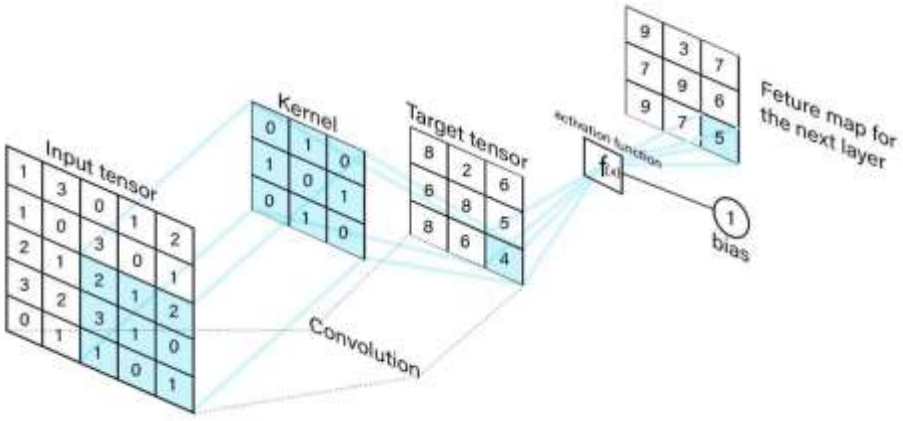


Figure 1-19: convolution layer neuron and the discrete convolution

A convolutional layer contains a set of kernels that need to be learned. Neurons with the same feature map in the same layer share the same kernel. As shown in Figure 1-19, each convolution layer neuron conducts a discrete convolution between the input and corresponding kernel to generate the neuron's input feature map, which is then passed through a nonlinear activation function to generate the neuron's output feature map. By stacking the feature maps of all kernels along the depth dimension, the convolutional layer's output volume is obtained. The convolutional layer's aim is to figure local intersections of features from the previous layer.

As in all neural networks, it is also possible to add biases to the forward operations. Each unique kernel in a convolutional layer is associated with its own bias. Therefore, For the j_{th} feature map in the i_{th} layer of the CNN, the value at the x_{th} row and y_{th} column is denoted as vijx,y and computed by the formula below :

$$vijx,y = f(b_{ij} + \sum_{m} \quad \sum_{p=0}^{P_i-1} wijmpv(xi+-p1),ym) \qquad (7)$$

Where f is the activation function,
b_{ij} the bias of the j^{th} kernel in the i^{th} layer,
m indexes over the set of feature maps in the $(i-1)^{th}$ layer connected to the current feature map, w_{ijm}^{p} is the value at the position p of the convolutional kernel, and P_i is the length of the convolutional kernel.

ii) Pooling layer

The pooling layer (subsampling layer), creates its own feature map by applying pooling operator to aggregate information within each small region of the input feature maps and then down sampling the results. The goal of a pooling layer is to produce a summary statistic of its input and to reduce the spatial dimensions of the feature maps [74].It is of note that there is no learnable parameter in any of the pooling layers. The most popular form of pooling operation is max pooling, which reports the maximal values in each rectangular neighborhood of each point (i,j), computed by the formula below:

$$v_{ij}^{x,y} = max_{1 \le q \le Q_i} \left(v_{(i-1)j}^{x+q,y}\right) \quad (8)$$

Where Q_i is the length of the pooling region.
The most common form of max pooling uses stride 2 together with pool size 2 [74], which corresponds to partitioning the feature map spatially into a regular grid of square with side 2 and taking the maximum value over such blocks for each input feature as shown in figure 1-20.

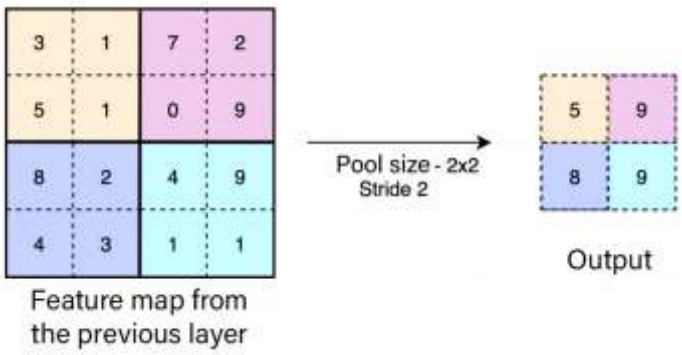


Figure 1-20: feature map and the output

iii) Fully connected layer

The fully connected (dense) layer, which is similar to the layer in a regular MLP, converts the retrieved information into final output, such as classification. Flattening is the process of converting the final convolution or pooling layer's output feature maps into a one-dimensional array of numbers (or vector) that may be used as input to the dense layer.

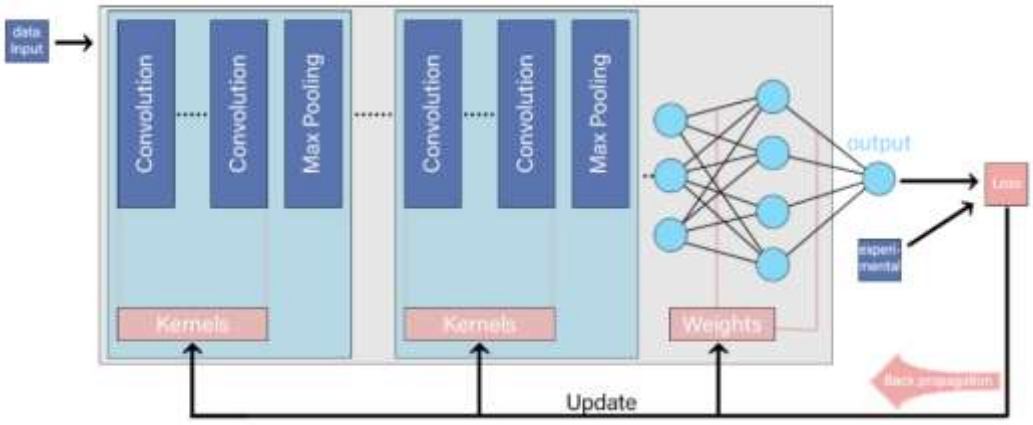


Figure 1-21: The fully connected layer

Recurrent Neural Network (RNN) and its Variants

A Recurrent Neural Network (RNN) is another popular neural network that uses sequential or time-series data and feeds the output from the previous step as input to the current stage [75, 76]. Recurrent networks, like feedforward and CNN, learn from training input but differ in that they have "memory," which allows them to influence current input and output by using information from previous inputs. Unlike traditional DNN, which assumes that inputs and outputs are independent of one another, RNN output is dependent on previous elements in the sequence. However, standard recurrent networks suffer from vanishing gradients, making learning long data sequences difficult. In following, several popular recurrent network variants that minimize issues and perform well in a wide range of real-world application domains are discussed

Long short-term memory (LSTM)

This is a popular type of RNN architecture that employs special units to solve the vanishing gradient problem, which was proposed by Hochreiter et al. [77]A memory cell in an LSTM unit can store data for long periods of time, and the flow of information into and out of the cell is controlled by three gates. For example, the 'Forget Gate' determines what information from the previous state cell will be memorized and what information will be removed that is no longer useful, while the 'Input Gate' determines which information should enter the cell state and the 'Output Gate' determines and controls the outputs. The LSTM network is considered one of the most successful RNNs because it solves the problems associated with training a recurrent network.

RNN/LSTM bidirectional

Bidirectional RNNs link two hidden layers running in opposite directions to a single output, allowing them to accept data from both the past and the future. Unlike traditional recurrent networks, bidirectional RNNs are trained to predict both positive and negative time directions at the same time. A Bidirectional LSTM, or BiLSTM, is an extension of the standard LSTM that can improve model performance on sequence classification problems

[78]. It is a sequence processing model made up of two LSTMs, one of which moves the input forward and the other backward. In natural language processing tasks, bidirectional LSTM is a popular choice.

Recurrent gated units (GRUs)

Cho et al. [79] introduced a Gated Recurrent Unit (GRU), a popular variant of the recurrent network that uses gating methods to control and manage information flow between cells in the neural network. The GRU is like an LSTM, however, has fewer parameters, as it has a reset gate and an update gate but lacks the output gate, as shown in Figure 1-22.

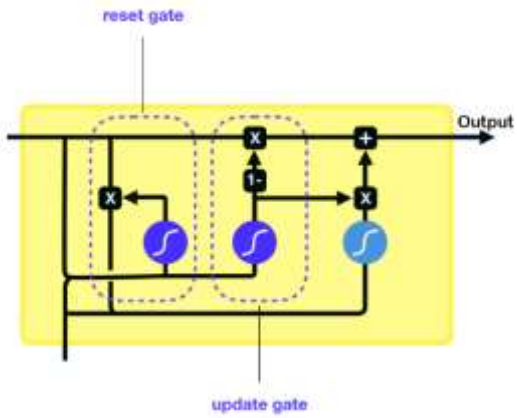


Figure 1-22: the Grus architecture

Thus, the primary distinction between a GRU and an LSTM is that a GRU has two gates (reset and update gates), whereas an LSTM has three gates (input, output, and forget gates). The GRU's structure enables it to capture dependencies from large sequences of data in an adaptive manner, without discarding information from earlier parts of the sequence. Thus, GRU is a slightly more streamlined variant that often provides comparable performance while being significantly faster to compute [80]. Although GRUs have been shown to perform better on smaller and less frequent datasets [80, 81], both variants of RNN have proven their effectiveness in producing the outcome.

Bayesian Neural Network

Bayesian neural networks (BNNs) are stochastic neural networks trained using a Bayesian approach or inference. it refers to the use of posterior inference to extend standard networks in order to control over-fitting. In general, the Bayesian approach employs statistical methodology to ensure that including model parameters, everything has a probability distribution attached to it (weights and biases in neural networks). Bayesian statistics offer a formalism to understand and quantify the uncertainty associated with deep neural network predictions. The Bayesian posterior for complex models such as artificial neural networks is a high dimensional and highly non-convex probability distribution [82].

A recurrent network's basic property is that it has at least one feedback connection, which allows activations to loop. This enables networks to perform temporal processing and sequence learning tasks like sequence recognition or reproduction, temporal association or prediction, and so on.

B) Deep Networks for Generative or Unsupervised Learning

This subset of deep learning techniques is frequently used to characterize high-order correlation properties or features for pattern analysis or synthesis, as well as the joint statistical distributions of visible data and their associated classes [73]. The fundamental idea behind generative deep architectures is that precise supervisory information, such as target class labels, is irrelevant during the learning process, the precise supervisory information, such as target class labels, is unimportant. As a result, the methods in this category are primarily used for unsupervised learning, as they are commonly used for feature learning or data generation and representation [83, 73]. Thus, generative modeling can also be used as preprocessing for supervised learning tasks, ensuring discriminative model accuracy. Deep neural network techniques commonly used for unsupervised or generative learning include Generative Adversarial Network (GAN), Autoencoder (AE), Restricted Boltzmann Machine (RBM), Self-Organizing Map (SOM), and Deep Belief Network (DBN), as well as variants.

Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN), created by Ian Goodfellow [84], is a type of neural network architecture for generative modeling that generates new plausible samples on demand. It involves automatically discovering and learning regularities or patterns in input data so that the model may be used to generate or output new examples from the original dataset. GANs are made up of two neural networks. a generator G that generates new data with properties similar to the original data and a discriminator D that predicts the likelihood of a subsequent sample being drawn from actual data rather than data provided by the generator. Thus, in GAN modeling, both the generator and the discriminator are trained to compete with one another. While the generator attempts to fool and confuse the discriminator by creating more realistic data, the discriminator tries to distinguish the genuine data from the fake data generated by G. GAN network deployment is generally intended for unsupervised learning tasks, but depending on the task, it has also proven to be a better solution for semi-supervised and reinforcement learning [85]. GANs are also used in state-of-the-art transfer learning research to enforce the alignment of the latent feature space [86] Inverse models, such as Bidirectional GAN (BiGAN) , can learn a mapping from data to latent space in the same way that a standard GAN model learns a mapping from a latent space to the data distribution. Overall, GANs have established themselves as a comprehensive domain of independent data expansion and as a solution to problems requiring a generative solution.

Auto-Encoder (AE) and Its Variants

An auto-encoder (AE) is a well-known unsupervised learning technique that employs neural networks to learn representations. Auto-encoders are typically used to work with

high-dimensional data, and dimensionality reduction describes how a set of data is represented. Encoder, code, and decoder are the three parts of an autoencoder. The encoder compresses the input and generates the code, which the decoder subsequently uses to reconstruct the input. Recently, AEs have been used to learn generative data models [87]. Many unsupervised learning tasks, such as dimensionality reduction, feature extraction, efficient coding, generative modeling, denoising, anomaly or outlier detection, and so on, make extensive use of the auto-encoder. PCA, which is also used to reduce the dimensionality of large data sets, is fundamentally similar to a single-layered AE with a linear activation function. Regularized autoencoders such as sparse, denoising, and contractive are useful for learning representations for later classification tasks, whereas variational autoencoders can be used as generative models [65].

i) Sparse Autoencoder (SAE)

A sparse autoencoder imposes a sparsity penalty on the coding layer as part of its training requirement. SAEs can have more hidden units than inputs, but only a limited number of hidden units can be active at the same time, resulting in a sparse model. Figure 1-23 depicts the schematic structure of a sparse autoencoder with several active units in the hidden layer.

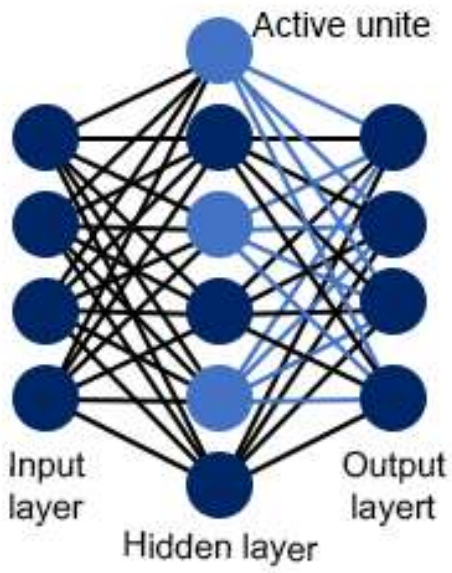


Figure 1-23: the schematic structure of a sparse autoencoder

This model is thus obliged to respond to the unique statistical features of the training data following its constraints [65].

ii) Autoencoder Denoising (DAE)

A denoising autoencoder is a variation on the basic autoencoder that attempts to improve representation (to extract useful features) by changing the reconstruction criterion, reducing the risk of learning the identity function in other words, it receives a corrupted data point as input and is trained to recover the original undistorted input as output by minimizing the average reconstruction error over the training data, i.e, cleaning the

corrupted input, or denoising. As a result, in the context of computing, DAEs can be thought of as extremely powerful filters that can be used for automatic pre-processing. For example, a denoising autoencoder could be used to automatically pre-process an image, improving its quality for recognition accuracy [65].

iii) Auto encoder Contractive (CAE)

Rifai et al. [89] proposed a contractive autoencoder to make autoencoders robust to small changes in the training dataset. A CAE includes an explicit regularize in its objective function that forces the model to learn an encoding that is robust to small changes in input values. As a result, the learned representation's sensitivity to the training input is reduced. While DAEs encourage the robustness of reconstruction as discussed above, CAEs encourage the robustness of representation

iv) Variational Autoencoder (VAE)

A variational autoencoder has a fundamentally unique property that distinguishes it from the classical autoencoder discussed above, which makes it so effective for generative modeling. Unlike traditional autoencoders, which map the input onto a latent vector, VAEs map the input data into the parameters of a probability distribution, such as the mean and variance of a Gaussian distribution. A VAE assumes that the source data has an underlying probability distribution and then attempts to discover the distribution's parameters. Although this approach was originally designed for unsupervised learning, its application in other domains such as semi-supervised learning and supervised learning has been demonstrated.

Although the earlier concept of AE was typically used for dimensionality reduction or feature learning, as mentioned above, AEs have recently been brought to the forefront of generative modeling, even the generative adversarial network is one of the popular methods in the area. Overall, as a conclusion auto-encoders and their variants can be useful for unsupervised feature learning using neural network architecture. [65]

Kohonen Map or Self-Organizing Map (SOM)

A Self-Organizing Map (SOM) or Kohonen Map [90] is another unsupervised learning technique for creating a low-dimensional (usually two-dimensional) representation of a higher-dimensional data set while preserving the data's topological structure. SOM is a neural network-based dimensionality reduction algorithm that is commonly used in clustering. A SOM adapts to the topological form of a dataset by repeatedly moving its neurons closer to the data points, allowing us to visualize enormous datasets and find probable clusters. The input layer is the first layer of a SOM, and the output layer or feature map is the second layer. In contrast to other neural networks that use error-correction learning, such as backpropagation with gradient descent [2], SOMs use competitive learning, which employs a neighborhood function to retain the topological

features of the input space. SOM is widely used in a variety of applications, including pattern recognition, health or medical diagnosis, anomaly detection, and detection of virus or worm attacks [91, 92]. The main advantage of using a SOM is that it can make high-dimensional data easier to visualize and analyze in order to understand the patterns. The dimensionality reduction and grid clustering make it simple to spot similarities in the data. As a result, depending on the data characteristics, SOMs can play a critical role in developing a data-driven effective model for a specific problem domain.

Boltzmann Machine Restricted (RBM)

A Restricted Boltzmann Machine (RBM) [93] is another type of generative stochastic neural network that can learn a probability distribution over its inputs. Boltzmann machines consist of visible and hidden nodes, and each node is linked to every other node, which helps us understand irregularities by learning how the system works normally circumstances. RBMs are a type of Boltzmann machine in which the number of connections between the visible and hidden layers is limited. This constraint allows training algorithms like the gradient-based contrastive divergence algorithm to be more efficient than Boltzmann machine training algorithms in general. RBMs have been used in a variety of applications, including dimensionality reduction, classification, regression, collaborative filtering, feature learning, topic modeling, and many more. They can be trained either supervised or unsupervised in the field of deep learning modeling, depending on the task. Overall, RBMs can automatically recognize patterns in data and develop probabilistic or stochastic models that are used for feature selection or extraction. as well as establishing a strong belief network [65].

Deep Belief Network (DBN)

A Deep Belief Network (DBN) [94] is a multi-layer generative graphical model composed of stacking several individual unsupervised networks, such as AEs or RBMs, that use the hidden layer of each network as the input for the next layer, i.e. connected sequentially. Thus, a DBN can be divided into (i) AE-DBN which is known as stacked AE, and (ii) RBMDBN that is known as stacked RBM, where AE-DBN is composed of autoencoders and RBM-DBN is composed of restricted Boltzmann machines, discussed earlier. The ultimate goal is to develop a faster-unsupervised training technique for each sub-network that depends on contrastive divergence. Based on its deep structure, DBN can capture a hierarchical representation of input data. It is based on the principle of training unsupervised feed-forward neural networks with unlabeled data before fine-tuning the network with labeled input. One of the most significant advantages of DBN over traditional shallow learning networks is the detection of deep patterns, which allows for reasoning abilities and the capture of the deep difference between normal and erroneous data. A continuous DBN is simply a standard DBN that allows for a continuous range of decimals rather than binary data. Overall, due to its strong feature extraction and classification capabilities, the DBN model can play a key role in a wide range of high-dimensional data applications and become one of the significant topics in the field of neural networks [65].

To summarize, the generative learning techniques discussed above typically allow to generate a new data representation through exploratory analysis. As a result, deep generative networks can be used as preprocessing for supervised or discriminative learning tasks while also ensuring model accuracy, whereas unsupervised representation learning can improve classifier generalization.

C) Deep Networks for Hybrid Learning and Other Approaches

In addition to the deep learning categories discussed above, hybrid deep networks and several other approaches, such as deep transfer learning (DTL) and deep reinforcement learning (DRL), are popular.

Hybrid Deep Neural Networks

Generative models are adaptable, with the ability to learn from both labeled and unlabeled data. Discriminative models, on the other hand, are unable to learn from unlabeled data but outperform their generative counterparts in supervised tasks. A framework for simultaneously training deep generative and discriminative models can benefit from both models, which motivates hybrid networks. Hybrid deep learning models are typically composed of multiple (two or more) deep basic learning models, where the basic model is a previously discussed discriminative or generative deep learning model. The three categories of hybrid deep learning models listed below may be useful for solving real-world problems based on the integration of various basic generative or discriminative models. These are as follows:

-Hybrid Model 1: A combination of generative and discriminative models used to extract more meaningful and robust features. CNN+LSTM, AE+GAN, and other combinations are possible.

– Hybrid Model 2: A generative model is combined with a discriminative model. DBN+MLP, GAN+CNN, AE+CNN, and other combinations are possible.

– Hybrid Model 3: A generative or discriminative model is combined with a non-deep learning classifier.

Examples include AE+SVM, CNN+SVM, and others. Thus, depending on the intended use, hybrid models can be either classification-focused or non-classification-focused. However, the majority of hybrid learning-related studies in the field of deep learning are classification-focused or supervised learning tasks that enable to improve the quality and quantity of training data, providing additional information for classification.

Deep Transfer Learning (DTL)

Transfer Learning is a technique for effectively using previously learned model knowledge to solve a new task with minimal training or fine-tuning. In comparison to traditional machine learning techniques [12], DL requires a large amount of training data. As a result,

the requirement for a large volume of labeled data is a significant barrier to addressing some critical domain-specific tasks, particularly in the medical sector, where creating large-scale, high-quality annotated medical or health datasets is both difficult and costly. Furthermore, the standard DL model demands a lot of computational resources, such as a GPU-enabled server, even though researchers are working hard to improve it. As a result, Deep Transfer Learning (DTL), a DL-based transfer learning method, may be useful in addressing this issue. Figure 1-24 depicts the general structure of the transfer learning process, in which knowledge from the pre-trained model is transferred into a new DL model. It's especially popular in deep learning right now because it allows you to train deep neural networks with very little data [65].

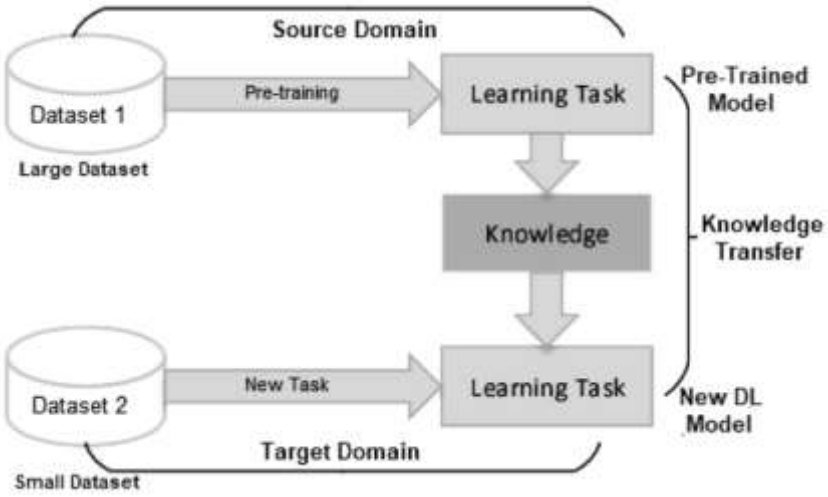


Figure 1-24 : the general structure of the transfer learning process

Transfer learning is a two-stage approach to training a deep learning model that consists of a pre-training step and a fine-tuning step in which the model is trained on the target task. Since deep neural networks have gained popularity in a wide range of fields, a large number of DTL methods have been presented, necessitating categorization and summarization. DTL can be classified into four categories based on the techniques used in the literature. These are (i) instances based deep transfer learning that utilizes instances in source domain by appropriate weight, (ii) mapping-based deep transfer learning that maps instances from two domains into a new data space with better similarity, (iii) network-based deep transfer learning that reuses the partial of network pretrained in the source domain, and (iv) adversarial based deep transfer learning that uses adversarial technology to find transferable features that both suitable for two domains. Adversarial-based deep transfer learning has exploded in popularity in recent years due to its high effectiveness and practicality. Depending on the circumstances between the source and target domains and activities, transfer learning can also be classified as inductive, transudative, or

unsupervised. While most current research is focused on supervised learning, how deep neural networks can transfer knowledge in unsupervised or semi-supervised learning may pique the interest of researchers in the future. Natural language processing, sentiment classification, visual recognition, speech recognition, spam filtering, and other fields benefit from DTL techniques [65].

Deep Reinforcement Learning (DRL)

Reinforcement learning approaches the sequential decision-making problem differently than the other approaches. In reinforcement learning, the concepts of an environment and an agent are frequently introduced first. The agent can take a series of actions in the environment, each of which has an effect on the state of the environment and can result in possible rewards (feedback) - "positive" for good sequences of actions that result in a "good" state, and "negative" for bad sequences of actions that result in a "bad" state. Reinforcement learning's goal is to learn good action sequences through interaction with the environment, which is commonly referred to as a policy. Deep reinforcement learning (DRL or deep RL) [95] combines neural networks with a reinforcement learning architecture to enable agents to learn appropriate actions in a virtual environment, as illustrated in Figure 1-25.

Model-based reinforcement learning (RL) is based on learning a transition model that allows for modeling of the environment without directly interacting with it, whereas model-free RL methods learn directly from interactions with the environment. Q-learning is a well-known model-free reinforcement learning (RL) technique for determining the best action-selection policy for any (finite) Markov Decision Process (MDP). MDP is a mathematical framework for modeling state, action, and reward decisions. Deep Q-Networks, Double DQN, Bi-directional Learning, Monte Carlo Control, and other techniques are also used in this area. It incorporates DL models, such as Deep Neural Networks (DNN), based on the MDP principle, as policy and/or value function approximators in DRL methods. CNN, for example, can be used in RL agents to learn directly from raw, high-dimensional visual inputs [65].

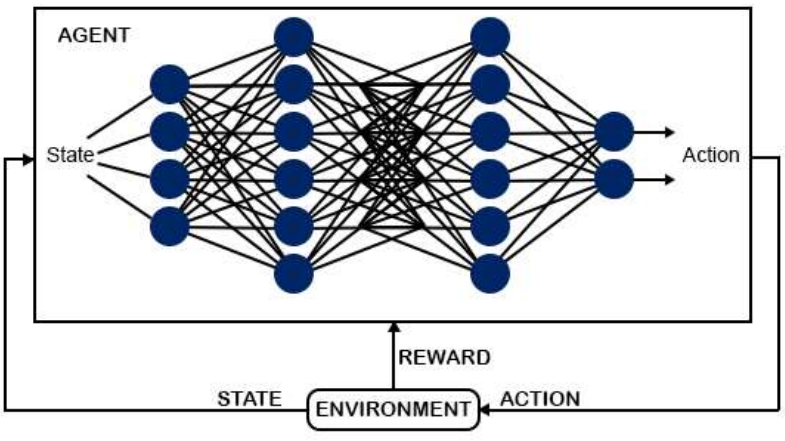


Figure 1-25 : Deep reinforcement learning combination

1.4.4 Real-World Applications of DL

Deep learning has been successfully applied to numerous problems in a variety of application areas over the last few years. Table 1-4 summarized various deep learning tasks and techniques that are used to solve the relevant tasks in several real-world applications areas.

Table 1-4: Real-World Applications of DL

Application areas	Tasks	Methods	References
Healthcare and Medical applications	Regular healthfactorsanalysis	CNN-based	Ismail et al.[96]
	Identifyingmaliciousbehaviors	RNN-based	Xue et al.[97]
	Coronary heart disease risk prediction	Autoencoderbased	Amarbayasgalan et al. [98]
	Cancer classification	Transfer learningbased	Sevakula et al. [99]
	Diagnosis of COVID-19	CNN and BiLSTMbased	Aslan et al. [100]
	Detection of COVID-19	CNN-LSTM based	Islam et al. [101]
Natural LanguageProcessing	Textsummarization	Auto-encoder based	Yousefi et al. [102]
	Sentiment analysis	CNN-LSTM based	Wang et al. [103]
	Sentiment analysis	CNN and Bi-LSTM based	Minaee et al. [104]
	Aspect-level sentiment classification	Attention-based LSTM	Wang et al. [105]
Speech recognition	Distant speech recognition	Attention-based LSTM	Zhang et al. [106]
	Speech emotion classification	Transfer learningbased	Latif et al. [107]
	Emotion recognition from speech	CNN and LSTM based	Satt et al. [108]
Cybersecurity	Zero-day malware detection	Autoencoders and GAN based	Kim et al. [109]
	Security incidents and fraud analysis	SOM-based	Lopez et al. [110]
	Android malware detection	Autoencoder and CNN based	Wang et al. [111]
	intrusiondetection classification	DBN-based	Wei et al. [112]
	DoSattackdetection	RBM-based	Imamverdiyev et al. [113]
	Suspicious flow detection	Hybriddeep-learning-based	Garg et al. [114]
	Network intrusion detection	AE and SVM based	Al et al. [115]

IoT and Smart cities	Smart energy management	CNN and Attention mechanism	Abdel et al. [116]
	Particulatematterforecasting	CNN-LSTM based	Huang et al. [117]
	Smart parking system	CNN-LSTM based	Piccialli et al. [118]
	Disaster management	DNN-based	Aqib et al. [119]
	Air qualityprediction	LSTM-RNN based	Kok et al. [120]
	Cybersecurity in smart cities	RBM, DBN, RNN, CNN, GAN	Chen et al. [120]
Smart Agriculture	Predicting life time and mechanical performance degradation of multilayer Greenhouse	CNN-SVM	AidLahcene, Djenane Mouloud et Dehbi Abdelkader[121]
	A smart agriculture IoT system	RL-based	Bu et al. [123]
	Plant diseasedetection	CNN-based	Ale et al. [124]
	Automatedsoilqualityevaluation	DNN-based	Sumathi et al. [125]
Business and Financial Services	Predictingcustomers' purchasebehavior	DNN based	Chaudhuri [126]
	Stock trend prediction	CNN and LSTM based	anuradha et al. [127]
	Financial loan default prediction	CNN-based	Deng et al. [128]
	Power consumptionforecasting	LSTM-based	Shao et al. [129]
Virtual Assistant and Chatbot Services	An intelligent chatbot	Bi-RNN and Attention model	Dhyani et al. [34]
	Virtual listener agent	GRU and LSTM based	Huang et al. [130]
	Smart blind assistant	CNN-based	Rahman et al. [131]
Object Detection and Recognition	Object detection in X-ray images	CNN-based	Gu et al. [132]
	Object detection for disaster response	CNN-based	Pi et al. [133]
	Medicine recognition system	CNN-based	Chang et al. [134]
	Face recognition in IoT-cloud environment	CNN-based	Masud et al. [135]
	Food recognition system	CNN-based	Liu et al. [136]

	Affect recognition system	DBN-based	Kawde et al. [135]
	Facial expression analysis	CNN and LSTM based	Li et al. [35]
Recommendation and Intelligent system	Hybridrecommender system	DNN-based	Kiran et al. [136]
	Visual recommendation and search	CNN-based	Shankar et al. [137]
	Recommendation system	CNN and Bi-LSTM based	Rosa et al. [138]
	Intelligent system for impaired patients	RL-based	Naeem et al. [139]
	Intelligent transportation system	CNN-based	Wang et al. [140]

1.5 Conclusion

In this chapter, we have presented a comprehensive view on AI-based modeling including the principles and capabilities of potential AI techniques such as Machine learning and Deep learning that can play an important role in developing intelligent and smart systems in a variety of real word application such as finance, smart cities, healthcare, agriculture and many others .

CHAPTER II

Nonlinear Regression

2.1 Introduction

Although the area of Artificial intelligence is huge, DL has emerged as the next major technological milestone, influencing the future of practically every business by making every process better, faster and more precise. As DL models learn from data, there are two general approaches to multivariate data analysis. Approaches that treat each variable equally, with the goal of understanding the correlation structure between the variables or reducing the data space dimension, examples include component, factor and cluster analysis. Another approach is to treat one variable as a response and the others as predictors, with the goal of understanding the variation in the response that can be explained partially by predictors. These are known as regression models. It is widely used when the dependent and independent variables are linked in a linear or non-linear fashion.

2.2 Regression analysis

Regression analysis is a statistical technique for investigating and modeling the relationship between variables, i.e., it is an important component of multivariate analysis, since it allows to focus on the effects of predictors on the response. Regression models attempt to partially explain the variation in the response by the predictors. In another words, regression models attempt to find the approximate relationship between the response and predictors. A regression model is an effective statistical tool for determining such an approximation. Its Applications are numerous and occur in almost every field, including engineering, the physical and chemical sciences, economics, management, life and biological sciences, and the social sciences. In fact, regression analysis may be the most widely used statistical technique.

Some common types of regression algorithms include linear, polynomial, lasso, and ridge regression, among others are briefly explained below.

2.2.1 Simple and multiple linear regression

This is a well-known regression technique as well as one of the most popular ML modeling techniques. In this technique, the dependent variable is continuous, the independent variable(s) may be continuous or discrete, and the form of the regression line is linear. Linear regression uses the best fit straight line to create a relationship between the dependent variable (Y) and one or more independent variables (X) (also known as the regression line). It is defined by the following equations:

$$y = a + bx + e \quad (9)$$

$$y = b_1x_1 + b_2x_2 + \cdots + b_nx_n + e \quad (10)$$

Where a is the intercept, b is the slope of the line, and e is the error term. based on the given predictor variable(s), this equation can predict the value of the target variable. Multiple linear regression is an extension of simple linear regression that allows two or more predictor variables to model a response variable, y, as a linear function defined in eq9.whereassimple linear regression has only 1 independent variable, defined in eq 10.

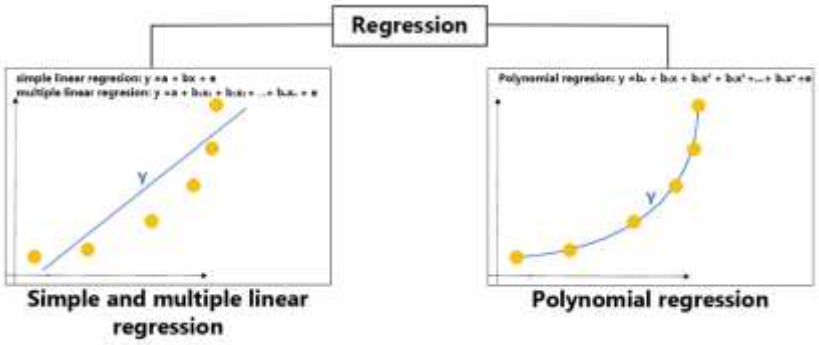


Figure 2-1 : simple, multiple and polynomial regression

2.2.2 Polynomial regression

Polynomial regression is a type of regression analysis in which the relationship between the independent and dependent variables x and y is not linear. However, is the polynomial degree of n^{th}in x. Polynomial regression equation is derived from linear regression (polynomial regression of degree 1) equation, which is defined as follows:

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_n x^n + e \qquad (11)$$

Here, the predicted/target output is y, $b_0, b_1 \ldots b_n$are the regression coefficients, x is an independent/input variable. In simple terms, if data are not distributed linearly, instead it is of the n^{th}degree of polynomial, we use polynomial regression to obtain the desired output.

2.2.3 LASSO and ridge regression

LASSO as shown in figure 2-2 (least absolute shrinkage and selection operator) and Ridge regression in figure 2-3 are well-known as powerful techniques that are commonly used for building learning models in presence of a large number of features, due to their ability to prevent over-fitting and reduce model complexity. The LASSO regression model uses L1 regularization technique that uses shrinkage, which penalizes absolute value of magnitude of coefficients (L1 penalty). As a result, it appears to render coefficients to absolute zero. Thus, the goal of LASSO regression is to identify the subset of predictors that minimizes the prediction error for a quantitative response variable. Ridge regression, on the other hand, employs L2 regularization, which is the "squared magnitude of coefficients" (L2 penalty). As a result, ridge regression forces the weights to be small

while never setting the coefficient value to zero, and does a non-sparse solution. Overall, LASSO regression can be used to obtain a subset of predictors by removing less important features, whereas ridge regression is useful when a data set has multicollinearity, which refers to predictors that are correlated with other predictors.

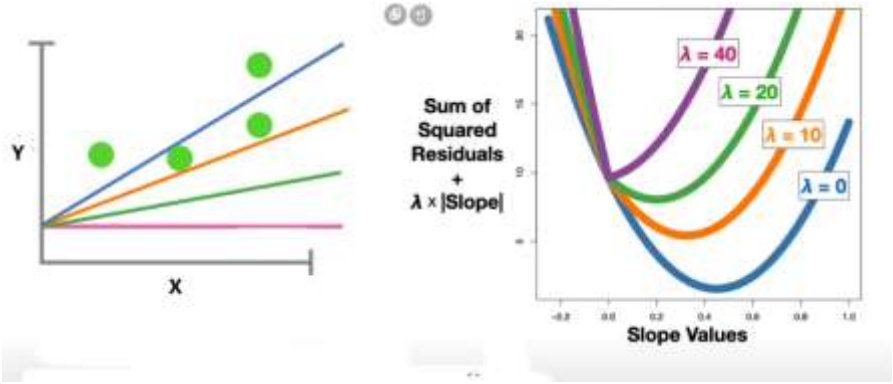


Figure 2-2 : Lasso regression

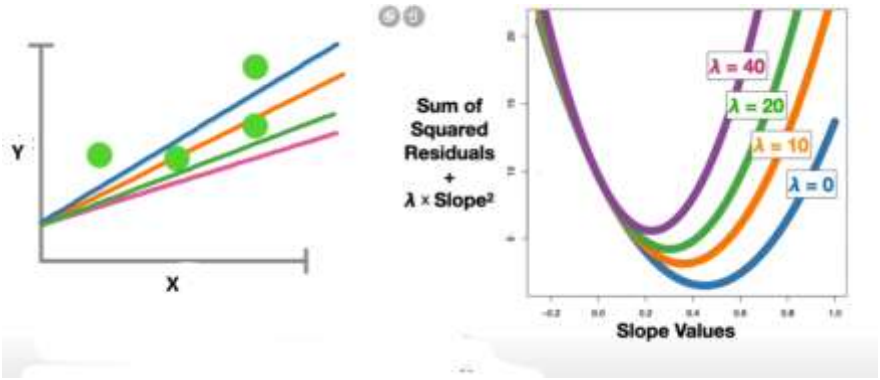


Figure 2-3 : Ridge regression

2.2.4 Logistic regression

The technique of logistic regression is used to investigate the relationship between one or more predictor variables and a response variable. It functions similarly to linear regression, but with a binomial response variable. i.e., Logistic regression is used to obtain odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial. The result is the impact of each variable on the odds ratio of the observed event of interest. The main advantage is to avoid confounding effects by analyzing the association of all variables together [141].

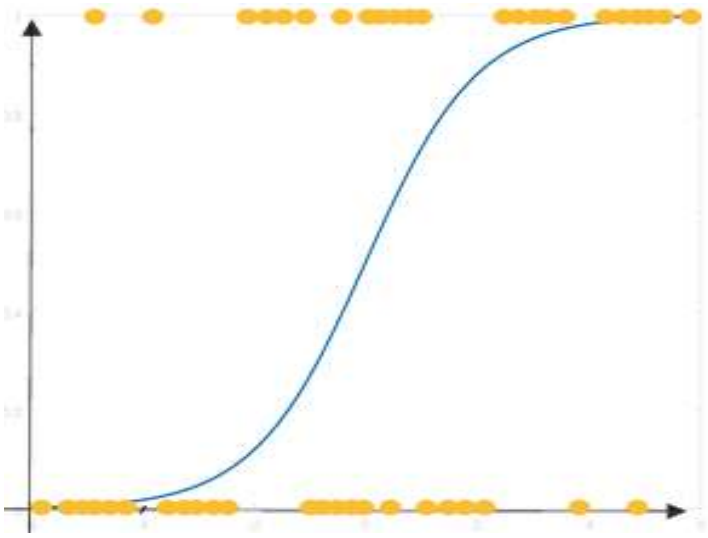


Figure 2-4 : Logistic regression

2.2.5 Quantile regression

The concept of quantile regression, as introduced by Koenker and Bassett (1978), is an extension of classical least squares estimation of conditional mean models to the estimation of an ensemble of models for conditional quantile functions. The central special case is the median regression estimator, which minimizes the sum of absolute errors. The remaining conditional quantile functions are estimated by minimizing an asymmetrically weighted sum of absolute errors. The ensemble of estimated conditional quantile functions provides a much more comprehensive view of the effect of covariates on the location, scale, and shape of the response variable distribution. The quantiles, or percentiles, or fractiles, refer to the general case of dividing a dataset into parts. Quantile regression seeks to extend these ideas to the estimation of conditional quantile functions, i.e. models in which the quantiles of the conditional distribution of the response variable are expressed as functions of observed covariates. To estimate the conditional median function, the median estimator minimizes the symmetrically weighted sum of absolute errors and other conditional quantile functions are estimated by minimizing an asymmetrically weighted sum of absolute errors, where the weights are functions of the quantile of interest. This makes quantile regression resistant to outliers. [142]

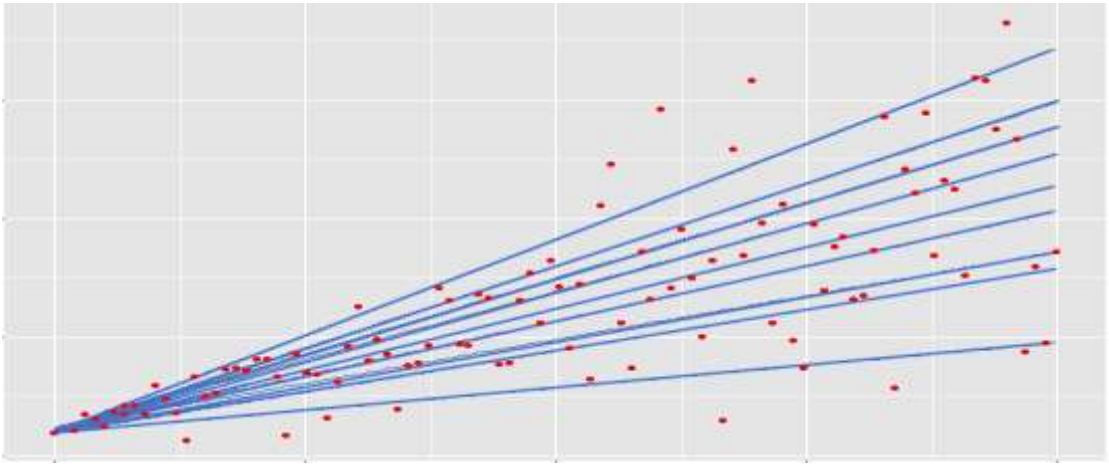


Figure 2-5 : Quantile regression

2.2.6 Bayesian linear regression

Bayesian statistics and modeling are approaches to data analysis based on Bayes' theorem, in which existing knowledge about parameters in a statistical model is updated with information from observed data. [143]. Bayesian linear regression provides a useful mechanism for dealing with insufficient or poorly distributed data. It is possible to apply a prior to the coefficients and the noise so that the priors can take over in the absence of data. More importantly, you can ask Bayesian linear regression which parts of it fit the data well and which parts are uncertain (perhaps based entirely on the priors) [144].

2.2.7 Principal components regression

Kendall (1957) proposed the idea of using principal components in regression in his book on Multivariate Analysis, as did Hotelling (1957) in an article published the same year, and Jeffers (1957) provided a well-known example (1967). These authors intended to use principal components to replace the original regressor variables with their principal components [145]. A large number of explanatory variables in a regression model are reduced to a small number of principal components using principal components regression (PCR). [146]. This method produces informative directions in the factor space, but they may not correspond to the shape of the predicted surface. [147]

2.2.8 Partial least squares regression

PLSR has proven to be a very versatile method for analyzing multivariate data. It is a supervised method developed to address the problem of making accurate predictions in multivariate problems. PLSR has no variable selection implementation in its original form because the method's focus is on finding the relevant linear subspace of the explanatory variables, not the variables themselves, but a large number of variable selection methods in PLSR have been proposed. [148]

2.2.9 Elastic Net regression

The elastic net (ENET) is a lasso extension that is resistant to high correlations among predictors [149]. The ENET was proposed for analyzing high dimensional data to avoid the instability of lasso solution paths when predictors are highly correlated. The ENET employs a hybrid of the lasso and ridge regression penalties [150]. i.e., it is a type of regularized linear regression optimization that bridges the gap between ridge regression and the lasso.

2.3 Linear and non-linear regression models

2.3.1 Linear regression models

By supposing that data are available on p+1 variables(y,x_1,x_2, …. x_p), where variable y is selected as a response based on scientific interest, and the other variables are considered as predictors or covariates. If the data is a sample of size n, it can be represented as {(y_i,x_{i1},x_{i2}, …. , x_{ip}), i=1,2,…,n}.

A linear regression model is the most basic regression model, in which the response and predictors are assumed to have a linear relationship. The general linear regression model is written as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i \qquad i = 1,2,\dots,n, \quad (12)$$

Or

$$E(y_i) = x_i^T \beta, \quad (13)$$

where $y_i$is the response for individual i, β_j's are unidentified parameters, x_{ij} is the j-th predictor for individual i, x_i= $(1, x_{i1}, \dots, x_{ip})^T$ is a set of all predictors, $\beta = (\beta_0, \dots \beta_p)^T$ is a vector of regression parameters, and the ε_i's are random errors with mean zeros with i = 1, 2, …,n. The response and predictors are assumed to have a linear relationship in linear model, with the unexplained variation accounted by the random error ε_i. [151]

Linear regression models range from first-order models in predictor variables to more complex models [152] are widely used in practice because they are simple and easy to interpret, even though they may not accurately represent the true relationship between the response and covariates.

After assuming a linear model for the variables, the next step is to estimate the unknown parameters and make statistical inferences using the observed data.

2.3.2 Nonlinear regression models

Nonlinear regression models (NLR) are similar to linear regression models in that they have the same form:$Y_i = f(X_i, \gamma) + \varepsilon_i$

An observation Y_i is the sum of a mean response $f(X_i, \gamma)$, given by the nonlinear response function $f(X_i, \gamma)$and the error term ε_i. The error terms are typically assumed to have expectation zero, constant variance, and to be uncorrelated, just as for linear regression models. Often, a normal error model is invoked assuming that the error terms are independent normal random variables with constant variance. The parameter vector in the response function f (X, γ) is denoted by γ rather than β in the linear model. This emphasizes that the response function is nonlinear in the parameters. A difference between linear and nonlinear regression models is that the number of regression parameters is not necessarily directly related to the number of X variables in the model. In linear regression models, if there are p – 1, X variables in the model, then there are p regression coefficients in the model. If the number of X variables in the nonlinear regression model is denoted by q, and we continue to denote the number of regression parameters in the response function by p. The general form of a nonlinear regression model is expressed as:

$$Y_i = f(X_i, \gamma) + \varepsilon_i \qquad (14)$$

Like in linear regression models, estimation of parameters of a nonlinear regression model can be carried out by the method of least squares or the method of maximum likelihood. Like in linear regression, both methods of estimation yield the same parameter estimates when the error terms are independent normal with constant variance. Unlike linear regression, it is usually not possible to find analytical expressions for the least squares and maximum likelihood estimators for nonlinear regression models. Instead, numerical search procedures are used with both estimation procedures. For example, The Gauss-Newton method, a.k.a. as the linearization method, uses a Taylor series expansion to approximate the nonlinear regression model with linear terms and then employs ordinary least squares to estimate the parameters. Iteration of these steps generally leads to a solution to the nonlinear regression problem [152].

despite of lack of knowledge on how to construct a DL model that could be applied to solve nonlinear regression problems, some tries are highlighted below.

Table 2-1: Deep learning research into non-linear problem solving

Title	**Description**	**References**
Wind Speed Prediction Model Using LSTM and 1D-CNN	This paper describes a prediction method for wind speed using a neural network and an investigation of the structure of the network. Generally, wind speed is observed as time series data, and the current wind speed is related to the past wind speed. Therefore, a prediction model was proposed using long short-term memory (LSTM) and a one-dimensional convolutional neural network (1D-CNN) in order to consider the past information for prediction. The prediction results of these networks and a fully connected neural network are compared for evaluation. The prediction accuracy and time delay are found to be improved by using LSTM and the 1D-CNN.	[153]
	Cardiovascular diseases are considered the number one cause of death across the globe which can be primarily identified by the abnormal heart rhythms of the patients. By generating	[154]

One-Dimensional CNN Approach for ECG Arrhythmia Analysis in Fog-Cloud Environments	electrocardiogram (ECG) signals, wearable Internet of Things (IoT) devices can consistently track the patient's heart rhythms. Although Cloud-based approaches for ECG analysis can achieve some levels of accuracy, they still have some limitations, such as high latency. Conversely, the Fog computing infrastructure is more powerful than edge devices but less capable than Cloud computing for executing compositionally intensive data analytic software. The Fog infrastructure can consist of Fog-based gateways directly connected with the wearable devices to offer many advanced benefits, including low latency and high quality of services. To address these issues, a modular one-dimensional convolution neural network (1D-CNN) approach is proposed in this work.	
Deep Learning for Event-Driven Stock Prediction	deep learning method was proposed for eventdriven stock market prediction. First, events are extracted from news text, and represented as dense vectors, trained using a novel neural tensor network. Second, a deep convolutional neural network is used to model both short-term and long-term influences of events on stock price movements	[155]
STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL	Stock market or equity market have a profound impact in today's economy. A rise or fall in the share price has an important role in determining the investor's gain. The existing forecasting methods make use of both linear and non-linear algorithms),but they focus on predicting the stock index movement or price forecasting for a single company using the daily closing price. a deep learning based formalization was proposed for stock price prediction. It is seen that deep neural network architectures are capable of capturing hidden dynamics and are able to make predictions	[156]

2.4 Conclusion

Overall, we drawn a big picture on regression as one of the most widely used techniques for analyzing multifactor data. we briefly review both linear and nonlinear regression models focusing on the nonlinear ones with basic principles necessary to apply regression models in a wide variety of application environments.

CHAPTER III

Deep learning for nonlinear real-world problem

3.1 Introduction

Functions are common conceptions in sciences and engineering that quantify the dependence or interaction between one variable and the others. Functions are classified as linear functions and nonlinear functions from the superposition principle prospective. Linear functions are analytical, easy to analyze mathematically, and satisfy the superposition principle, while nonlinear functions are complicated and even nonanalytical. Neural networks and its various models can provide a robust solution to our problems, but there is a lack of knowledge on how to construct a model that could be applied to solve the regression problems. Indeed, one of the most crucial issues is how to solve the problem of how to design a reasonable model for continuous input and output, targeting the nonlinear regression problems such as wind speed prediction, ECG analysis, localization in smart cars, stock prediction.

in this chapter, We will seek to find a DL approach that can handle non-linear problems, then utilize it to address a real world problem which is the mechanical degradation of greenhouse LDPE films, The model will forecast both the full stress-strain curves and the material lifetime under real-world conditions.

The convolutional neural network was almost certainly the model to choose. It's typically employed in the extraction of picture features scenario. However, the same filters can exploit temporal correlation in raw data.

3.2 Case of study

Low-density polyethylene (LDPE) is a widely utilized substance in polymers, and it is commonly employed as agricultural greenhouse covers. Its lightness and transparency, in particular, have contributed to its success.

Co-extrusion, which has emerged as a new technology in recent years, allows the combination of multiple layers of polymers. Each layer has its unique characteristics, such as increased strength, improved appearance, cold resistance, improved welding or sealing, and barrier properties [157].

Despite developments in film formulation, most LDPE films in use today are expected to last for a relatively short amount of time, ranging from one to five cultivating seasons. Typically, film producers promise a maximum functional lifetime of four seasons (i.e., four winters and three summers). Even this predicted lifetime is heavily influenced by the actual environmental conditions that the film will be subjected to throughout its use.

Polyethylene (PE) environmental degradation is a complex process in which numerous degradation mechanisms work together to completely destroy the material. Degradation is a complex non-linear time-dependent process that affects numerous properties of the

material that are related to its functional features directly or indirectly. A material does not match its functional requirements and is easily prone to mechanical breakdown in its ultimate stage of degradation [158].

The essential degradation in the mechanical characteristics (modulus of elasticity, fracture stress and elongation at break) has been evaluated in our previous work [157].

The stress-strain curve provides the fundamental information required to comprehend and forecast the mechanical degradation of greenhouse LDPE films. Indeed, the stress–strain curve of greenhouse LDPE films derived from tensile testing provides information about the material's elastic characteristics, the nature and degree of its plastic deformation, as well as its yield and tensile strength. Furthermore, to gain a thorough understanding of the effect of ageing time on film properties (e.g., mechanical performance) under various ageing conditions, plot the film property value after exposure to ageing conditions normalized with the corresponding property value of the unaged material against ageing time. This will also provide information about the service duration limit based on the criterion of 50% loss in the original property [159]. Furthermore, we require a model for analyzing their final lifespan, that is, a model for predicting the material lifetime under use conditions. This will assist manufacturers in improving their products and will be beneficial to engineering designers.

However, in order to enhance the lifetime of greenhouse LDPE films, it is necessary to forecast stress-strain curves for a certain ageing time, despite the fact that experimentally examining LDPE mechanical qualities can be costly and time intensive.

Deep neural networks have been demonstrated to outperform typical statistical techniques and human performance in making predictions about outcomes in a variety of fields, including computer vision, self-driving automobiles, speech recognition, natural language processing, and pattern identification [160].

Deep neural networks' applicability in material science are relatively limited. It is mostly due to the challenges of compiling large datasets in material science [161]. Indeed, few works in this area have been published, including material classification [162], defect classification [163], microstructure reconstruction [164], and microstructure image recognition [165].

The main goal of this work is to combine a recent deep learning-based algorithm and a traditional classifier SVM (Support Vector Machine) to forecast both the full stress-strain curves and the material lifetime under real-world conditions. To the best of the researchers’ knowledge, no previous machine learning-based methodology has been employed to forecast stress-strain curves of LPDE films.

In addition, our goals in this work are to (1) demonstrate the application of deep convolutional neural networks (CNN) in the field of polymer characterization, (2) capture

the nonlinear relationship between the ageing of polymeric greenhouse covers and their stress-strain curves, and (3) demonstrate the feasibility of training CNN models with small datasets.

3.3 Materials and methods

3.3.1 Materials

Agrofilm SA (Setif-Algeria) manufactures the LDPE film utilized in this study utilizing a three-layer co-extrusion technology that is employed in a variety of industrial applications. The three-layered coextruded film has a total thickness of 180 µm with proportions of 1/4, 1/2, and 1/4 over the total film thickness. Without a stabilizer, the LDPE melt flow index is 0.33 g/10 min, while with a stabilizer, it is 10 g/10 min. Before extrusion, the density of LDPE is 0.923 g/cm3. The film's original hue is a milky yellow. The real composition of the film is kept confidential by the supplier.

3.3.2 Dataset

In this study, we used the dataset from our previous work [166], The mechanical behavior of unaged/virgin, naturally and artificially aged films was evaluated in different combined conditions of temperatures (10,25,40, and 50 C), water, and UV-A radiations by conducting the tensile test, the tensile tests were performed according to ISO 527-3. The weight on the specimen was applied in a direction parallel to the average molecular orientation acquired after film processing. For each film, the engineering stress has been recorded as a function of engineering strain. 120, 550, 1010, 2040, and 4000 hours were regarded as ageing durations. In other words, a 4072-point dataset of the five stress-strain curves in the temperature 10° without the use of water or UV radiation was obtained.

3.3.3 The Proposed model

A deep Convolutional Neural Network (CNN) and a Support Vector Machine (SVM) are merged in this part to present a hybrid model, called hybrid SVM-CNN, for predicting both the whole stress-strain curves and the material lifetime under the conditions of use. The prediction of a stress-strain curve consists of two problems: one is a classification problem that separates each point on the curve as a function of strain values, i.e., material fracture classification, and the other is a regression problem that predicts stress values.

Figure 3-1 shows the hybrid SVM-CNN model, which is composed of two submodels: the SVM submodel and the CNN submodel. The two submodels were trained independently before being put to the test together.

In the proposed model, SVM serves as classifier, whereas CNN serves as a feature extractor.

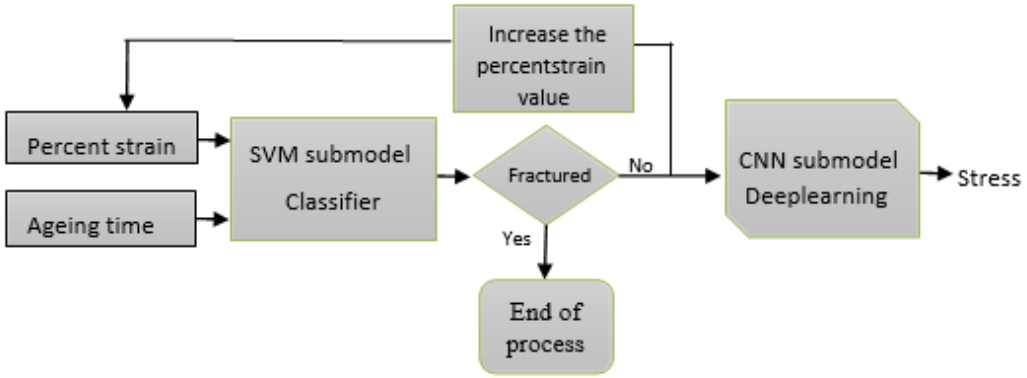


Figure 3-1: The proposed hybrid SVM-CNN model architecture.

The implementation code was written in Python 3.8 on a PC (Intel(R) Core(TM) i5-6200U CPU 2.4GHz, 8 Gbyte RAM) while for the CNN submodel we have used Keras deep learning framework with a TensorFlow backend [167], and Scikit-learn library [168] was used for the SVM submodel.

A) The CNN submodel

The topology of a deep convolutional neural network and the hyperparameters that determine the learning process have a substantial influence on its performance. Deep CNN is most commonly utilized in the extraction of picture features. However, there is a scarcity of information about how to build a deep CNN model that can be used to handle regression problems. One of the most important challenges is how to tackle the difficulty of designing a suitable model for continuous input and output, which is a nonlinear regression problem. On the one hand, applying deep CNN to tiny datasets remains difficult. Despite the fact that our dataset has 4072 coordinate points, only five stress-strain curves are represented. To increase prediction result, on the other hand, the model must implement efficient feature extraction. After experimenting with numerous structures, we discovered that the needed structure has its own distinct properties for the prediction of stress-strain curves. The final deep learning architecture of the 1D CNN submodel is shown in Figure 3-2.

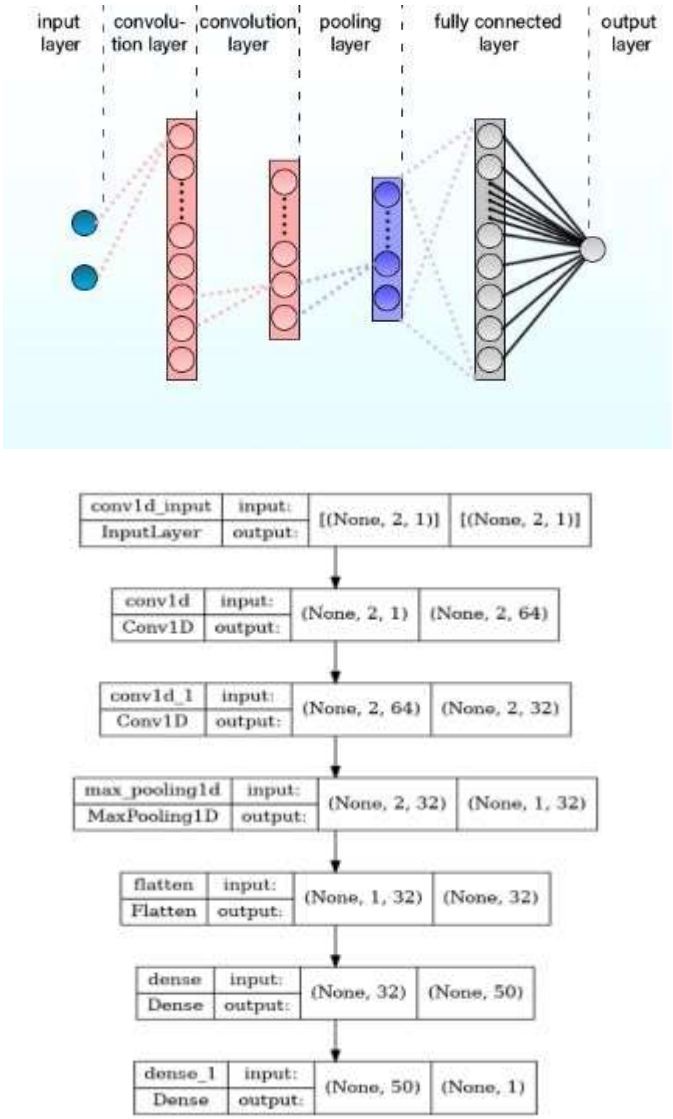


Figure 3-2: 1D CNN submodel architecture.

Figures 3-2 shows the suggested 1D CNN submodel, which includes an input layer, two convolutional layers, a pooling layer, a fully connected layer, and an output layer. This neural network's two convolutional layers can be viewed as feature extractors, while the next layer combines semantically similar features into one, and the final layer maps the extracted features into final output. This 1D CNN submodel estimates the relevant stress using the percent strain and ageing time as inputs. For an optimally built 1D CNN, the hyperparameters filter sizes, stride, padding, pool size, and number of filters were optimized by experience. Table 3-1 shows the selected hyperparameter values.

Table 3-1: A list of hyperparameters in the 1D CNN submodel.

	Hyperparameters
Convolution layer	Kernel size = 1, filters = 64, strides = 1, padding = valid, activation = relu
Convolution layer	Kernel size = 1, filters = 32, strides = 1, padding = valid, activation = relu
Pooling layer	Pooling method: maxpooling, pool size: 2, stride = 2, padding = valid
Fully connected layer	Neurons = 50, activation function = relu

The model was trained with the log hyperbolic cosine (log-cosh) loss function. In fact, log-cosh approach has been widely used in regression-based problem for smoothing the curve [39]. The log-cosh loss formula is given as follows:

$$L(X) = \frac{1}{N}\sum_{i=1}^{N} \log(\cosh(y_i - f_i(X))) \qquad (15)$$

Where $f_i(x)$ is the predicted values, y_i is the true values, $coshcosh\ (x) = \frac{e^x + e^{-x}}{2}$ and N the number of samples.

Three well-known activation functions have been tested successively, and they are hyperbolic tangent (tanh), sigmoid and rectified linear unit (ReLU). We have used the same activation function in all layers, except the output layer where we have used linear activation function. The performances are displayed in figure 3-3 and table 3-2. We can see that the model converges much faster using ReLU as activation function. A ReLU of x is simply the maximal value of 0 and x, meaning that it will return a 0 if the input is negative or the raw input otherwise. In symbols : f(x) = max (x, 0).

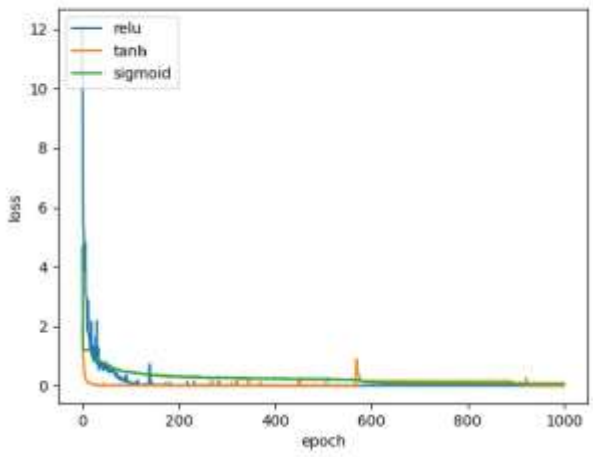


Figure 3-3: Comparison of loss curves during the training progress

Table 3-2: Effect of activation functions on model convergence

	Log-cosherror		
Epochs	**Tanh**	**Sigmoid**	**ReLU**
1	2,42	3,4792	11,23
2	1,0521	1,204	5,8434
4	0,4482	1,1971	5,2444
8	0,2173	1,1816	3,5787
16	0,0849	0,8108	1,1814
32	0,0305	0,7495	0,9528
64	0,0496	0,745	0,2367
128	0,002	0,5894	0,0167
256	0,0019	0,2051	0,0092
512	0,0144	0,1062	0,0017
1024	0,147	0,0195	0,0009691

To provide a more stable error gradient, we used batch gradient descent for training [169]. This resulted in more stable convergence. The error is estimated for each example in the training dataset in batch gradient descent, but the model is updated only after all training examples have been evaluated. This process is repeated until convergence is achieved. An epoch is a hyperparameter that represents a single pass across the whole training dataset.

Four of the five available stress-strain curves (80 % of the dataset) were used to train the 1D CNN submodel. The test set is the data which was never used by the model during training. However, the fifth curve (20 % of the dataset) was used to test the submodel. In other words, the trained 1D CNN was requested to predict the fifth curve and the output of the network is compared with the available experimental data to check the validity of the submodel.
The Adam optimizer [170] was used to update the parameters during backpropagation. The prediction model was trained with up to 1750 epochs. Early stopping was applied using Callback functions to prevent overfitting (i.e., network shows high accuracy during training but less accuracy when new data is given during testing). At the end of the training process, the training error value was 0.0001302.

We have evaluated the quality of the prediction model using the three metrics of mean squared error (MSE), root-mean squared error (RMSE) and mean absolute error (MAE). These metrics are widely used to evaluate the error rates for prediction using regression models. The results obtained are shown in Table 3-3. All the evaluation metrics were calculated using Python package keras-metrics [167]. Let y_i^* represent the values of variables for n prediction samples of stress, and let y_i represent the observed values. Eqs. 16 to 18 then represent the MSE, RMSE and MAE, respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - y_i^*)^2 \quad (16)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(y_i - y_i^*\right)^2} \quad (17)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y_i^*| \quad (18)$$

Table 3-3: Evaluation metrics for 1D CNN submodel

Activation functions in hidden layers	Evaluation metrics		
	MSE	RMSE	MAE
tanh	1.46	1.21	0.69
Sigmoid	1.51	1.23	1.4
ReLU	0.0002	0.0155	0.02

The trained network was asked to simulate a stress-strain curve of ageing time1010 hours for each activation function, and the results were compared to the existing experimental data to ensure the submodel's validity. It's worth noting that this curve was not seen by the network during the training phase.

As shown in figure 3-4, the most accurate prediction is obtained using the ReLU activation function. The curves exactly coincide with each other, but the predict curve exceeds the fracture point, even for the training curves while the simulation is stopped manually. For this reason, we have decided to add another submodel to predict the fracture point using SVM. Once the fracture point is predicted, the simulation will stop automatically without any human action. This hybrid model automatically extracts features from raw data and generates the predictions. The data is first passed via the SVM submodel, where the percent strain value is categorized into the appropriate class based on the ageing time (i.e., the material is fractured or not). The process finishes if the material is broken, otherwise, data is passed through the CNN submodel for predicting the stress value, the procedure is then repeated from the beginning, with only the percent strain value being incremented.

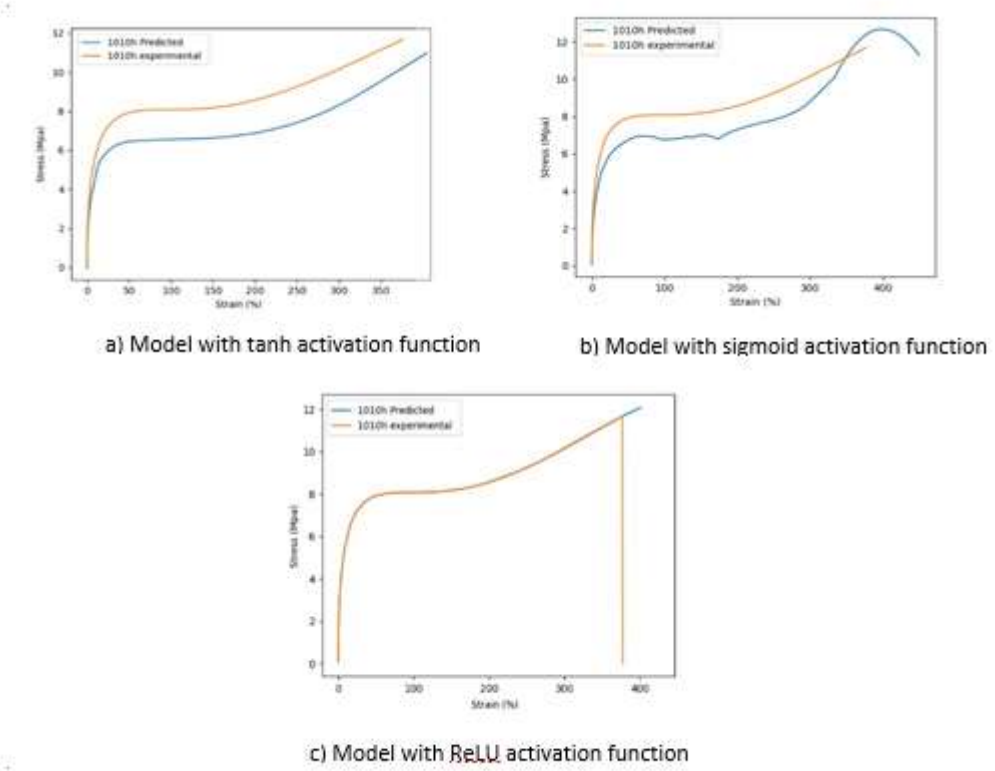


Figure 3-4: Graphical comparison between predicted and unseen experimental stress-strain curve

B) The SVM submodel

Figure 3-5 shows the fracture point or fracture stress, which is the final point in the stress-strain curve. To estimate the fracture stress of aging films, a support vector machine submodel was suggested. Each point on the stress-strain curve was treated as individual binary classification problem. Therefore, the first class on the stress-strain curve was designated '1', which corresponded to strain values ranging from zero to breaking strain. Exceeding this value, the second class was labeled '0'.

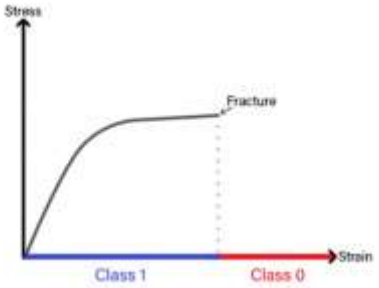


Figure 3-5: The two classes of stress-strain curve

The inputs to the SVM submodel are percent strain and ageing time as shown in figure 3-6, the target output is a binary value 1 or 0 corresponding to not fractured or fractured. The SVM submodel was trained using a set of 4072 coordinate points of the stress-strain curves, automatically labeled with a script.

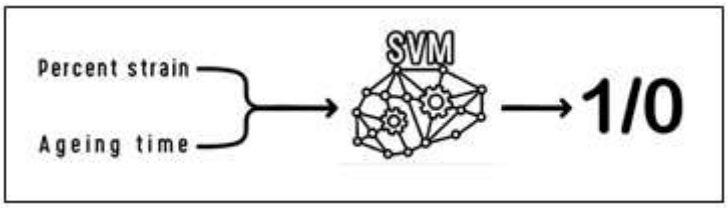


Figure 3-6: The SVM submodel.

To evaluate the SVM submodel we have used the most popular metrics [171], which are accuracy and AUC. Accuracy is basically the proportion of correct results among all predictions. If y_i^* is the predicted value of the i^{th} sample and y_i is the corresponding true value, then the fraction of correct predictions over N samples is defined as:

$$\text{accuracy}(y, y^*) = \frac{1}{N}\sum_{i=0}^{N-1} 1(y_i^* = y_i) \qquad (19)$$

Where 1(x) is the indicator function.

Area under the curve (AUC) is a combined measure of sensitivity and specificity. Assume we are given dataset D = $\{(x_i, y_i)\}$ i=1,...,N with N examples, class labels $y_i \in \{-1,+1\}$, and input vectors x_i, the number of positive examples (i.e., where $y_i = +1$) is N^{pos}, and the number of negative examples is $N^{neg} = N - N^{pos}$.

The AUC of a predictor f is defined as [172]:

$$AUC(f, D) = \frac{1}{N}\sum_{j=1}^{N^{neg}} \sum_{k=1}^{N^{pos}} 1[f(x_j) < f(x_k)] \qquad (20)$$

Where 1(a) is the indicator function, 1(a) = 1 if a is true and 0 otherwise.

The AUC and the accuracy were calculated using Python 3.8 with Sklearn.metrics [168]. We have obtained 0.969 as AUC and 0.98 as accuracy, which means that the submodel performs well ,i.e., the material is fractured or not is well predicted.

3.4 Results and discussion

Our results for the hybrid SVM-CNN model are shown in Figure 3-7. All figures in this work were generated using Matplotlib [172]. For each ageing time (120, 550, 1010, 2040, and 4000 h), the model was requested to predict the stress-strain curve and the output of the model was compared with the available experimental data. It can be seen that the curves exactly coincide with each other. One can notice that the model demonstrated strong ability to simulate the trend of the curves. The mechanical properties such as yield

strength and fracture stress and strain are predicted with accuracy that is very close to the maximum of 100%. In addition, from the predicted stress-strain curves, one can calculate other mechanical properties such as tensile strength, strain at break, percent elongation, young modulus, toughness and the ratio of tensile strength to young's modulus.

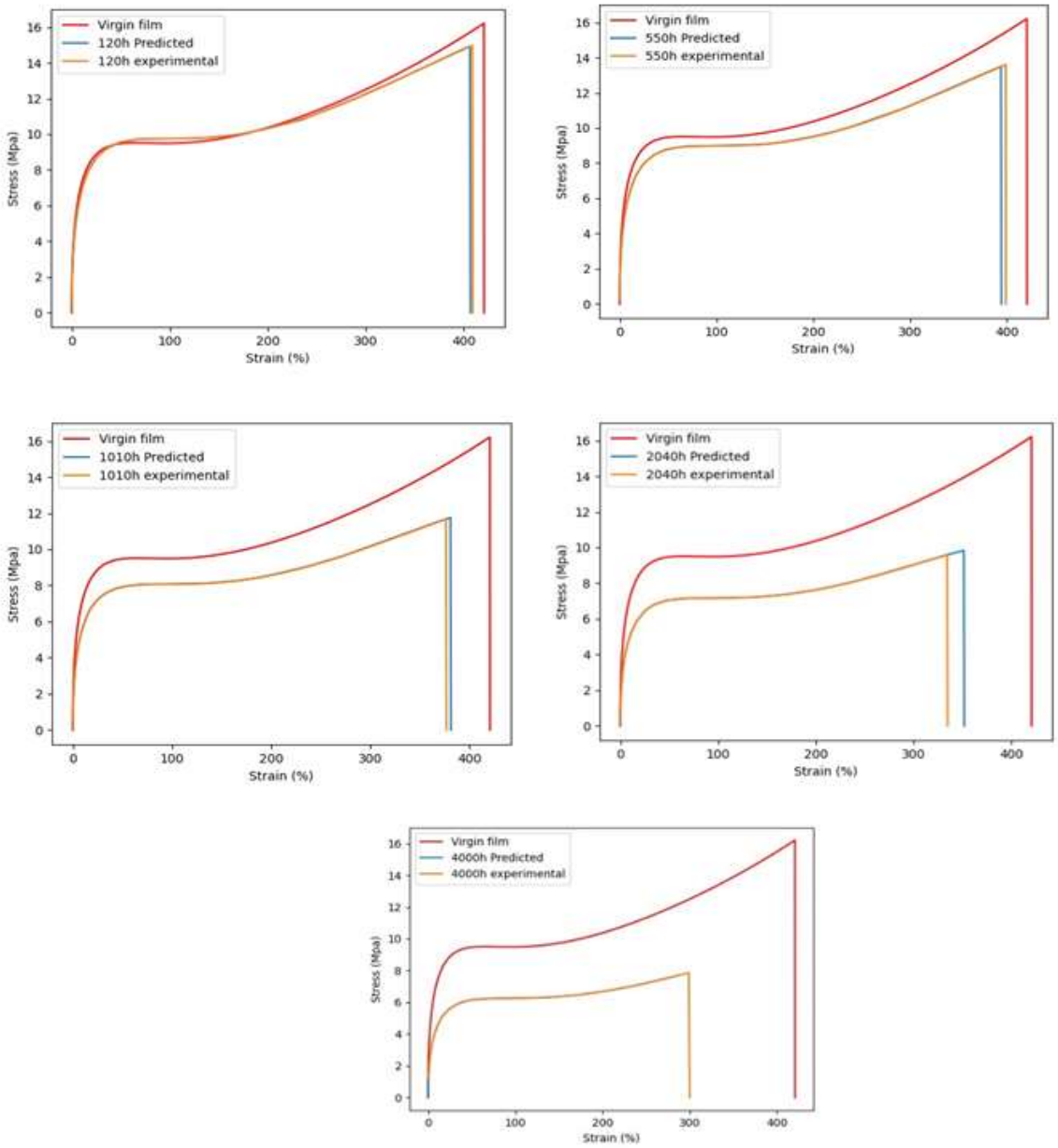


Figure 3-7: Graphical comparison between experimental and predicted stress-strain curves for different ageing time.

Designing and tuning deep neural networks require large efforts from developers, especially with a small dataset like ours. Indeed, unlike typical CNN, we have used two convolutional layers successively, this makes the 1D CNN submodel perform well. Our dataset is composed by five ageing periods up to 4000 h, although the hybrid SVM-CNN model can exceed these periods and predicts the stress-strain curves for any given ageing time. For example, as shown in figure 3-8, the model can easily predict the curves for

5000, 6000 and 9000 h. This revealed the powerful ability of learning high-level feature representations of deep CNN.

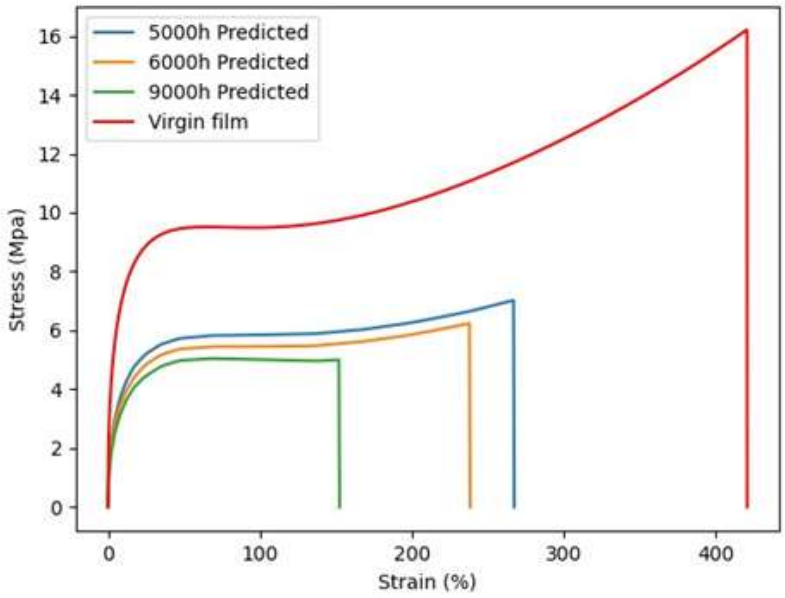


Figure 3-8: Predicted stress-strain curves for different ageing time.

In addition, the hybrid SVM-CNN model could eventually be used to predict the maximum degradation time under the condition of use. Indeed, using python loop, we have increased the ageing time until 18000 h (2 years). The model was requested to predict the stress for each value of ageing time. The maximum degradation time (ageing time) was 14325 h, from this value the model does not predict the stress and displays "material fractured". This means that the material is completely destroyed. On the other hand, the predicted curve represent stress-strain curve of the LDPE film during degradation at selected period. Therefore, one can plot, on the same graph, both the predicted curve and the corresponding stress-strain curve for the unaged material, as we have done. This will also provide information about the limit of use service time according to the criterion of the 50% loss in the original property [159].

In the following, we will compare the model performance to the most commonly used type of neural networks for supervised learning.

3.4.1 Comparison with MLP model

The multilayer perceptron (MLP) is a Feedforword Neural Network (FNN) with one or more layers of neurons between the input and output layers [174]. The architecture of an MLP can be represented as an acyclic graph, so that neurons in any layer are connected to all neurons in the next layer and no feedback between layers.

Many studies [175-177] have shown that MLP with one hidden layer can approximate arbitrarily well any continuous function of several real variables. Therefore, our MLP model developed to predict the stress-strain curves has one hidden layer between the input and output layers. The output neuron has a linear activation function, while the ReLU activation function has been chosen for neurons in the hidden layer. The model inputs are the percent strain and ageing time, while the output is the stress. We have trained several MLP networks. In fact, we have increased the number of neurons in the hidden layer up to 100. We found that over 80 hidden neurons, network becomes too complex and generalization ability becomes very poor. The MLP networks were trained with batch gradient descent. The Adam optimizer was used to update the MLP weights during backpropagation, while the log-cosh was used as loss function. Each network was trained with 80 % of dataset, i.e., four stress-strain curves and then evaluated with the stress-strain curve of ageing time1010 hours (20% of the dataset). The performances of the best networks are shown in table 3-4. The outputs of each MLP network after simulation process are shown in figure 3-9. The MLP networks were implemented using Keras [167].

Table 3-4: MLP Networks having best performances

	log-cosh error	MAE	MSE	RMSE
MLP 30 neurons	0.63	1.07	2.11	1.45
MLP 50 neurons	0.62	1.11	2.06	1.44
MLP 80 neurons	0.50	0.92	1.65	1.29

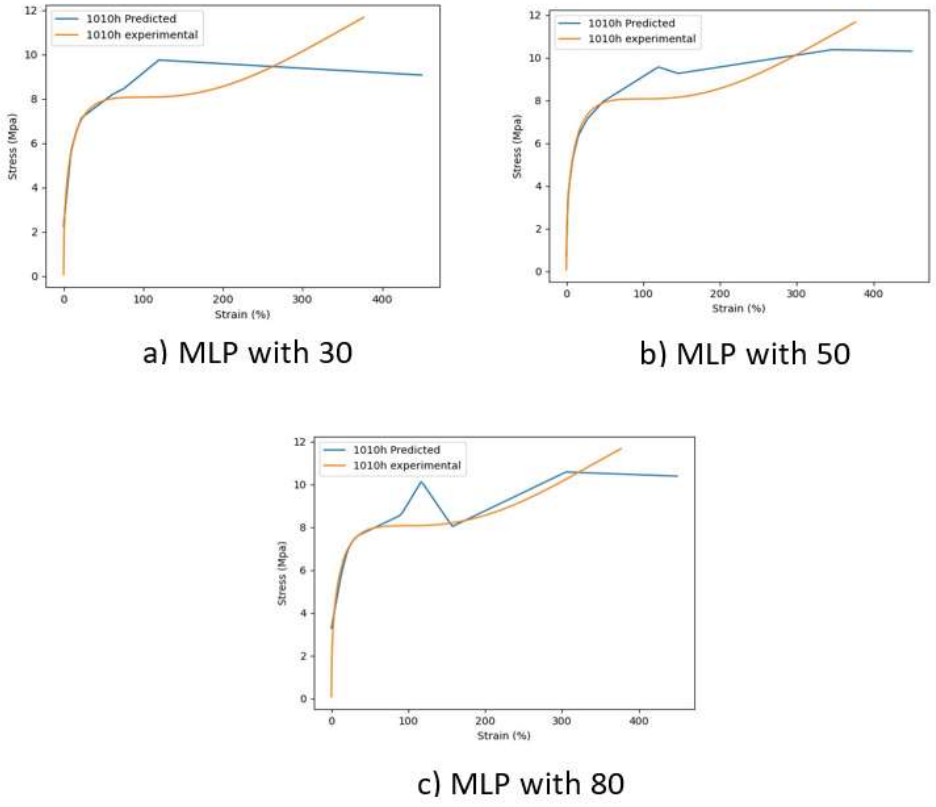


Figure 3-9: Graphical comparison between MLP-predicted curves

A comparison between the two models (1D CNN and MLP) is done in terms of training error and testing error as illustrated in Table 3-5. From the table, we can see that the best performance has been obtained with the 1D CNN model.

Table 3-5: Comparison of the two metrics for different models

	Training error (*log-cosh*)	Testing error (MSE)
1D CNN	0.0001	0.0002
MLP 80 neurons	0.50	1.65
MLP 50 neurons	0.62	2.06
MLP 30 neurons	0.63	2.11

From a structural point of view, both models (1D CNN and MLP) contain one fully connected layer, apply ReLu function as hidden activation function, use Linear function as the activation function of the output layer and Log-cosh as the loss function, whereas, the results show that the MLP performs worse than the 1D CNN.

As demonstrated in Figure 3-7, the 1D CNN-based method predict very well the target stress-strain curves. 1D CNN performs well on unseen data by efficiently learning from small dataset. The failure of the MLP network, as shown in figure 3-9, results from the inability to extract features from raw data and the low size of dataset.

3.5 Software

After reaching our goal of developing an efficient hybrid model capable of predicting both the entire stress-strain curve and the material lifespan under actual usage circumstances, we must provide the user with an efficient and user-friendly program.

3.5.1 User interaction and design

We understand that using a CLI (command-line interface) is difficult and complicated for most people, thus we chose PySimpleGUI to create a simple and easy-to-use GUI.

PySimpleGUI is a python package that transforms the tkinter, Qt and WxPython GUI frameworks into a simpler interface. And that what we're seeking for a simple interface.

The main window will look like this:

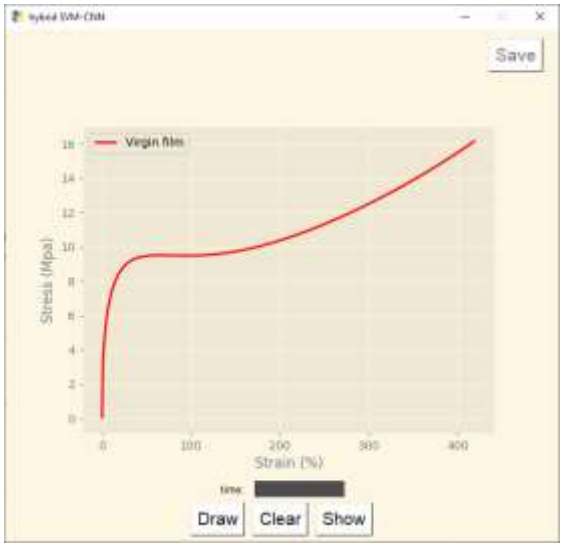


Figure 3-10: main window

We have a canvas that takes up the majority of the window space, with an initial curve of virgin film to compare to the stress strain curves provided by the model. at the bottom We have an input for the ageing time that we want to draw its curve for. We simply need to type in the desired time and click the draw button, and the model will predict and plot the stress strain curve.

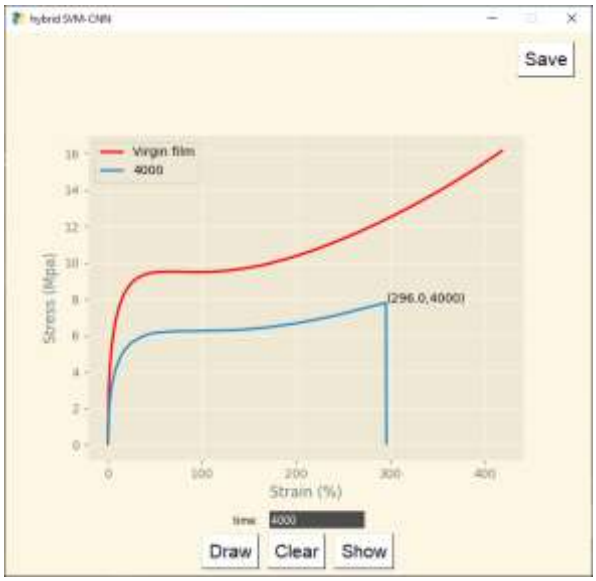


Figure 3-11: plot the stress strain curve

Multiple stress strain curves can also be drawn adjacent to each other:

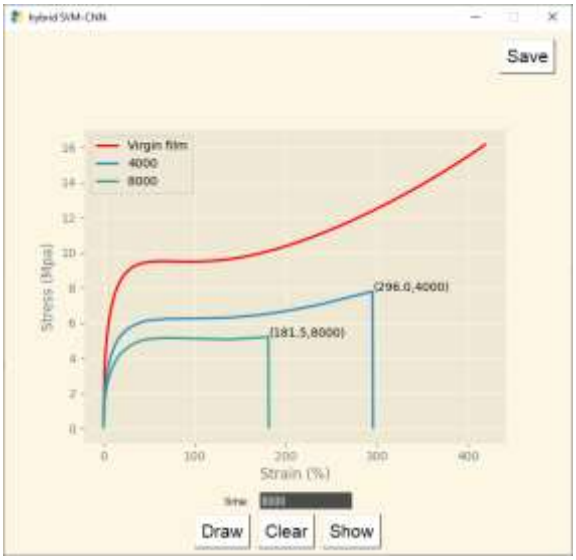


Figure 3-12: multiple stress strain curves adjacent to each other

For a more detailed look at the chart we can click the show button, where we can zoom and move on the chart as well as save it as an image on our disk.

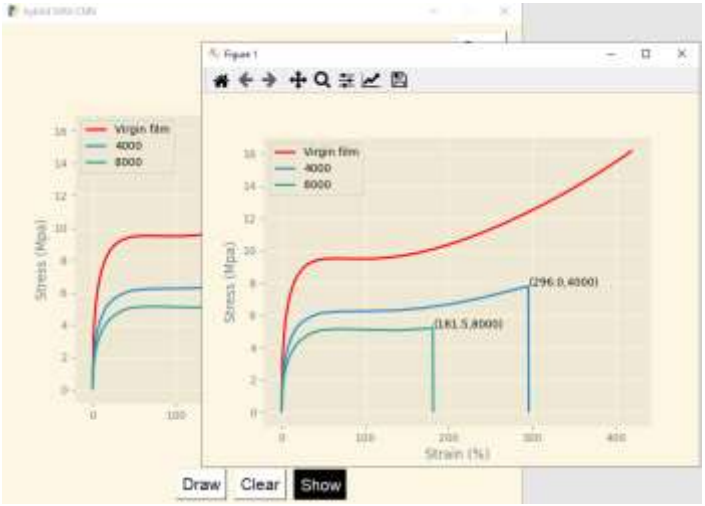


Figure 3-13: detailed vue at the chart

We can also save the model's predicted stress strain curve to an excel file by clicking the save button in the main window.

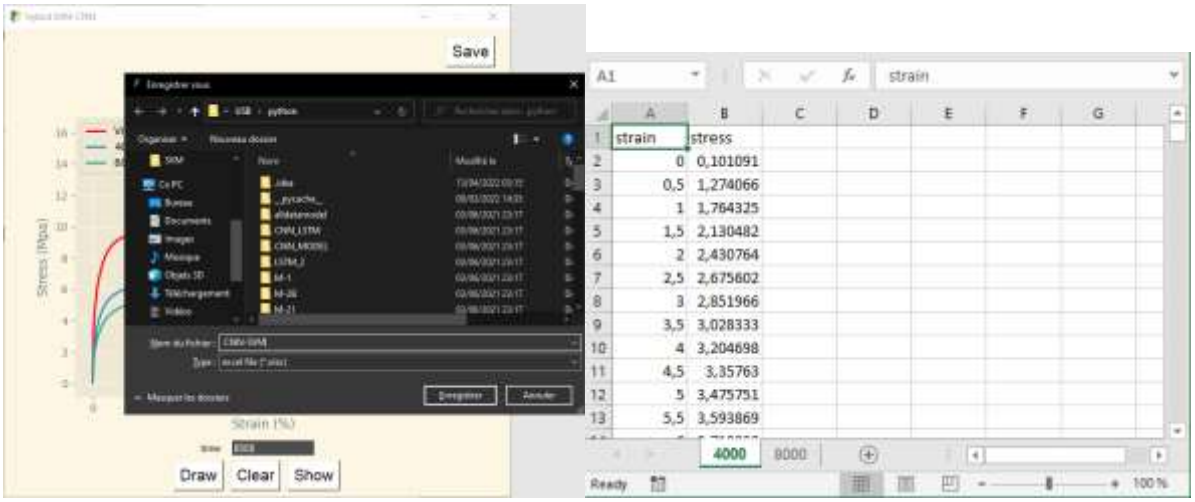


Figure 3-14: saving to excel

We implemented a cache file system to save time and increase efficiency, so that the model doesn't have to predict strain stress curves of ageing times which it has already predicted.

3.6 Conclusion

We have developed and validated a hybrid deep learning model capable of solving a nonlinear problem which was the mechanical performance degradation of greenhouse tri-layer LDPE films. The DL model was successful in both training and validation phases and demonstrate the capability of using 1D CNN for nonlinear regression.

Though CNN (2D or 3D) with big datasets is the optimal solution, 1D CNN with small datasets can be a reasonable choice when big datasets are unavailable in material study. The attractive feature of 1D CNN is its ability to exploit temporal correlation in raw data.

The hybrid SVM-CNN model predicts the stress-strain curves as a function of ageing time. This can save time and cost, and provide easy tools to predict both the mechanical properties of greenhouse coverings and their lifetime under the conditions of use. This will help manufacturers to create materials capable of long-term use and be useful for engineering designers.

This work leads to the conclusion that 1D CNN is a promising model method that can provide a significant solution to nonlinear challenges.

General conclusion

Due to the development of various efficient learning methods and network structures in the late 1980s, neural networks became a popular topic in the fields of Machine Learning and Artificial Intelligence. While neural networks are successfully used in many applications, interest in this topic has waned. Following that, Hinton et al introduced Deep Learning in 2006, which was based on the concept of artificial neural network.

This study on deep learning-based solutions opens up an intriguing direction and showed that DL is a beneficial method for solving non-linear problems Just like linear ones.

Overall, this work on deep learning and non-linear regression by choosing the prediction of lifetime and mechanical performance degradation of Multilayer Greenhouse Polyethylene films as real-world problem in a promising path and can be utilized as a reference guide for future research and implementations in relevant application domains.

Perspective

- We were able to create a hybrid model that can forecast both the lifetime and mechanical performance degradation of LDPE films. However, the model can only forecast stress-strain curves under certain real-world conditions (10° and without the usage of water or UV rays).so our next challenge is to implement a DL model that can predict the stress-strain curves in any conditions.
- And we can go even further and try to solve more nonlinear problems with deferent DL models.

ANNEXE

COMPOSITES | Published: 03 February 2022

A Hybrid Deep Learning Model for Predicting Lifetime and Mechanical Performance Degradation of Multilayer Greenhouse Polyethylene Films

Aid Lahcene, Djenane Mouloud Amine & Dehbi Abdelkader

Polymer Science, Series B **63**, 964–977 (2021) | Cite this article

52 Accesses | Metrics

Abstract

Stress-strain curves of greenhouse LDPE films represent the mechanical properties of materials, such as tensile strength, strain at break, percent elongation, Young modulus and toughness. Stress-strain curves of aged greenhouse LDPE films give a general description to evaluate the degradation of mechanical performance. The aim of this work is to predict both the lifetime and the mechanical performance degradation of the LDPE films. A key element, to handle this kind of difficult and challenging problem, is the use of deep learning techniques that attempts to learn high-level features in data by using structures composed of multiple non-linear transformations. Indeed, based on an experimental dataset, a novel hybrid deep learning model that integrates a deep convolutional neural network and support vector machine is developed, able to predict both the entire stress-strain curves and the material lifetime under the conditions of use. The model performance was validated using several evaluators and compared to a multilayer perceptron neural network. The results revealed that the hybrid deep learning model has high prediction performance.

ISSN 1560-0904, Polymer Science, Series B, 2021, Vol. 63, No. 6, pp. 964–977. © Pleiades Publishing, Ltd., 2021.

COMPOSITES

A Hybrid Deep Learning Model for Predicting Lifetime and Mechanical Performance Degradation of Multilayer Greenhouse Polyethylene Films

Aid Lahcene[a],*, Djenane Mouloud Amine[a], and Dehbi Abdelkader[b],**

[a] *Department of Computer Science, University of Ibn Khaldoun Tiaret, Tiaret, Algeria*

[b] *Engineering Laboratory, Materials Science Faculty, University of Tiaret, Tiaret, 14000 Algeria*

**e-mail: a_lahcen@esi.dz*

***e-mail: abddehbi@gmail.com*

Received April 26, 2021; revised August 23, 2021; accepted September 9, 2021

Abstract—Stress-strain curves of greenhouse LDPE films represent the mechanical properties of materials, such as tensile strength, strain at break, percent elongation, Young modulus and toughness. Stress-strain curves of aged greenhouse LDPE films give a general description to evaluate the degradation of mechanical performance. The aim of this work is to predict both the lifetime and the mechanical performance degradation of the LDPE films. A key element, to handle this kind of difficult and challenging problem, is the use of deep learning techniques that attempts to learn high-level features in data by using structures composed of multiple non-linear transformations. Indeed, based on an experimental dataset, a novel hybrid deep learning model that integrates a deep convolutional neural network and support vector machine is developed, able to predict both the entire stress-strain curves and the material lifetime under the conditions of use. The model performance was validated using several evaluators and compared to a multilayer perceptron neural network. The results revealed that the hybrid deep learning model has high prediction performance.

DOI: 10.1134/S1560090421060117

INTRODUCTION

Low-density polyethylene (LDPE) is one of the most used materials in plasticulture and its utilization as agricultural greenhouse covers is a common application. The main properties that have ensured its success are especially its lightness and transparency. Owing to the co-extrusion, one can combine several layers of polymers together. Each layer has its own qualities; increased strength, improved appearance, resistance to cold, improving the welding or sealing, barrier properties [1]. It is known that despite all the advances in the formulation of the films, most of the LDPE films in use today are expected to last for a rather limited period, between one and five cultivating seasons. The maximum useful lifetime usually advertised by film producers is four seasons (i.e., four winters and three summers). Even this expected lifetime is significantly affected by the actual environmental conditions the film will face during its use.

Polyethylene (PE) environmental degradation is a complex process, as many degradation mechanisms act together toward the total destruction of the material. The degradation is the complicated non-linear time-dependent process that affects directly, or indirectly several properties of the material related to its functional characteristics. In its final stage of degradation, a material does not meet its functional requirements and is easily prone to mechanical failure [2]. The essential degradation in the mechanical characteristics (modulus of elasticity, fracture stress, and elongation at break) has been evaluated in our previous work [1].

S. N. Zhurkov [3] has study the time to failure for uniaxial tensile specimens of some 50 materials, measured in some cases over test decades of time. He has suggested a universal rate relation between lifetime, stress, and temperature of the form $\tau = \tau_0 \exp[(U_0 - \gamma\sigma)/RT]$. The constant τ_0 is essentially the reciprocal of the natural oscillation frequency of atoms in the solid, U_0 is the binding energy on the atomic scale, and γ is proportional to the disorientation of the molecular structure. Assuming the kinetic nature of bond destruction through the thermofluctuation mechanism, direct experimental verification of the phenomenon for polymers has been obtained using electron paramagnetic resonance. Yu. M. Boiko et al. [4] have studied the potentials of the multi-stage hot-zone drawing technique for enhancing the tensile strength σ of ultra-high-molecular-weight polyethylene (UHMWPE) gel-

cast highly oriented film threads, the applicability of the Weibull statistics to the σ distribution, and the solvent role in the film thread strength are presented. E. Damaskinskaya et al. [5] have developed an analysis of data obtained in laboratory investigations of deformation of rocks by acoustic emission and X-Ray microtomography. We found that defect accumulation occurs in fundamentally differing manners during loading. At first, defects are generated randomly and have a specific size determined by a typical structural element of a material (e.g., a grain in granite). Then the defects with sizes not dictated by the material structure are generated. D. Briassoulis et al. [6] have attempted to collect and compare the standards of plastic films for use as greenhouse cover. They have noted about the disparity of the results and the lack of coordination in the field of greenhouse covering. M. Turmine et al. [7] have investigated the effect of water on PE films used in greenhouses. The water that condenses forms round droplets because this is the shape that minimizes the contact area between the water and the film. The reason why water condenses in that form is the difference between the surface tensions of the water and the hydrophobic polymer. These droplets reduce the light transmission. A. Dehbi et al. [8, 9] investigated the behavior of PE under normal usage conditions (temperature, sand wind etc.) The qualitative and quantitative effects of the various ageing factors on the degradation of the film are usually monitored by measuring selected critical properties of the material. Chemical changes in the PE polymer structure can explain the degradation mechanisms. Changes in selected critical properties (e.g. mechanical, physical and chemical) can be used to monitor the evolution of ageing.

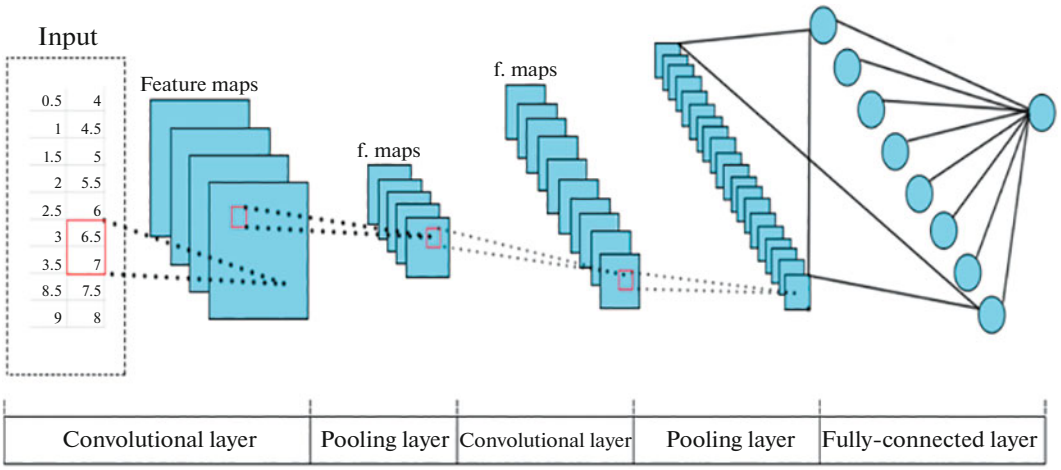
The basic information needed to understand and predict the degradation of mechanical performance of greenhouse LDPE films is the stress-strain curve. Indeed, the stress–strain curve of greenhouse LDPE films obtained from a tensile test gives information concerning the material's elastic properties, the character and extent of its plastic deformation, and its yield and tensile strength. Also, to get deep insight on the effect of the ageing time on the film properties (e.g., mechanical performance) under the different ageing conditions, one can plot the film property value after exposure to ageing conditions normalized with its corresponding property value of the unaged material against ageing time. This will also provide information about the limit of use service time according to the criterion of the 50% loss in the original property [10]. Furthermore, we need a model for examining their ultimate lifetime; in other words, a model for the prediction of the material lifetime under the conditions of use. This will help manufacturers to improve their products and be useful for engineering designers.

However, to increase the lifetime of the greenhouse LDPE films it is important to predict stress-strain curves for a given ageing time, knowing that the process of experimentally investigating LDPE mechanical properties can be costly and time-consuming. Deep learning (DL) is a major concept in machine learning, which is a subset of artificial intelligence (AI) that enables computers to learn from raw data, gather insights, and make predictions about new data using the information learned. DL has been able to develop so rapidly in recent years mainly due to the rapid development of computer hardware that provides a powerful computing power [11]. Deep neural networks have been shown to have high degrees of accuracy and precision that exceed the abilities of standard statistical techniques and human performance to make predictions about outcomes in various domains, such as computer vision, self-driving cars, speech recognition, natural language processing, and pattern detection [12]. The applications of deep neural networks in material science are still limited. It is mainly due to the difficulties of assembling big dataset in material science [13]. Indeed, few works have been done in this area, such as material classification [14], defect classification [15], microstructure reconstruction [16], and microstructure image recognition [17].

The principal aim in this work is to combine a recent deep learning based method and a traditional classifier Support Vector Machine (SVM) to predict both the entire stress-strain curves and the material lifetime under the conditions of use. To the best of the author's knowledge, machine learning based methodology has not previously been used to predict stress-strain curves of LPDE films. In addition, our goals in this work are to demonstrate that deep convolutional neural networks (CNN) can be applied in the field of polymer characterization, capture the nonlinear relationship between the ageing of the polymeric greenhouses covers and their stress-strain curves, and show the feasibility of training CNN model with a small dataset.

Background

In recent years, the convolutional neural network has become an important state-of-the-art deep learning algorithm. CNN is a feedforward neural network with a deep structure that has one or more convolutional layers, which is inspired by the organization of animal visual cortex [18]. The network employs a mathematical operation called convolution, hence the name, used for feature extraction. This can be viewed as multiplication by a matrix [19]. Unlike the traditional fully connected neural networks, CNN neurons have local connectivity (sparse interactions), so they do not have to be connected to all the outputs from the previous layer of neurons. CNNs are designed to process data that come in the form of multiple arrays [11].



Scheme 1.

3D CNNs process 3D data such as video or volumetric images; 2D CNNs operates on 2D data like images or audio spectrograms; 1D CNNs process 1D data such as signals and sequences. 1D CNNs, used in this work, are the recent variants of the well-known 2D CNNs, they were introduced only a few years ago [20, 21]. 1D CNNs have immediately achieved the state-of-the-art performance levels in various applications such as cardiac arrhythmia classification [22], electrical motor fault detection [23], wind prediction [24], and acoustic waste sorting [25].

A typical CNN, as shown in Scheme 1, is generally a convolution layer adjacent to a pooling layer, alternating in turn and finally output by a fully connected layer.

The core of the CNN is the convolution operation, where a small square matrix of numbers, called a kernel (filter), is applied across the input, which is a matrix of numbers, called a tensor. At each location of the input, a Hadamard product between the kernel and the tensor is calculated and summed to obtain the output value in the corresponding position of the target tensor, as shown in Scheme 2. For example, if we use two-dimensional data I of $p \times q$ elements as our input, we probably also want to use a two-dimensional kernel K of $n \times n$ elements, the discrete convolution, typically denoted with an asterisk, is calculated by the following formula [26]:

$$V(i,j) = (I * K)(i,j) = \sum_{x=0}^{n}\sum_{y=0}^{n} I(i+x, j+y)K(x,y), \quad (1)$$

where $V(i, j)$ is the output value in the corresponding position of the target tensor.

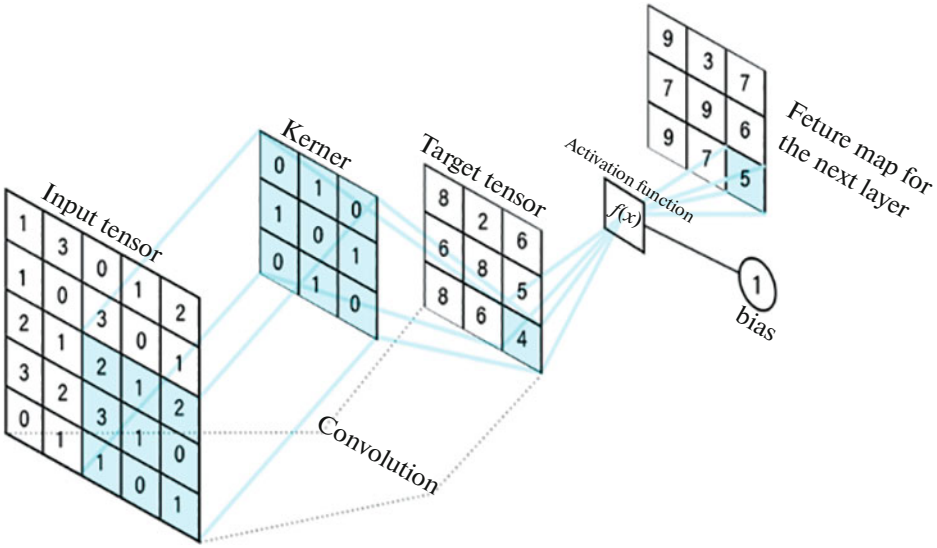
A convolutional layer contains a set of kernels to be learned. In the same layer, neurons with the same feature map share the same kernel [27]. Each neuron of the convolution layer performs a discrete convolution between the input and corresponding kernel to generate the input feature map of the neuron, which is then passed through a nonlinear activation function to generate the output feature map of the neuron. The output volume of the convolutional layer is obtained by stacking the feature maps of all kernels along the depth dimension. The role of the convolutional layer is to detect local conjunctions of features from the previous layer [11].

As in all neural networks, it is also possible to add biases to the forward operations. Each unique kernel in a convolutional layer is associated with its own bias. Therefore, for the jth feature map in the ith layer of the CNN, the value at the xth row and yth column is denoted as $v_{ij}^{x,y}$ and computed by the formula below [28]:

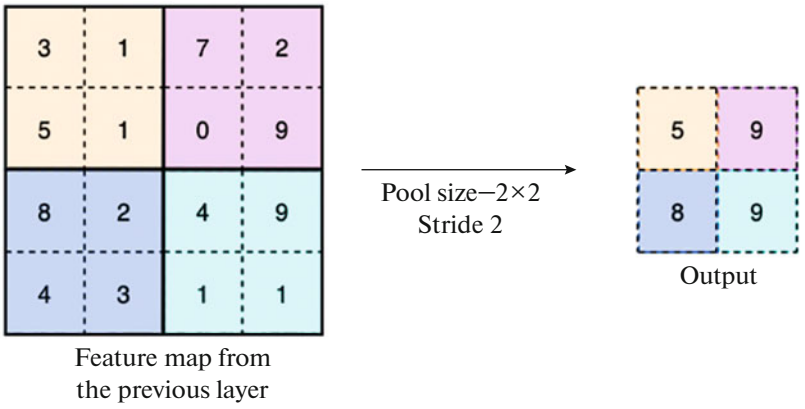
$$v_{ij}^{x,y} = f\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1} w_{ijm}^{p} v_{(i-1)m}^{x+p,y}\right), \quad (2)$$

where f is the activation function, b_{ij} the bias of the jth kernel in the ith layer, m indexes over the set of feature maps in the $(i-1)$th layer connected to the current feature map, w_{ijm}^{p} is the value at the position p of the convolutional kernel, and P_i is the length of the convolutional kernel.

The pooling layer (subsampling layer), creates its own feature map by applying pooling operator to aggregate information within each small region of the input feature maps and then downsampling the results. The goal of a pooling layer is to produce a summary statistic of its input and to reduce the spatial dimensions of the feature maps [29]. It is of note that there is no learnable parameter in any of the pooling layers. The most popular form of pooling operation is max pooling, which reports the maximal values in



Scheme 2.



Scheme 3.

each rectangular neighborhood of each point (i, j), computed by the formula below:

$$v_{ij}^{x,y} = max_{1\le q\le Qi}(v_{(i-1)j}^{x+q,y}), \quad (3)$$

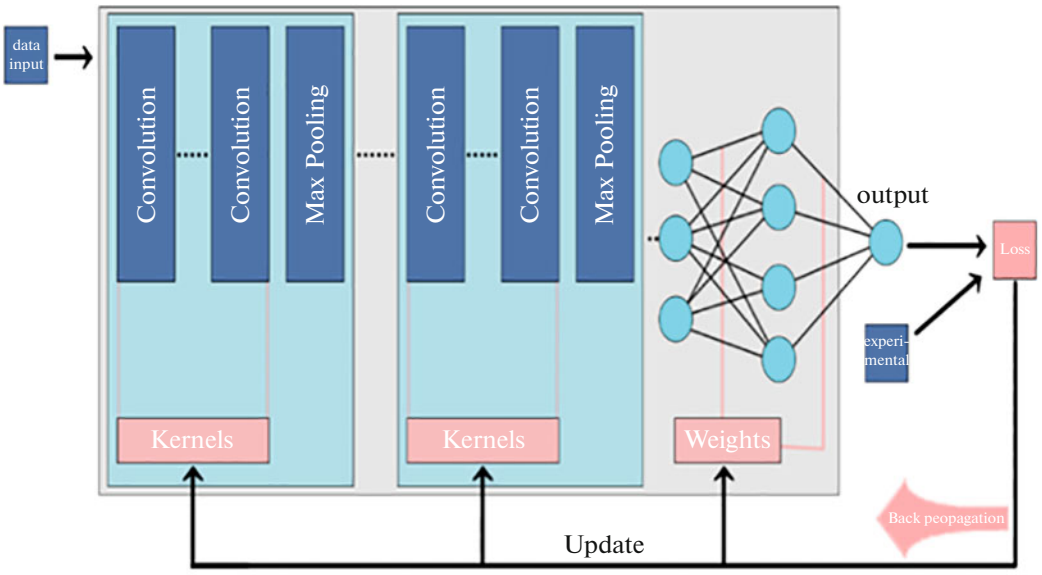
where Q_i is the length of the pooling region.

The most common form of max pooling uses stride 2 together with pool size 2 [29], which corresponds to partitioning the feature map spatially into a regular grid of square with side 2 and taking the maximum value over such blocks for each input feature as shown in Scheme 3.

The fully connected (dense) layer, identical to the layer of a typical Multilayer Perceptron (MLP), maps the extracted features into final output, such as classification. Flattening is the process of converting the output feature maps of the final convolution or pooling layer into a one-dimensional array of numbers (or vector), which is presented as input to the dense layer.

CNN hyperparameters shape how the network functions and must be tuned in order to obtain optimal model performance. Tuning model hyperparameters are usually fixed settings. Two key hyper parameters that define the convolution operation are size and number of kernels, whereas pool size, stride, and padding are hyperparameters in pooling operations. Stride is the distance between two successive kernel positions, while padding adds rows and columns of zeros on each side of the input tensor.

The loss function compares the target and predicted phases in the final output layer, while the kernels in convolution layers and weights in fully connected layers are updated based on back propagation and gradient descent algorithm in a way that the value of the loss function is minimized, as illustrated in



Scheme 4.

Scheme 4. All CNN hyperparameters should be set before the training process starts. It should be noted that the back propagating gradients through a CNN is as simple as through a regular network [11].

Support Vector Machines (SVM) is a learning algorithm originally introduced by Vapnik [30, 31] and successively extended by many other researchers. SVM is a widely used and one of the most powerful supervised classification techniques [32, 33]. Based on the training data $\{x_1, ..., x_n\}$, where $x_i \in N$, together with class labels $\{y_1, ..., y_n\}$ where $y_i \in \{-1, 1\}$, SVM can be trained to create a model [34]. Using this model, it predicts the class of new testing sample. SVM creates a hyperplane with the largest margin between two classes as shown in Scheme 5. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points. This helps to reduce the generalization error for classifying a new data point.

Linear SVM separate data in the n-dimensional input space with the use of the decision hyperplane defined as:

$$f(x) = w^T x + b = 0, \quad (4)$$

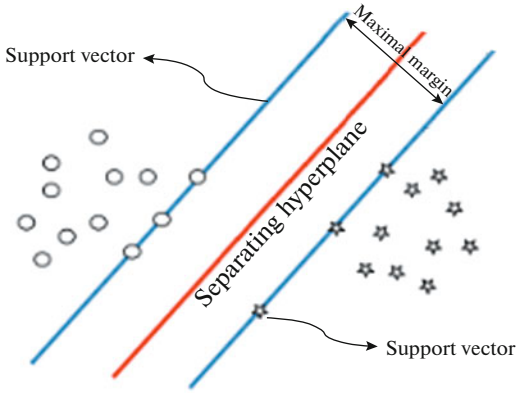
where w is the hyperplane normal vector, $w \in \mathbb{R}^n$, and $b/\|w\|$ is the perpendicular distance between the hyperplane and the origin ($\|\cdot\|$ is the 2-norm), $b \in \mathbb{R}$. This hyperplane is positioned such that the distance between the closest vectors of the opposite classes to the hyperplane is maximal. For two linearly separable classes (as already mentioned, with the class labels $y_i \in \{+1, -1\}$), the training data must satisfy the following conditions:

$$w^T x_i + b \geq 1 \quad y_i = +1, \quad (5)$$

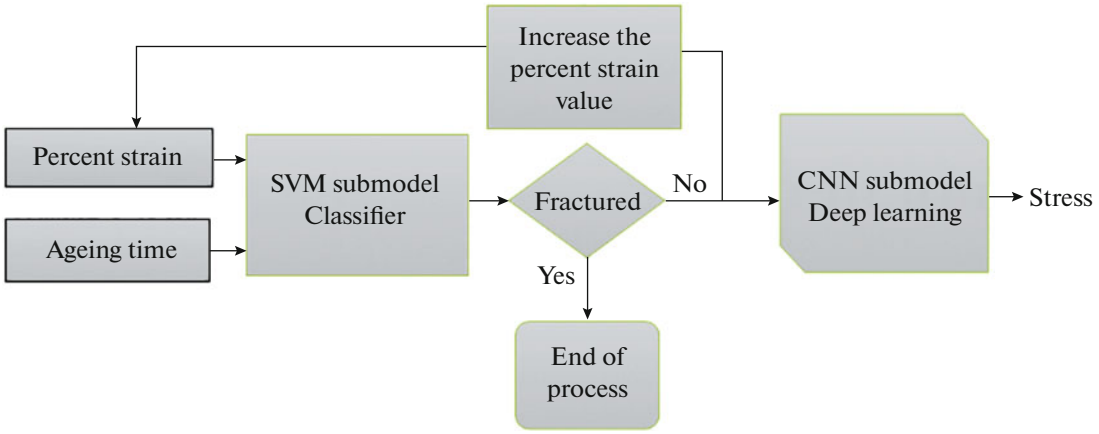
$$w^T x_i + b \leq 1 \quad y_i = -1. \quad (6)$$

These two hyperplanes are called canonical hyperplanes. The width of the band formed by the canonical hyperplanes is $2/\|w\|$. To find the maximum margin separator, it is sufficient to search among the separators checking for all the examples $y_i \times f(x_i) \geq 1$, the separator for which $\|w\|$ is minimal.

In cases where data points are clustered so that linear separation is not possible, the data points can be mapped into feature space (higher dimensional space) where a linear separation is possible. This hyperplane which is linear in feature space will be non linear in its corresponding input space [35]. Different kernel func-



Scheme 5.



Scheme 6.

tions, including linear kernel ($\mathbf{x}^T\mathbf{x}'$), polynomial kernel ($(\mathbf{x}^T\mathbf{x}' + 1)^d$), RBF kernel ($\exp(-\gamma\|\mathbf{x}\mathbf{x}'\|2)$) and sigmoidal kernel ($\tanh(\gamma\mathbf{x}\mathbf{x}' + C)$) can be used in SVM for the nonlinear problem [36].

EXPERIMENTAL

The LDPE film used in this study is manufactured by Agrofilm SA (Setif-Algeria) using the three-layer co-extrusion technology which is used in several industrial applications. The total thickness of the three layered coextruded film is 180 μm with proportions of 1/4, 1/2, and 1/4 over the total film thickness. The LDPE melt flow index has a value of 0.33 g/10 min without stabilizer and 10 g/10 min with stabilizer. The density of LDPE before extrusion is 0.923 g/cm^3. The initial color of the film is milky yellow. The real composition of the film is kept confidential by the supplier.

In this study, we used the dataset from our previous work [37], where the mechanical behavior of the unaged/virgin and naturally and artificially aged films was evaluated in different combined condition of temperatures (10, 25, 40 and 50°C) water and UV-A radiations by conducting the tensile test.

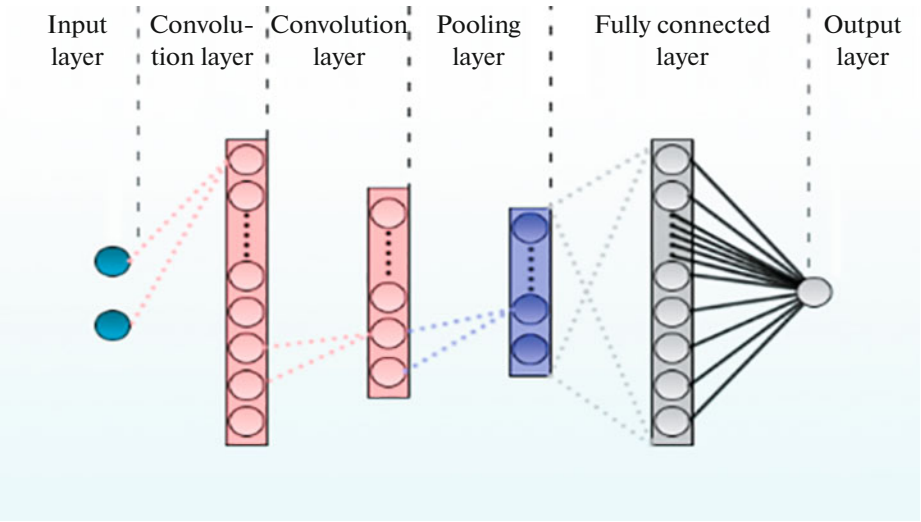
The LDPE films were photooxidized with UV-A lamp of 40 W power (Philips: TL-K 40W UV-A). This lamp gives radiation with wavelengths of between 315 and 380 nm (equivalent to that of the UV-A of the solar radiation). The distance between the lamp and the sample is 40 cm. Samples were irradiated for the different periods. The same materials were also subjected to thermal aging at the different temperatures and times. A greenhouse of 32 m length, 8 m width and 3.50 m height was specially design, built and equipped with a tri-layer LDPE film of 180 μm total thickness. The greenhouse was located in the neighborhood of Oran region of Algeria at 31° 40' N latitude, 00° 36' longitude and 120 m altitude with an east/west orientation. The samples have been taken from each side of the greenhouse roof to represent the north side film and south side film. The samples have been taken every month over the total ageing period.

The tensile tests were performed according to ISO 527-3 [38, 39]. The load was applied on specimen in a direction parallel to the average molecular orientation obtained during the film processing. The engineering stress as a function of the engineering strain has been recorded for each film. Five ageing periods were considered, namely, 120, 550, 1010, 2040, and 4000 h. In other words, a dataset of 4072 coordinate points of the five stress-strain curves was created.

A deep CNN and SVM are combined to propose a hybrid model to predict both the entire stress-strain curves and the material lifetime under the conditions of use, called hybrid SVM-CNN. The prediction of stress-strain curve consists of two problems: one is a classification problem to separate each point in the curve as a function of strain values, i.e., material's fracture classification; the second is a regression problem to predict the stress values.

The hybrid SVM-CNN model, as shown in Scheme 6, is composed of two sub models namely SVM sub model and CNN sub model. The two sub models were trained separately and were then tested together.

In the proposed model, SVM works as a classifier and CNN performs as a features extractor. This hybrid model automatically extracts features from the raw data and generates the predictions. As can be seen in Scheme 6, the hybrid SVM-CNN model has two inputs: the percent strain and the ageing time. The data first passes through the SVM submodel, in which the value of percent strain is classified, according to the ageing time, to the appropriate class (i.e., the material is fractured or not). If the material is fractured, the process ends; otherwise, data passes through the CNN submodel for predicting the stress value, then only the percent strain value is incremented and the procedure is repeated from the beginning.



Scheme 7.

The implementation code was written in Python 3.8 on a PC (Intel(R) Core(TM) i5-6200U CPU 2.4 GHz, 8 Gbyte RAM) while for the CNN submodel we have used Keras deep learning framework with a Tensor Flow backend [40], and Scikit-learn library [41] was used for the SVM submodel.

The performance of a deep convolutional neural network is strongly linked to its structure and the choice of the hyperparameters that define the learning process. Deep CNN is mostly used in image feature extraction scenarios. However, there is a lack of knowledge on how to construct a deep CNN model that could be applied to solve the regression problems. Indeed, one of the most crucial issues is how to solve the problem of how to design a reasonable model for continuous input and output, targeting the nonlinear regression problem. On the one hand, the use of deep CNN remains challenging in small datasets. Our dataset has 4072 coordinate points, but they represent only five stress-strain curves. The model needs to implement efficient feature extraction to improve the prediction results. After tried many structures, we eventually found that, for the prediction of stress-strain curves, the required structure has its own unique characteristics. The final deep learning architecture of the 1D CNN submodel is shown in Scheme 7. The proposed 1D CNN submodel has an input layer, two convolutional layers, a pooling layer, a fully connected layer and an output layer. The two convolutional layers of this neural network can be viewed as features extractors, the next layer merge semantically similar features into one and the last layer maps the extracted features into final output. This 1D CNN submodel takes as inputs the percent strain and ageing time, and predicts the corresponding stress. The hyperparameters: filter sizes, stride, padding, pool size and number of filters were tuned by experiences for an optimally designed 1D CNN. The selected hyperparameter values are shown in Table 1.

The model was trained with the log hyperbolic cosine (*log-cosh*) loss function. In fact, *log-cosh* approach has been widely used in regression based problem for smoothing the curve [42].

The *log-cosh* loss formula is given as follows:

$$L(X) = \frac{1}{N}\sum_{i=1}^{N}\log(\cosh(y_i - f_i(X))), \qquad (7)$$

where $f_i(x)$ is the predicted values, y_i is the true values, $\cosh(x) = \frac{e^x + e^{-x}}{2}$ and N is the number of samples.

Table 1. A list of hyperparameters in the 1D CNN submodel

	Hyperparameters
Convolution layer	Kernel size = 1, filters = 64, strides = 1, padding = valid, activation = relu
Convolution layer	Kernel size = 1, filters = 32, strides = 1, padding = valid, activation = relu
Pooling layer	Pooling method: maxpooling, pool size: 2, stride = 2, padding = valid
Fully connected layer	Neurons = 50, activation function = relu

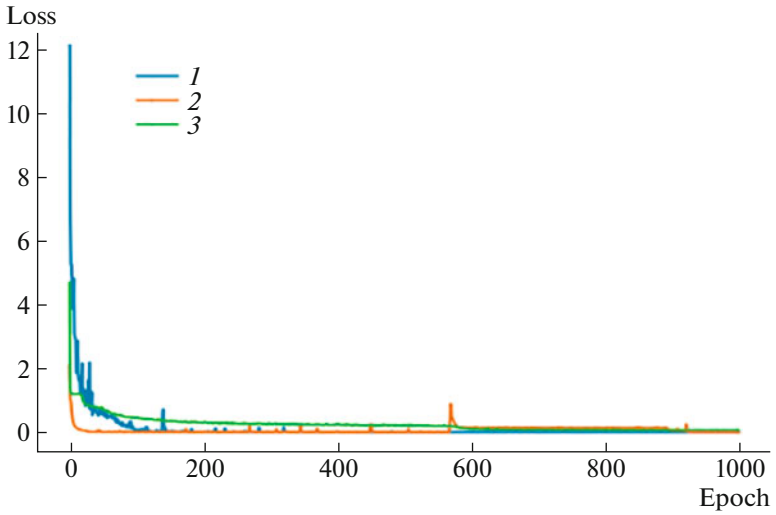


Fig. 1. Comparison of loss curves during the training progress: (*1*) ReLU, (*2*) tanh, (*3*) sigmoid.

Three well-known activation functions have been tested successively, and they are hyperbolic tangent (tanh), sigmoid and rectified linear unit (ReLU). We have used the same activation function in all layers, except the output layer where we have used linear activation function. The performances are displayed in Fig. 1 and Table 2. We can see that the model converges much faster using ReLU as activation function. A ReLU of x is simply the maximal value of 0 and x, meaning that it will return a 0 if the input is negative or the raw input otherwise. In symbols: $f(x) = \max(x, 0)$.

We have used batch gradient descent for training to provide a more stable error gradient, which resulted in more stable convergence [43]. In batch gradient descent, the error is computed for every example within the training dataset, but the model will be updated only after the evaluations of all training examples are completed. This is repeated until there is convergence. A single pass through the complete training dataset is called an epoch, which is a hyperparameter to be set.

Table 2. Effect of activation functions on model convergence

Epochs	log-cosh error		
	tanh	Sigmoid	ReLU
1	2.42	3.48	11.2
2	1.05	1.20	5.84
4	0.45	1.20	5.24
8	0.22	1.20	3.58
16	0.08	0.81	1.18
32	0.03	0.75	0.95
64	0.05	0.74	0.24
128	0.002	0.59	0.02
256	0.002	0.20	0.01
512	0.01	0.11	0.002
1024	0.15	0.02	0.001

Four of the five available stress-strain curves (80% of the dataset) were used to train the 1D CNN submodel. The test set is the data that was never used by the model during training. However, the fifth curve (20% of the dataset) was used to test the submodel. In other words, the trained 1D CNN was requested to predict the fifth curve and the output of the network is compared with the available experimental data to check the validity of the submodel.

The Adam (Adaptive Moment estimation) optimizer [44] was used to update the parameters during back propagation. The prediction model was trained with up to 1750 epochs. Early stopping was applied using Callback functions to prevent over fitting (i.e., network shows high accuracy during training but less accuracy when new data is given during testing). At the end of the training process, the training error value was 0.0001302.

We have evaluated the quality of the prediction model using the three metrics of mean squared error (MSE), root-mean squared error (RMSE) and mean absolute error (MAE). These metrics are widely used to evaluate the error rates for prediction using regression models. The results obtained are shown in Table 3. All the evaluation metrics were calculated using Python package keras-metrics [40].

Let y_i^* represent the values of variables for n prediction samples of stress, and let y_i represent the observed

values. Eqs. (8) to (10) then represent the MSE, RMSE and MAE, respectively:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^*)^2, \quad (8)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^*)^2}, \quad (9)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^*|. \quad (10)$$

For each activation function, the trained network was requested to simulate stress-strain curve of ageing time1010 hours and the results was compared against the available experimental data to check the validity of the submodel. It should be noted that the network has not seen this curve during the training phase.

As shown in Fig. 2, the most accurate prediction is obtained using the ReLU activation function. The curves exactly coincide with each other, but the predict curve exceeds the fracture point, even for the training curves while the simulation is stopped manually. For this reason, we have decided to add another submodel to predict the fracture point using SVM. Once the fracture point is predicted, the simulation will stop automatically without any human action.

Table 3. Evaluation metrics for 1D CNN submodel

Activation functions in hidden layers	Evaluation metrics		
	MSE	RMSE	MAE
tanh	1.46	1.21	0.69
Sigmoid	1.51	1.23	1.4
ReLU	0.0002	0.0155	0.02

The final point in the stress-strain curve, as shown in Scheme 8, is known as fracture point or fracture stress. A support vector machine submodel was designed to predict the fracture stress of aged films.

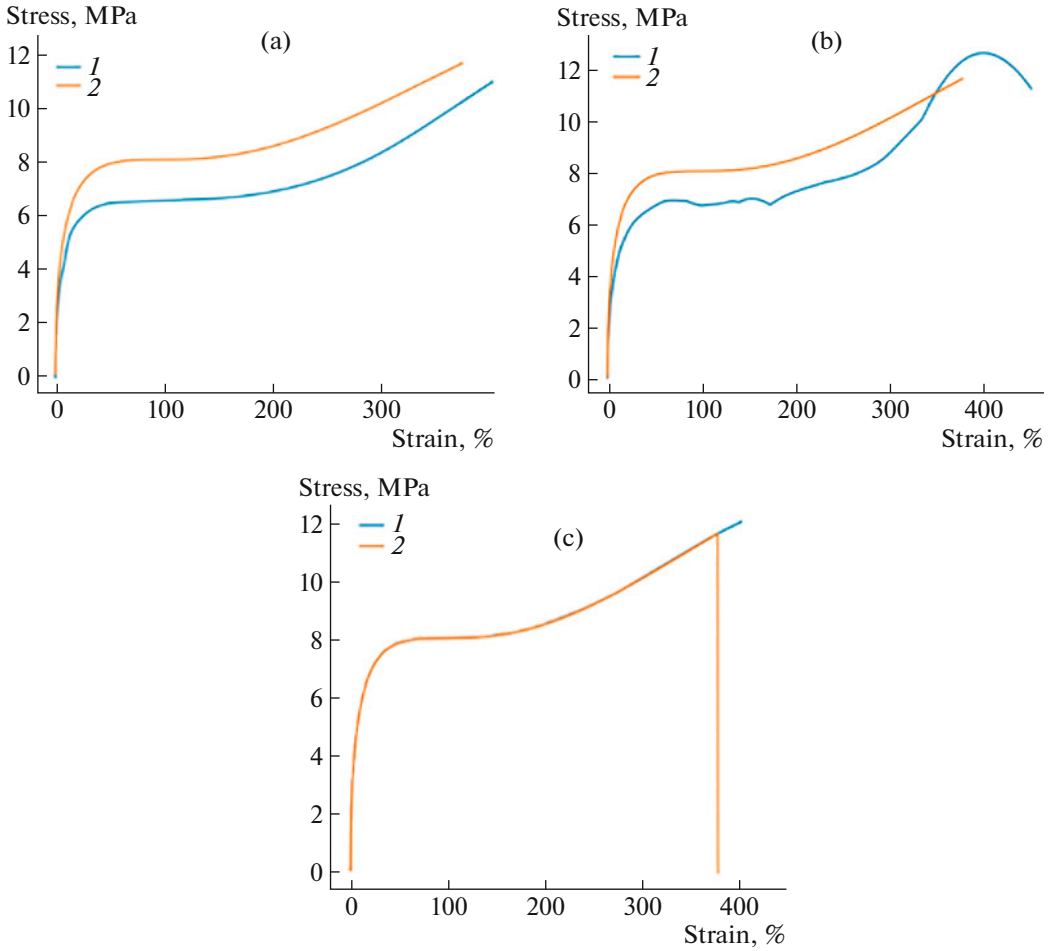


Fig. 2. Graphical comparison between (*1*) predicted and (*2*) unseen experimental stress-strain: (a) model with tanh activation function, (b) model with sigmoid activation function, (c) model with ReLU activation function. The ageing time is 1010 h.

Each point in stress-strain curve was treated as individual binary classification problem. Therefore, in the stress-strain curve, the first class was labeled '1' corresponding to strain values, from zero up to the breaking strain. Exceeding this value, the second class was labeled '0' as shown in Scheme 8. The inputs to the SVM submodel are percent strain and ageing time as shown in Scheme 9, the target output is a binary value 1 or 0 corresponding to not fractured or fractured. The SVM submodel was trained using a set of 4072 coordinate points of the stress-strain curves, manually labeled. To evaluate the SVM submodel we have used the most popular metrics [45], which are accuracy and AUC. Accuracy is basically the proportion of correct results among all predictions. If y_i^* is the predicted value of the i^{th} sample and y_i is the corresponding true value, then the fraction of correct predictions over N samples is defined as:

$$accuracy(y, y^*) = \frac{1}{N}\sum_{i=0}^{N-1} 1(y_i^* = y_i), \qquad (11)$$

where $1(x)$ is the indicator function.

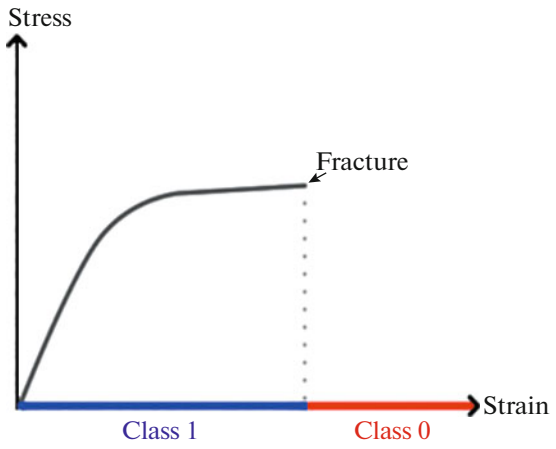
Area under the curve (AUC) is a combined measure of sensitivity and specificity. Assume we are given dataset $D = \{(x_i, y_i)\}$, $I = 1, \ldots, N$ with N examples, class labels $y_i \in \{-1, +1\}$, and input vectors x_i; the number of positive examples (i.e., where $y_i = +1$) is N^{pos}, and the number of negative examples is $N^{neg} = N - N^{pos}$. The AUC of a predictor f is defined as [46]:

$$AUC(f, D) = \frac{1}{N}\sum_{j=1}^{N^{neg}}\sum_{k=1}^{N^{pos}} 1[f(x_j) < f(x_k)], \qquad (12)$$

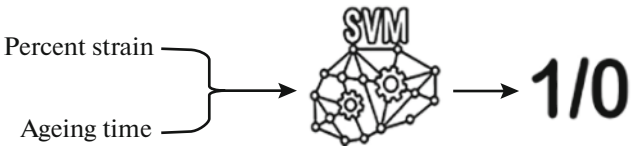
where $1(a)$ is the indicator function; $1(a) = 1$ if a is true and 0 otherwise. The AUC and the accuracy were calculated using Python 3.8 with Sklearn metrics [41]. We have obtained 0.969 as AUC and 0.98 as accuracy, which means that the submodel performs well, i.e., the material is fractured or not is well predicted.

RESULTS AND DISCUSSION

Our results for the hybrid SVM-CNN model are shown in Fig. 3. All figures in this work were generated using Matplotlib [47]. For each ageing time, the model was requested to predict the stress-strain curve and the output of the model was compared with the available experimental data. It can be seen that the curves exactly coincide with each other. One can notice that the model demonstrated a strong ability to simulate the trend of the curves. The mechanical properties such as yield strength and fracture stress and strain are predicted with accuracy that is very close to the maximum of 100%. Besides, from the predicted stress-strain curves, one can calculate other mechanical properties such as tensile strength, percent elongation,



Scheme 8.



Scheme 9.

young modulus, toughness and the ratio of tensile strength to young's modulus.

Designing and tuning deep neural networks require large efforts from developers, especially with a small dataset like ours. Indeed, unlike typical CNN, we have used two convolutional layers successively, this makes the 1D CNN submodel perform well. Our dataset is composed by five ageing periods up to 4000 h, although the hybrid SVM-CNN model can exceed these periods and predicts the stress-strain curves for any given ageing time. For example, as shown in Fig. 4, the model can easily predict the curves for 5000, 6000, and 9000 h. This revealed the powerful ability of learning high-level feature representations of deep CNN. In addition, the hybrid SVM-CNN model could eventually be used to predict the maximum degradation time under the condition of use. Indeed, using python loop, we have increased the ageing time until 18000 h (2 years). The model was requested to predict the stress for each value of ageing time. The maximum degradation time (ageing time) was 14325 h, from this value the model does not predict the stress and displays "material fractured". This means that the material is completely destroyed.

On the other hand, the predicted curve represents stress-strain curve of the LDPE film during degradation at selected period. Therefore, one can plot, on the same graph, both the predicted curve and the corre-

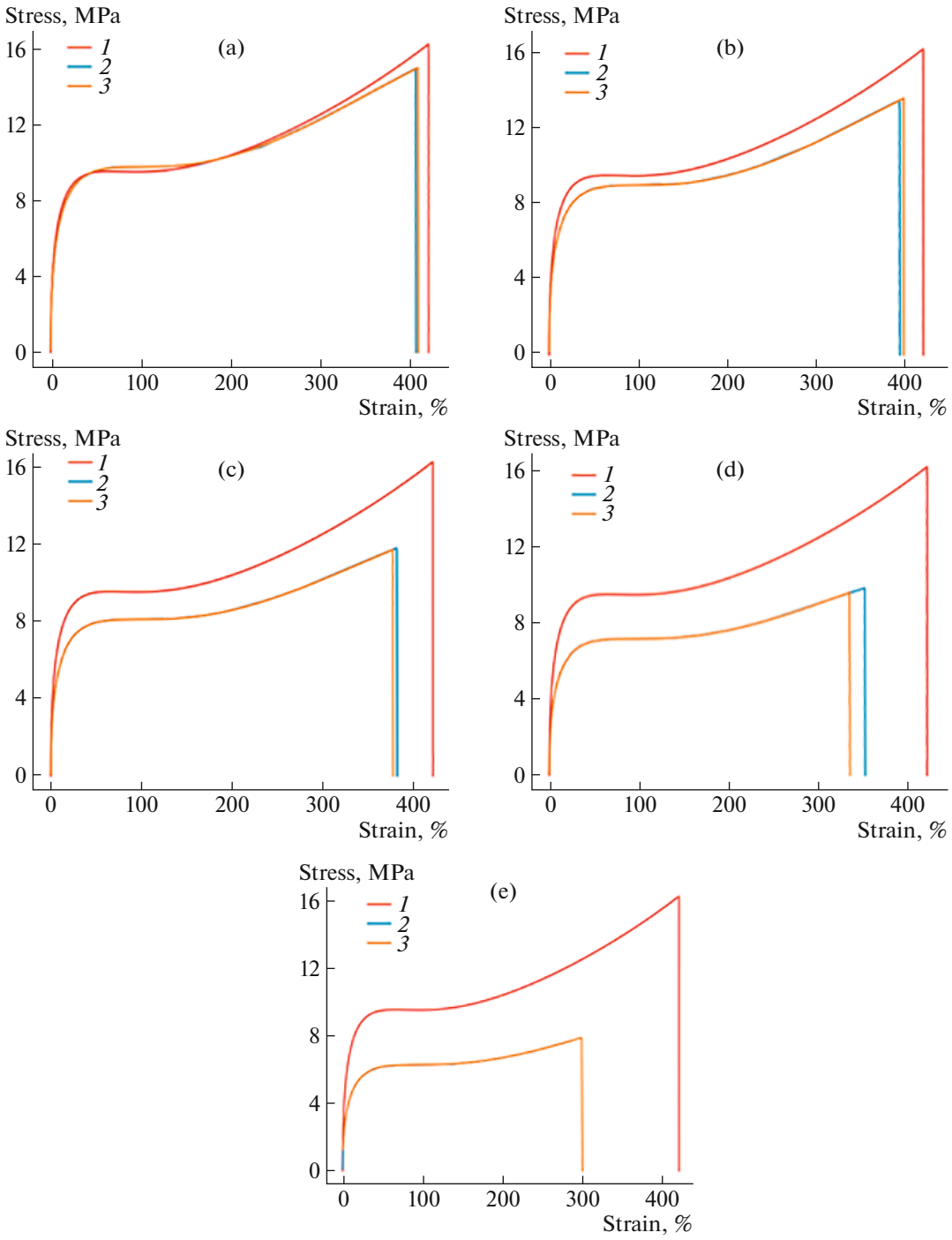


Fig. 3. Graphical comparison between stress-strain curves for (*1*) virgin film, (*2*) predicted and (*3*) experimental curves for different ageing time: (a) 120, (b) 550, (c) 1010, (d) 2040, (e) 4000 h.

sponding stress-strain curve for the unaged material, as we have done. This will also provide information about the limit of use service time according to the criterion of the 50% loss in the original property [9].

In the following, we will compare the model performance to the most commonly used type of neural networks for supervised learning.

The multilayer perceptron (MLP) is a Feedforward Neural Network (FNN) with one or more layers of neurons between the input and output layers [48]. The architecture of an MLP can be represented as an acyclic graph so that neurons in any layer are connected to all neurons in the next layer and no feedback between layers. Many studies [49–51] have shown that MLP

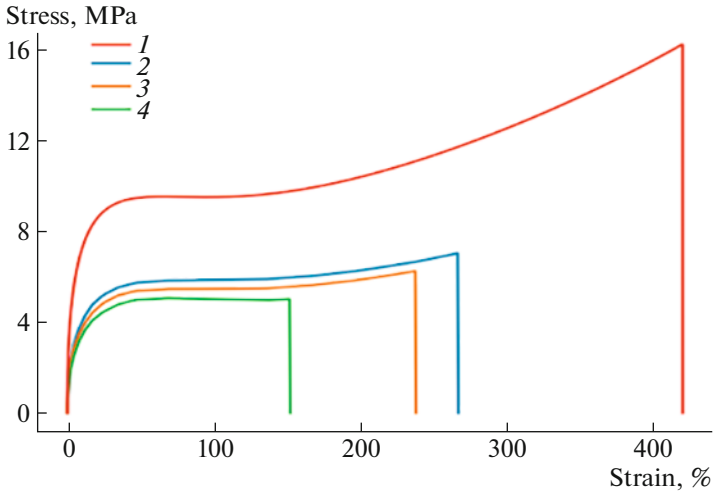


Fig. 4. Predicted stress-strain curves for (*1*) virgin film and for different ageing time: (*2*) 5000, (*3*) 6000, (*4*) 9000 h.

with one hidden layer can approximate arbitrarily well any continuous function of several real variables. Therefore, our MLP model developed to predict the stress-strain curves has one hidden layer between the input and output layers. The output neuron has a linear activation function, while the ReLU activation function has been chosen for neurons in the hidden layer. The model inputs are the percent strain and ageing time, while the output is the stress.

We have trained several MLP networks. In fact, we have increased the number of neurons in the hidden layer up to 100. We found that over 80 hidden neurons, network becomes too complex and generalization ability becomes very poor. The MLP networks were trained with batch gradient descent. The Adam optimizer was used to update the MLP weights during backpropagation, while the *log-cosh* was used as loss function. Each network was trained with 80% of dataset, i.e., four stress-strain curves and then evaluated with the stress-strain curve of ageing time 1010 hours (20% of the dataset). The performances of the best networks are shown in Table 4. The outputs of each MLP network after simulation process are shown in Fig. 5. The MLP networks were implemented using Keras [40]. A comparison between the two models (1D CNN and MLP) is done in terms of training error and testing error as illustrated in Table 5. From the table, we can see that the best performance has been obtained with the 1D CNN model.

Table 4. MLP Networks having best performances

	log-cosh error	MAE	MSE	RMSE
MLP 30 neurons	0.63	1.07	2.11	1.45
MLP 50 neurons	0.62	1.11	2.06	1.44
MLP 80 neurons	0.50	0.92	1.65	1.29

Table 5. Comparison of the two metrics for different models

	Training error (*log-cosh*)	Testing error (MSE)
1D CNN	0.0001	0.0002
MLP 80 neurons	0.50	1.65
MLP 50 neurons	0.62	2.06
MLP 30 neurons	0.63	2.11

From a structural point of view, both models (1D CNN and MLP) contain one fully connected layer, apply ReLu function as hidden activation function, use Linear function as the activation function of the output layer and *log-cosh* as the loss function; whereas, the results show that the MLP performs worse than the 1D CNN. As demonstrated in Fig. 3, the 1D CNN-based method predict very well the target stress-strain curves. 1D CNN performs well on unseen data by efficiently learning from small dataset. The failure of the MLP network, as shown in Fig. 5, results from the inability to extract features from raw data and the low size of dataset.

CONCLUSIONS

We have developed and validated a hybrid deep learning model to predict both the lifetime and the mechanical performance degradation of greenhouse tri-layer LDPE films. The DL model was successful in both training and validation phases and demonstrate the capability of the prediction of stress-strain curve of an aged LDPE film in different periods. Though CNN

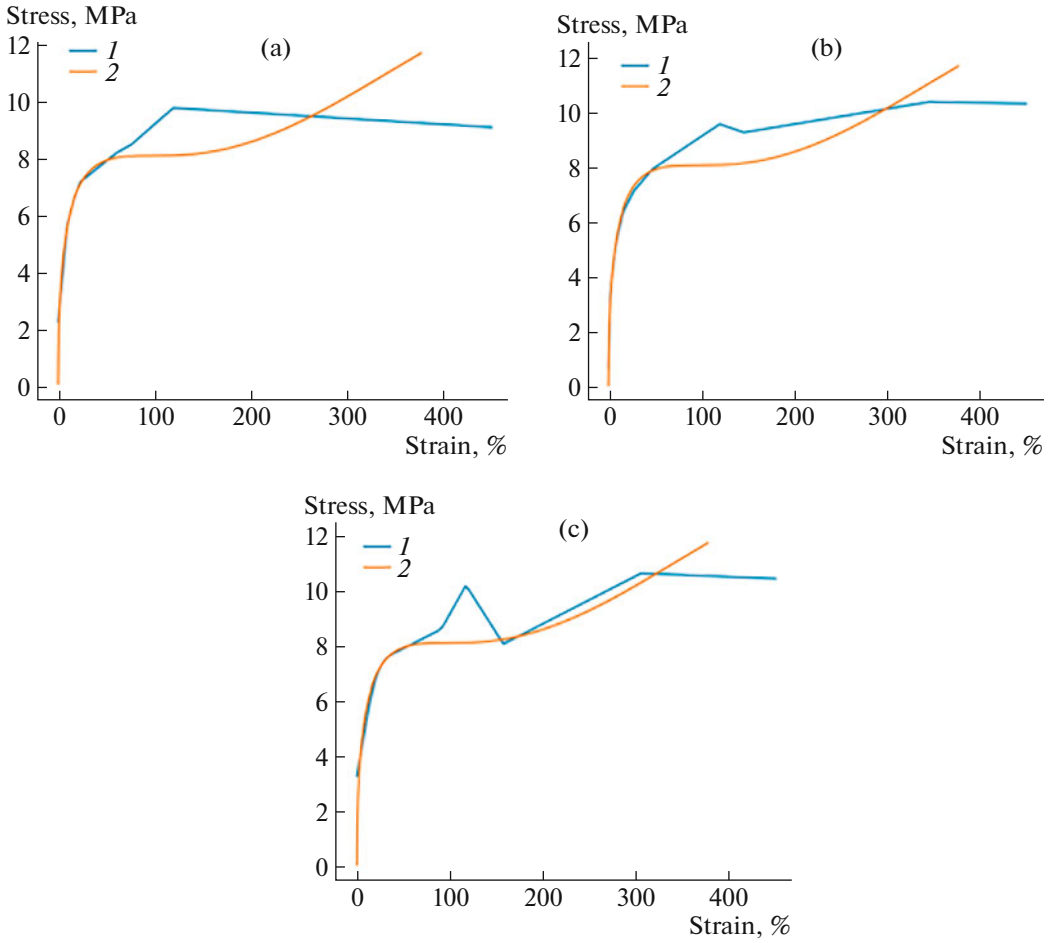


Fig. 5. Graphical comparison between MLP (*1*) predicted and (*2*) experimental curves with (a) 30, (b) 50, (c) 80 neurons. The ageing time is 1010 h.

(2D or 3D) with big datasets is the optimal solution, 1D CNN with small datasets can be a reasonable choice when big datasets are unavailable in material study. The attractive feature of 1D CNN is its ability to exploit temporal correlation in raw data. The hybrid SVM-CNN model predicts the stress-strain curves as a function of ageing time. This can save time and cost, and provide easy tools to predict both the mechanical properties of greenhouse coverings and their lifetime under the conditions of use. This will help manufacturers to create materials capable of long-term use and be useful for engineering designers. This work leads to the conclusion that DL is a promising model approach that can greatly motivate the research community to extend its applications in the field of polymer characterization.

REFERENCES

1. A. Dehbi, A. H. I. Mourad, and A. Bouaza, Process Eng. **10**, 466 (2011).
2. K. Djakhdane, A. Dehbi, A. I. Mourad, A. Zaoui, and P. Picuno, Plast. Rubber Compos. **45**, 346 (2016).
3. S. N. Zhurkov, Int. J. Fract. **26**, 295 (1984).
4. Yu. M. Boiko, V. A. Marikhin, and E. I. Radovanova, J. Mater. Sci. **52**, 1727 (2017).
5. E. Damaskinskaya, D. Frolov, D. Gafurova, D. Korost, and I. Panteleev, Interpretation **5** (4), 1 (2017).
6. D. Briassoulis, D. Waaijenberg, J. Gratraud, and B. von Elsner, J. Agric. Eng. Res. **67**, 171 (1997).
7. M. Turmine and P. Letellier, J. Colloid Interface Sci. **227**, 71 (2000).
8. A. Dehbi, A.-H.-I. Mourad, and A. Bouaza, J. Appl. Polym. Sci. **124**, 2702 (2012).
9. M. Aouinet, K. Djakhdane, A. Dehbi, D. Castronuovo, and P. Picuno, J. Elastomers Plast. **51**, 211 (2019).
10. D. Briassoulis, Polym. Degrad. Stab. **91**, 1256 (2006).
11. F. Jiang, Y. Fu, B. B. Gupta, F. Lou, S. Rho, F. Meng, Z. Tian, IEEE Trans. Sustainable Comput. **5**, 204 (2018).
12. Y. LeCun, Y. Bengio, and G. Hinton, Nature **521** (7553), 436 (2015).
13. S. Feng, H. Zhou, and H. Dong, Mater. Des. **162**, 300 (2019).

14. H. Zheng, L. Fang, M. Ji, M. Strese, Y. Ozer, and E. Steinbach, IEEE Trans. Multimedia **18**, 2407 (2016).

15. Y.-J. Cha, W. Choi, and O. Büyüköztürk, Comput.-Aided Civ. Infrastruct. Eng. **32**, 361 (2017).

16. X. Li, Z. Yang, L. C. Brinson, A. Choudhary, A. Agrawal, and W. Chen, in *Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Canada, 2018* (Quebec City, 2018), pp. V02BT03A008–V02BT03A008.

17. A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, and L. Song, Acta Mater. **146**, 76 (2018).

18. K. Fukushima, Biol. Cybern. **36**, 193 (1980).

19. J. Heaton, Gen. Program. Evolvable Mach. **19**, 305 (2018).

20. S. Kiranyaz, T. Ince, and M. Gabbouj, IEEE Trans. Biomed. Eng. **63**, 664 (2016).

21. S. Kiranyaz, O. Avci, O. Abdeljaber, TurkerInce, M. Gabbouj, and D. J. Inman, Mech. Syst. Signal Process. **151**, 107398 (2021).

22. Ö. Yıldırım, P. Pławiak, R. S. Tan, and U. R. Acharya, Comput. Biol. Med. **102**, 411 (2018).

23. T. Ince, Electr. Eng. **101**, 599 (2019).

24. S. Harbola, and V. Coors, Energy Convers. Manag. **195**, 70 (2019).

25. G. Lu, Y. Wang, H. Yang, and J. Zou, J. Cleaner Prod. **271**, 122393 (2020).

26. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning Book* (MIT Press, Cambridge, 2016).

27. Q. Wu, Multimedia Tools Appl. **79**, 9419 (2019).

28. P. P. San, P. Kakar, X.-L. Li, P. K. Shonali, J.-B. Yang, and M. N. Nguyen, "Deep Learning for Human Activity Recognition," in *Big Data Analytics for Sensor-Network Collected Intelligence*, Ed. by H.-H. Hsu, C.-Y. Chang, and C.-H. Hsu (Academic Press, London; San Diego; Cambridge; Oxford, 2017), pp. 186–204.

29. *Handbook of Medical Image Computing and Computer Assisted Intervention*, Ed. by S. K. Zhou, D. Rueckert, and G. Fichtinger (Academic Press, London; San Diego; Cambridge; Oxford, 2019).

30. V. N. Vapnik, *Statistical Learning Theory* (John Wiley and Sons, New York, 1998).

31. V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. (Springer, New York, 2000).

32. T. Eitrich and B. Lang, J. Comput. Appl. Math. **196**, 425 (2006).

33. J. Vaidya, H. Yu, and X. Jiang, Knowl. Inf. Syst. **14**, 161 (2008).

34. K. P. Soman, V. Ajav, and R. Loganathan, *Machine Learning with SVM and Other Kernel Methods* (PHI Learning Pvt. Ltd., New Delhi, 2009).

35. B. Scholkopf, A. Smola, R. C. Williamson, and P. L. Bartlett, Neural Comput. **12**, 1207 (2000).

36. L. Zhang, H. Chen, X. Tao, H. Cai, J. Liu, Y. Ouyang, Q. Peng, and Y. Du, Mater. Des. **193**, 108835 (2020).

37. A. Dehbi, A. H. I. Mourad, K. Djakhdane, and A. Hilal-Alnaqbi, Polym. Eng. Sci. **55**, 287 (2015).

38. *ISO 527-3: Plastics—Determination of Tensile Properties, Part 3: Test Conditions for Films and Sheets* (1995).

39. A.-H. I. Mourad, H. F. M. Elsayed, and D. C. Barton, Strength, Fract. Complexity. **2** (4), 149 (2004).

40. A. Gulli and S. Pal, *Deep Learning with Keras* (Packt Publ. Ltd., Birmingham, 2017).

41. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, C. Universite, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, J. Mach. Learn. Res. **12**, 2825 (2011).

42. S. Jadon, in *Proceedings of 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, Via del Mar, Chile, 2020* (IEEE, Via del Mar, 2020), pp. 1–7.

43. S. Ray, in *Proceedings of 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019* (IEEE, Faridabad, 2019), pp. 35–39.

44. D. P. Kingma and J. Ba, arXiv.org, e-Print Arch., Comput. Sci. 1412.6980 (2014).

45. D. Brzezinski and J. Stefanowski, Knowl. Inf. Syst. **52**, 531 (2017).

46. T. Calders and S. Jaroszewicz, "Efficient AUC optimization for classification," in *European Conference on Principles of Data Mining and Knowledge Discovery* (Springer, Berlin; Heidelberg, 2007), pp. 42–53.

47. J. D. Hunter, Comput. Sci. Eng. **9** (3), 90 (2007).

48. L. Aid, M. Loudini, and W. K. Hidouci, Int. J. Comput. Appl. **15** (5), 14 (2011).

49. N. Hahm, and B. I. Hong, Comput. Math. Appl. **47**, 1897 (2004).

50. Z. Zainuddin and O. Pauline, WSEAS Trans. Math. **7**, 333 (2008).

51. T. H. Nguyen, D. Nong, and K. Paustian, Ecolog. Modell. **400**, 1 (2019).

References

References

[1] Sarker Iqbal H. Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective. SN Comput Sci. 2021,20:21.

[2] Han J, Pei J, Kamber M. Data mining: concepts and techniques.Amsterdam: Elsevier, 2011.
[3] Iqbal H. Sarker. AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems. SN Computer Science (2022) 3:158

[4] Sarker IH, Colman A, Han J, Watters P. Context-aware machine learning and mobile data analytics: automated rule-based services with intelligent decision-making. Berlin: Springer, 2022.

[5] Sarker IH, Colman A, Han J. Recencyminer: mining recencybasedpersonalized behavior from contextual smartphone data. JBig Data. 2019,6(1):1–21.

.
[6] Zadeh LA. Is there a need for fuzzy logic? Inf Sci.2008,178(13):2751–79.

[7] Hamamoto AH, Carvalho LF, Sampaio LDH, Abrao T, Proenca ML Jr. Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Syst Appl. 2018,92:390–402.

[8] Stephan G, Pascal H, Andreas A. Knowledge representationand ontologies. Semant Web Serv Concepts Technol Appl.2007,20:51–105

[9] Maedche A, Staab S. Ontology learning for the semantic web. IEEE Intell Syst. 2001,16(2):72–9.

[10] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Burlington: Morgan kaufmann, 1988.

[11] M. Onat Topal, Anil Bas, Imke van. Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet

[12] Sarker IH. Machine learning: algorithms, real-world applicationsand research directions. SN Comput Sci. 2021,2(3):1–21.

[13] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. J ArtifIntell Res. 1996,4:237–85.
[14] Polydoros AS, Nalpantidis L. Survey of model-based reinforcement learning: applications on robotics. J Intell Robot Syst. 2017,86(2):153–73.
[15] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: machine learning in python. J Mach Learn Res. 2011,12:2825–30
[16] T. Eitrich and B. Lang, J. Comput. Appl. Math. 196,425(2006).
[17] J.Vaidya, H.Yu, and X.Jiang, Knowl.Inf.Syst.14, 161(2008)
[18] V. N. Vapnik, Statistical Learning Theory (john wiley and Sons, New York, 1998)
[19] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed.(Springer, New York, 2000)

[20] B. Scholkopf, A. Smola, R. C. Williamson, and P. L. Bartlett, Neural Comput. 12,1207(2000)
[21] L. Zhang, H.Chen, X. Tao, H. Cai, J. Liu, Y.ouyang, Q. Peng, and Y.Du, Mater. Des.193,108835(2020)
[22] Sarker IH. A machine learning based robust prediction model for real-life mobile phone data. Internet Things. 2019,5:180–93
[23] Sarker IH, Abushark YB, Alsolami F, Khan A. Intrudtree: a machine learning based cyber security intrusion detection model. Symmetry. 2020,12(5):754.
[24] Sarker IH, Abushark YB, Khan A. Contextpca: predicting context-aware smartphone apps usage based on machine learning techniques. Symmetry. 2020,12(4):499.
[25] Liu H, Motoda H. Feature extraction, construction and selection: A data mining perspective, vol. 453. Springer Science & Business Media, 1998.

[26] Sarker IH, Alqahtani H, Alsolami F, Khan A, Abushark YB, Siddiqui MK. Context pre-modeling: an empirical analysis for classifcation based user-centric context-aware predictive modeling. J Big Data. 2020,7(1):1–23
[27] Hotelling H. Analysis of a complex of statistical variables into principal components. J Edu Psychol. 1933,24(6):417
[28] Liii Pearson K. on lines and planes of closest ft to systems of points in space. LondEdinb Dublin Philos Mag J Sci. 1901,2(11):559–72
[29] Blumenstock J. Machine learning can help get covid-19 aid to those who need it most. Nature. 2020,20:20.
[30] Sarker Iqbal H. Cyberlearning: efectiveness analysis of machine learning security modeling to detect cyber-anomalies and multiattacks. Internet Things. 2021,100:393.
[31] Sarker IH, Kayes ASM, Badsha S, Alqahtani H, Watters P, Ng A. Cybersecurity data science: an overview from machine learning perspective. J Big Data. 2020,7(1):1–29
[32] Saharan S, Kumar N, Bawa S. An efcient smart parking pricing system for smart city environment: a machine-learning based approach. Future Gener Comput Syst. 2020,106:622–40
EE Access. 2019,7:93998–4011
[33] Ramzan B, Bajwa IS, Jamil N, Amin RU, Ramzan S, Mirza F, Sarwar N. An intelligent data analysis for recommendation sys
[34] Dhyani M, Kumar R. An intelligent chatbot using deep learning with bidirectional rnn and attention model. Mater Today Proc. 2021,34:817–24
[35] Li T-HS, Kuo P-H, Tsai T-N, Luan P-C. Cnn and lstm based facial expression analysis model for a humanoid robot. IEEE Access. 2019,7:93998–4011.
[36] Hamed M, Mahmoud T, Gómez JM, Kfouri G. Using data mining and business intelligence to develop decision support systems in Arabic higher education institutions. In: Modernizing academic teaching and research in business and economics. Berlin: Springer, 2017. p. 71–84
[37] Alazab A, Bevinakoppa S, Khraisat A. Maximising competitive advantage on e-business websites: a data mining approach. In: 2018 IEEE conference on big data and analytics (ICBDA), IEEE, 2018. p. 111–116.
[38] Afzaliseresht N, Miao Y, Michalska S, Liu Q, Wang H. From logs to stories: human-centred data mining for cyber threat intelligence. IEEE Access. 2020,8:19089–99.
[39] Poort J, Omrani PS, Vecchia AL, Visser G, Janzen M, Koenes J. An automated diagnostic analytics workfow for the detection of production events-application to mature gas felds. In: Abu Dhabi international petroleum exhibition and conference. OnePetro, 2020.
[40] Srinivas S, Ravindran AR. Optimizing outpatient appointment system using machine learning algorithms and scheduling rules: a prescriptive analytics framework. Expert Syst Appl. 2018,102:245–61.

[41] Sarker Iqbal H, Kayes ASM. Abc-ruleminer: user behavioral rule-based machine learning method for context-aware intelligent services. J Netw Comput Appl. 2020,10:2762.
[42] Borah A, Nath B. Identifying risk factors for adverse diseases using dynamic rare association rule mining. Expert Syst Appl. 2018,113:233–63.

[43] Bhavithra J, Saradha A. Personalized web page recommendation using case-based clustering and weighted association rule mining. Cluster Comput. 2019,22(3):6991–7002.

[44] Ruihua X, Luo F. Risk prediction and early warning for air trafc controllers' unsafe acts using association rule mining and random forest. Saf Sci. 2021,135:105125.

[45] Reddy GT, Reddy MPK, Lakshmanna K, Rajput DS, Kaluri R, Srivastava G. Hybrid genetic algorithm and a fuzzy logic classifer for heart disease diagnosis. Evol Intel. 2020,13(2):185–96.

[46] Krishnan RS, Julie EG, Robinson YH, Raja S, Kumar R, Thong PH, et al. Fuzzy logic based smart irrigation system using internet of things. J Clean Prod. 2020,252:119902.

[47] Hamamoto AH, Carvalho LF, Sampaio LDH, Abrão T, Proença ML Jr. Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Syst Appl. 2018,92:390–402.

[48] Kang X, Porter CS, Bohemia E. Using the fuzzy weighted association rule mining approach to develop a customer satisfaction product form. J Intell Fuzzy Syst. 2020,38(4):4343–57

[49] Goel D, Pahal N, Jain P, Chaudhury S. An ontology-driven context aware framework for smart trafc monitoring. In: 2017 IEEE region 10 symposium (TENSYMP), IEEE, 2017. p. 1–5.

[50] Kiran GM, Nalini N. Enhanced security-aware technique and ontology data access control in cloud computing. Int J Commun Syst. 2020,33(15):e4554.

[51] Syed R. Cybersecurity vulnerability management: a conceptual ontology and cyber intelligence alert system. Inform Manage. 2020,57(6):103334.

[52] Sarker IH, Khan AI, Abushark YB, Alsolami F. Mobile expert system: exploring context-aware machine learning rules for personalized decision-making in mobile applications. Symmetry. 2021,13(10):1975.

[53] Lamy J-B, Sekar B, Guezennec G, Bouaud J, Séroussi B. Explainable artifcial intelligence for breast cancer: a visual casebased reasoning approach. Artif Intell Med. 2019,94:42–53.

[54] González-Briones A, Prieto J, De La Prieta F, Herrera-Viedma E, Corchado JM. Energy optimization using a case-based reasoning strategy. Sensors. 2018,18(3):865.

[55] Khosravani MR, Nasiri S, Weinberg K. Application of case-based reasoning in a fault detection system on production of drippers. Appl Soft Comput. 2019,75:227–32.

[56] Corrales DC, Ledezma A, Corrales JC. A case-based reasoning system for recommendation of data cleaning algorithms in classification and regression tasks. Appl Soft Comput. 2020,90:106180.

[57] Phan HT, Tran VC, Nguyen NT, Hwang D. Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. IEEE Access. 2020,8:14630–41.

[58] Aytuğ Onan. Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. Concurr Comput Pract Exp. 2020,20:e5909.

[59] Subramaniyaswamy V, Logesh R, Abejith M, Umasankar S, Umamakeswari A. Sentiment analysis of tweets for estimating criticality and security of events. In: Improving the safety and efciency of emergency services: emerging tools and technologies for frst responders. IGI global, 2020. p. 293–319.

[60] Nawaz MS, Bilal M, Lali MIU, Mustafa RU, Aslam W, Jajja S. Efectiveness of social media data in healthcare communication. J Med Imaging Health Inform. 2017,7(6):1365–71

[61] Elakkiya R, Subramaniyaswamy V, Vijayakumar V, Aniket Mahanti. Cervical cancer diagnostics healthcare system using hybrid object detection adversarial networks. IEEE J Biomed Health Inform. 2021,20:20

[62] Harrou F, Zerrouki N, Sun Y, Houacine A. An integrated visionbased approach for efcient human fall detection in a home environment. IEEE Access. 2019,7:114966–74.

[63] Pan M, Liu Y, Jiayi Cao Yu, Li CL, Chen C-H. Visual recognition based on deep learning for navigation mark classifcation. IEEE Access. 2020,8:32767–75.

[64] Onah JO, Abdullahi M, Hassan IH, Al-Ghusham A, et al. Genetic algorithm based feature selection and naïve bayes for anomaly detection in fog computing environment. Mach Learn Appl. 2021,6:100156.

[65] Iqbal H. Sarker. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN Computer Science (2021) 2:420.

[66] Sarker IH, Abushark YB, Alsolami F, Khan AI. Intrudtree: a machine learning based cyber security intrusion detection model. Symmetry. 2020,12(5):754.

[67] Sarker IH, Salah K. Appspred: predicting context-aware smartphone apps using random forest learning. Internet of Things. 2019,8:100106.

[68] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015,521(7553):436–44.

[69] Sarker IH, Kayes ASM, Watters P. Effectiveness analysis of machine learning classifcation models for predicting personalized context-aware smartphone usage. J Big Data. 2019,6(1):1–28.

[70] Coelho IM, Coelho VN, da Eduardo J, Luz S, Ochi LS, Guimarães FG, Rios E. A gpu deep learning metaheuristic based model for time series forecasting. Appl Energy. 2017,201:412–8

[71] Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, Wang C. Machine learning and deep learning methods for cybersecurity. Ieee access. 2018,6:35365–81.

[72] Deng L, Dong Yu. Deep learning: methods and applications. Found Trends Signal Process. 2014,7(3–4):197–387.

[73] Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans Signal Inf Process. 2014, p. 3.

[74] Du K-L, Swamy MNS. Neural networks and statistical learning. Berlin: Springer Science & Business Media, 2013.

[75] Dupond S. A thorough review on the current advance of neural network structures. Annu Rev Control. 2019,14:200–30

[76] Mandic D, Chambers J. Recurrent neural networks for prediction: learning algorithms, architectures and stability. Hoboken: Wiley, 2001.

[77] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997,9(8):1735–80.

[78] Siami-Namini S, Tavakoli N, Namin AS. The performance of lstm and bilstm in forecasting time series. In: 2019 IEEE International Conference on Big Data (Big Data), 2019, p. 3285–292. IEEE.

[79] Cho K, Van MB, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using rnn encoderdecoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[80] Chung J, Gulcehre C, Cho KH, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[81] Gruber N, Jockisch A. Are gru cells more specifc and lstm cells more sensitive in motive classifcation of text? Front Artif Intell. 2020,3:40.

[82] Hands-on Bayesian Neural Networks – A Tutorial for Deep Learning Users

[83] Da'u A, Salim N. Recommendation system based on deep learning methods: a systematic review and new directions. Artif Intel Rev. 2020,53(4):2709–48.

[84] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Advances in neural information processing systems. 2014, p. 2672–680

[85] Aggarwal A, Mittal M, Battineni G. Generative adversarial network: an overview of theory and applications. Int J Inf Manag Data Insights. 2021, p. 100004.

[86] Li B, François-Lavet V, Doan T, Pineau J. Domain adversarial reinforcement learning. arXiv preprint arXiv:2102.07097, 2021

[87] Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing. 2017,234:11–26.

[88] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1. Cambridge: MIT Press, 2016.

[89] Rifai S, Vincent P, Muller X, Glorot X, Bengio Y. Contractive auto-encoders: Explicit invariance during feature extraction. In: Icml, 2011.

[90] Kohonen T. The self-organizing map. Proc IEEE. 1990,78(9):1464–80.

[91] Kohonen T. Essentials of the self-organizing map. Neural Netw. 2013,37:52–65.

[92] Qu X, Lin Y, Kai G, Linru M, Meng S, Mingxing K, Mu L, editors. A survey on the development of self-organizing maps for unsupervised intrusion detection. Mob Netw Appl. 2019, p. 1–22

[93] Marlin B, Swersky K, Chen B, Freitas N. Inductive principles for restricted boltzmann machine learning. In: Proceedings of the Thirteenth International Conference on artifcial intelligence and statistics, p. 509–16. JMLR Workshop and Conference Proceedings, 2010.

[94] Hinton GE. Deep belief networks. Scholarpedia. 2009,4(5):5947.

[95] Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep reinforcement learning: a brief survey. IEEE Signal Process Mag. 2017,34(6):26–38.

[96] Ismail WN, Hassan MM, Alsalamah HA, Fortino G. Cnn-based health model for regular health factors analysis in internet-ofmedical things environment. IEEE. Access. 2020,8:52541–9.

[97] Xue Q, Chuah MC. New attacks on rnn based healthcare learning system and their detections. Smart Health. 2018,9:144–57.

[98] Amarbayasgalan T, Lee JY, Kim KR, Ryu KH. Deep autoencoder based neural networks for coronary heart disease risk prediction. In: Heterogeneous data management, polystores, and analytics for healthcare. Springer, 2019. p. 237–48.

[99] Sevakula RK, Singh V, Verma NK, Kumar C, Cui Y. Transfer learning for molecular cancer classification using deep neural networks. IEEE/ACM Trans Comput Biol Bioinf. 2018,16(6):2089–100.

[100] Aslan MF, Unlersen MF, Sabanci K, Durdu A. Cnn-based transfer learning-bilstm network: a novel approach for covid-19 infection detection. Appl Soft Comput. 2021,98:106912.

[101] Islam MZ, Islam MM, Asraf A. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. Inf Med Unlock. 2020,20:100412.

[102] Yousef-Azar M, Hamey L. Text summarization using unsupervised deep learning. Expert Syst Appl. 2017,68:93–105.

[103] Wang J, Liang-Chih Yu, Robert Lai K, Zhang X. Tree-structured regional cnn-lstm model for dimensional sentiment analysis. IEEE/ACM Trans Audio Speech Lang Process. 2019,28:581–91.

[104] Minaee S, Azimi E, Abdolrashidi AA. Deep-sentiment: sentiment analysis using ensemble of cnn and bi-lstm models. arXiv preprint arXiv:1904.04206, 2019.

[105] Wang Y, Huang M, Zhu X, Zhao L. Attention-based lstm for aspect-level sentiment classifcation. In: Proceedings of the 2016 Conference on empirical methods in natural language processing, 2016, p. 606–615.

[106] Zhang Y, Zhang P, Yan Y. Attention-based lstm with multi-task learning for distant speech recognition. In: Interspeech, 2017, p. 3857–861.

[107] Latif S, Rana R, Younis S, Qadir J, Epps J. Transfer learning for improving speech emotion classifcation accuracy. arXiv preprint arXiv:1801.06353, 2018.

[108] Satt A, Rozenberg S, Hoory R. Efcient emotion recognition from speech using deep learning on spectrograms. In: Interspeec, 2017, p. 1089–1093.

[109] Kim J-Y, Seok-Jun B, Cho S-B. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. Inf Sci. 2018,460:83–102.

[110] López AU, Mateo F, Navío-Marco J, Martínez-Martínez JM, Gómez-Sanchís J, Vila-Francés J, Serrano-López AJ. Analysis of computer user behavior, security incidents and fraud using self-organizing maps. Comput Secur. 2019,83:38–51.

[111] Wang W, Zhao M, Wang J. Efective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. J Ambient Intell Humaniz Comput. 2019,10(8):3035–43.

[112] Wei P, Li Y, Zhang Z, Tao H, Li Z, Liu D. An optimization method for intrusion detection classifcation model based on deep belief network. IEEE Access. 2019,7:87593–605.

[113] Imamverdiyev Y, Abdullayeva F. Deep learning method for denial of service attack detection based on restricted Boltzmann machine. Big Data. 2018,6(2):159–69.

[114] Garg S, Kaur K, Kumar N, Rodrigues JJPC. Hybrid deeplearning-based anomaly detection scheme for suspicious fow detection in sdn: a social multimedia perspective. IEEE Trans Multimed. 2019,21(3):566–78.

[115] Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K. Deep learning approach combining sparse autoencoder with svm for network intrusion detection. IEEE Access. 2018,6:52843–56.

[116]Abdel-Basset M, Hawash H, Chakrabortty RK, Ryan M. Energy-net: a deep learning approach for smart energy management in iot-based smart cities. IEEE Internet of Things J. 2021.

[117] Huang C-J, Kuo P-H. A deep cnn-lstm model for particulate matter (pm2. 5) forecasting in smart cities. Sensors. 2018,18(7):2220.

[118] Piccialli F, Giampaolo F, Prezioso E, Crisci D, Cuomo S. Predictive analytics for smart parking: A deep learning approach in forecasting of iot data. ACM Trans Internet Technol (TOIT). 2021,21(3):1–21.

[119] Aqib M, Mehmood R, Albeshri A, Alzahrani A. Disaster management in smart cities by forecasting trafc plan using deep learning and gpus. In: International Conference on smart cities, infrastructure, technologies and applications. Springer, 2017. p. 139–54.

[120] Kök İ, Şimşek MU, Özdemir S. A deep learning model for air quality prediction in smart cities. In: 2017 IEEE International Conference on Big Data (Big Data), 2017, p. 1983–990. IEEE

[121] Chen D, Wawrzynski P, Lv Z. Cyber security in smart cities: a review of deep learning-based applications and case studies. Sustain Cities Soc. 2020, p. 102655.

[122] Aid lahcene, Djenane Mouloud Amine, Dehbi Abdelkader. Predicting life time and mechanical performance degradation of multilayer Greenhouse. ISSN 1560-0904, Polymer Science, Series B,2021.

[123] Bu F, Wang X. A smart agriculture iot system based on deep reinforcement learning. Futur Gener Comput Syst. 2019,99:500–7

[124] Ale L, Sheta A, Li L, Wang Y, Zhang N. Deep learning based plant disease detection for smart agriculture. In: 2019 IEEE Globecom Workshops (GC Wkshps), 2019, p. 1–6. IEEE.

[125] Sumathi P, Subramanian R, Karthikeyan VV, Karthik S. Soil monitoring and evaluation system using edl-asqe: enhanced deep learning model for ioi smart agriculture network. Int J Commun Syst. 2021, p. e4859.

[126] Chaudhuri N, Gupta G, Vamsi V, Bose I. On the platform but will they buy? predicting customers' purchase behavior using deep learning. Decis Support Syst. 2021, p. 113622.

[127] Anuradha J, et al. Big data based stock trend prediction using deep cnn with reinforcement-lstm model. Int J Syst Assur Eng Manag. 2021, p. 1–11.

[128] Deng S, Li R, Jin Y, He H. Cnn-based feature cross and classifer for loan default prediction. In: 2020 International Conference on image, video processing and artifcial intelligence, volume 11584, page 115841K. International Society for Optics and Photonics, 2020.

[129] Shao X, Kim CS. Multi-step short-term power consumption forecasting using multi-channel lstm with time location considering customer behavior. IEEE Access. 2020,8:125263–73.

[130] Huang H-H, Fukuda M, Nishida T. Toward rnn based micro non-verbal behavior generation for virtual listener agents. In: International Conference on human-computer interaction, 2019, p. 53–63. Springer.

[131] Rahman MW, Tashfa SS, Islam R, Hasan MM, Sultan SI, Mia S, Rahman MM. The architectural design of smart blind assistant using iot with deep learning paradigm. Internet of Things. 2021,13:100344.

[132] Gu B, Ge R, Chen Y, Luo L, Coatrieux G. Automatic and robust object detection in x-ray baggage inspection using deep convolutional neural networks. IEEE Trans Ind Electron. 2020.

[133] Pi Y, Nath ND, Behzadan AH. Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. Adv Eng Inf. 2020,43:101009.

[134] Chang W-J, Chen L-B, Hsu C-H, Lin C-P, Yang T-C. A deep learning-based intelligent medicine recognition system for chronic patients. IEEE Access. 2019,7:44441–58.

[135] Masud M, Muhammad G, Alhumyani H, Alshamrani SS, Cheikhrouhou O, Ibrahim S, Hossain MS. Deep learning-based intelligent face recognition in iot-cloud environment. Comput Commun. 2020,152:215–22.

[136] Liu C, Cao Y, Luo Y, Chen G, Vokkarane V, Yunsheng M, Chen S, Hou P. A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. IEEE Trans Serv Comput. 2017,11(2):249–61.

[135] Kawde P, Verma GK. Deep belief network based afect recognition from physiological signals. In: 2017 4th IEEE Uttar Pradesh Section International Conference on electrical, computer and electronics (UPCON), 2017, p. 587–92. IEEE.

[136] Kiran PKR, Bhasker B. Dnnrec: a novel deep learning based hybrid recommender system. Expert Syst Appl. 2020.

[137] Sujay Narumanchi H, Ananya Pramod Kompalli Shankar A, Devashish CK. Deep learning based large scale visual recommendation and search for e-commerce. arXiv preprint arXiv:1703.02344, 2017.

[138] Rosa RL, Schwartz GM, Ruggiero WV, Rodríguez DZ. A knowledge-based recommendation system that includes sentiment analysis and deep learning. IEEE Trans Ind Inf. 2018,15(4):2124–35.

[139] Naeem M, Paragliola G, Coronato A. A reinforcement learning and deep learning based intelligent system for the support of impaired patients in home treatment. Expert Syst Appl. 2021,168:114285.

[140] Wang X, Liu J, Qiu T, Chaoxu M, Chen C, Zhou P. A realtime collision prediction mechanism with deep learning for intelligent transportation system. IEEE Trans Veh Technol. 2020,69(9):9497–508.

[141] Sandro Sperandei. Understanding logistic regression analysis. School of Physical Education and Sports-Federal University of Rio de Janeiro, Rio de Janeiro, Bbrazil .

[142] Quantile regression : its application in investment analysis, JASSA THE FINISIA JOURNAL OF APPLIED FINANCE ISSUE 4 2009.

[143] Bayesian statistics and modelling, NATURE REVIEWS | METhoDS PrImErS | Article citation ID: (2021) 1:1.

[144] Bayesian Linear Regression, Technical Report · June 2019 .

[145] Ian T. Jolliffe, A Note on the Use of Principal Components in Regression, Journal of the Royal Statistical Society, 17/08/2010 12:50.

[146] Heidi Artigue & Gary Smith | (2019) The principal problem with principal components regression, Cogent Mathematics & Statistics, 6:1, 1622190, DOI: 10.1080/25742558.2019.1622190

[147] An Introduction to Partial Least Squares Regression, Randall D. Tobias, SAS Institute Inc., Cary, NC.

[148] Lars Snipen, A review of variable selection methods in Partial Least Squares Regression, Chemometrics and Intelligent Laboratory Systems.

[149] On the Adaptive Elastic-Net with a Diverging Number of Parameters , Article in The Annals of Statistics · August 2009

[150] Joseph O Ogutu* , Torben Schulz-Streeck, Hans-Peter Piepho. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. 19-20 May 2011.

[151] Chapter 8 : linear and nonlinear regression models , lang WU and Jin QIU , book applied multivariate statistical analysis and related topics with R

[152] Ron S. Kenett and Chris Gotwalt , Functional Data Analysis and Nonlinear Regression Models: An Information Quality Perspective, , Research Triangle, NC, USA.

[153] Wind Speed Prediction Model Using LSTM and 1D-CNN , Rui Fukuoka, Hiroshi Suzuki, Takahiro Kitajima, Akinobu Kuwahara and Takashi Yasuno , Journal of Signal Processing,July 2018

[154] One-Dimensional CNN Approach for ECG Arrhythmia Analysis in Fog-Cloud Environments, OMAR CHEIKHROUHOU , REDOWAN MAHMUD , RAMZI ZOUARI , MUHAMMAD IBRAHIM, ATEF ZAGUIA , AND TUAN NGUYEN GIA , July 16, 2021, date of current version July 28, 2021.

[155] Deep Learning for Event-Driven Stock Prediction , Xiao Ding, Yue Zhang , Ting Liu , Junwen Duan , Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence.

[156] Stock price prediction using LSTM, RNN and CNN-sliding window model Conference Paper · September 2017.

[157] Dehbi, A., Mourad, A. H. I., & Bouaza, A. (2011). Ageing effect on the properties of tri-layer polyethylene film used as greenhouse roof. Procedia Engineering, 10, 466-471.

[158] Djakhdane, K., Dehbi, A., Mourad, A. I., Zaoui, A., &Picuno, P. (2016). The effect of sand wind, temperature and exposure time on tri-layer polyethylene film used as greenhouse roof. Plastics, Rubber and Composites, 45(8), 346-351.

[159] Briassoulis, D. (2006). Mechanical behaviour of biodegradable agricultural films under real field conditions. PolymerDegradation and Stability, 91(6), 1256-1272.

[160] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deeplearning. nature, 521(7553), 436-444.

[161] Feng, S., Zhou, H., & Dong, H. (2019). Using deep neural network with small dataset to predict material defects. Materials & Design, 162, 300-310.

[162] H. Zheng, L. Fang, M. Ji, M. Strese, Y. Ozer, E. Steinbach, Deep learning for surface material classification ¨ using haptic and visual information, IEEE Transactions on Multimedia 18 (12) (2016) 2407–2416.

[163] Y.-J. Cha, W. Choi, O. B¨uy¨uk¨ozt¨urk, Deep learning-based crack damage detection using convolutional neural networks, Computer-Aided Civil and Infrastructure Engineering 32 (5) (2017) 361–378.

[164] X. Li, Z. Yang, L. C. Brinson, A. Choudhary, A. Agrawal, W. Chen, A deep adversarial learning methodology for designing microstructural material systems, in: ASME 2018 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, 2018, pp. V02BT03A008–V02BT03A008

[165] A. Cecen, H. Dai, Y.C. Yabansu, S.R. Kalidindi, L. Song, Material structure-property linkages using threedimensional convolutional neural networks, Acta Mater. 146 (2018) 76–84.

[166] [Dehbi 2014] Dehbi, A., Mourad, A. H. I., Djakhdane, K., & Hilal-Alnaqbi, A. (2015). Degradation of thermomechanical performance and lifetime estimation of multilayer greenhouse polyethylene films under simulated climatic conditions. Polymer Engineering & Science, 55(2), 287-298.

[167] Gulli, A., Pal, S.: Deep Learning with Keras. PacktPublishing Ltd., Birmingham (2017).

[168] Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830 (2011)

[169] RAY, Susmita. A quick review of machine learning algorithms. In : 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, 2019. p. 35-39.
[170] KINGMA, Diederik P. et BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[171] Brzezinski, D., &Stefanowski, J. (2017). Prequential AUC: properties of the area under the ROC curve for data streams with concept drift. Knowledge and Information Systems, 52(2), 531-562.
[172] Calders, T., &Jaroszewicz, S. (2007, September). Efficient AUC optimization for classification. In European Conference on Principles of Data Mining and Knowledge Discovery (pp. 42-53). Springer, Berlin, Heidelberg.
[173] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science & engineering, 9(3),90-95.
[174] Aid, L., Loudini, M., &Hidouci, W. K. (2011). An Admission Control Mechanism for Web Servers using Neural Network. International Journal of Computer Applications, 15(5), 14-19.
[175] Hahm, N., & Hong, B. I. (2004). An approximation by neural networkswith a fixed weight. Computers &Mathematicswith Applications, 47(12), 1897-1903.
[176] Zainuddin, Z., & Pauline, O. (2008). Function approximation using artificial neural networks. WSEAS Transactions on Mathematics, 7(6), 333-338.
[177] Nguyen, T. H., Nong, D., &Paustian, K. (2019). Surrogate-based multi-objective optimization of management options for agricultural landscapes using artificial neural networks. EcologicalModelling, 400, 1-13.