UNIVERSITY *"MEDITERRANEA"* OF REGGIO CALABRIA
DEPARTMENT OF LAW, ECONOMICS AND HUMAN SCIENCES
Ph.D. in Law and Economics - XXXIV Cycle
& Doctoral Europaeus
Faculty of Business and Law – University of Portsmouth

Thesis in Mathematical Methods for Economics, Actuarial and Financial Sciences (Primary) and Applied Economics (Secondary)
S.S.D.: SECS-S/06 - SECS-P/06

Advances on Dynamics Modeling: New issues on Mathematical Programming, Artificial Intelligence and Covid-19 transmission forecasting

Ph.D. Candidate

Dr. Tiziana Ciano

Scientific Supervisor: ***Prof. Massimiliano Ferrara***

Law and Economics Ph.D. Coordinator: ***Prof. Attilio Gorassini***

Tutor: ***Prof. Massimo Finocchiaro Castro***

Accademic Year 2020/2021

Dietro ogni traguardo c'è una nuova partenza.
Dietro ogni risultato, un'altra sfida.

Madre Teresa di Calcutta

Abstract

This research work focuses on the study of dynamic modeling and the role that uncertainty plays within the various complex decision-making problems. Complexity theory deals with analyzing the computational complexity of an algorithm. The main goal was to elaborate the most efficient (to our best knowledge) algorithms to solve the proposed problems. Therefore, the use of innovative Machine Learning techniques allows you to automate the learning process and autonomously provide scenarios and solutions to support decision-making processes. The value and reliability of data-based decisions depend on the quality of the data, the correct analysis of the same and the right interpretations. However, this cannot happen in the absence of a correct use of Big Data and Machine Learning techniques.

The present Ph.D. thesis developed jointly with an international research group focuses on the development of mathematical optimization models and new Machine Learning and Deep Learning algorithms for making predictions and analyzing social dynamics, in particular pandemic and financial issues. The obtained results have shown the efficiency and robustness of the new approaches to address the conditions of uncertainty.

In this work new findings were gained in the frame of (fuzzy) fractional calculus, stochastic programming, mathematical programming, machine and deep learning. A fruitful and rigorous combination among mathematics and artificial intelligence.

Keywords: *Dynamics, Dynamical Systems, Fractional Calculus, Stochastic Programming, Mathematical Programming, Epidemics Modeling, Machine Learning and Deep Learning, Forecasting Algorithms.*

AMS 2020 Classification:

34A07 - *Fuzzy differential equations*
34A08 - *Fractional differential equations*
65K05 - *Mathematical Programming*
98C15 - *Stochastic Programming*
90C29 - *Multi-Objective and Goal Programming*
68T01 - *Artificial Intelligence*
68T05 - *Learning and Adaptive Systems*
68T50- *Natural Language Processing*
91G10 - *Portfolio Theory*
91G80 - *Financial Applications*
92D30 – *Epydemiology*

Index

Preface

In the language of Physics and modern Mathematics (in general also for the Social Sciences with some exceptions), the condition of uncertainty covers over the prediction of a dynamic system governed by deterministic laws. Although dynamic evolution seems to assume a random aspect, since these systems are based on non-linear laws, it is precisely these laws that allow their study and classification through the mathematical platforms. The basic accuracy of the Knowledge of the initial state of the System determines the unpredictability, i.e. the impossibility of determining, after a finite time, its future state. Chaos essentially comes in two forms. In "Conservative Systems" - of which Hamiltonian ones are an important subset, the volume in phase space is conserved during the temporal evolution of the system. In "Dissipative Systems," the volume decreases due to irreversible processes, such as e.g. viscosity or thermal conduction in the case of Quantum Physics. For both classes of systems, Chaos is the result of successive elongations and bending of the volume elements of the phase space.

These technical aspects, which specifically concern the theory of Dynamical Systems, represented the initial reflective starting point that inspired the choice to direct my doctoral studies to the deepening of various facets of "Dynamics". In this context which is obviously very broad and generalist in its essence, not only sematic, it was intended to place some original results that were obtained - in particular - from the second year of the Ph.D starting from the study and research stay at the University of Portsmouth in the first quarter of 2020.

During this period, studies relating to stochastic programming were launched, in particular with the development of an original two-stage model with application in the energy market. The **Chapter 5** collect the original results obtained in this direction and published in the Journal Energy Economics. Specifically, the paper deals with the energy procurement and economic management problem for an aggregation of prosumers at a strategic/tactical level. This decision process, usually in charge of the "aggregator", the entity which coordinates market operations and resource management for the entire coalition, consists into the definition of the optimal mix of energy to procure from the available sources (bilateral contracts, self-production, day-ahead market) and the tariff scheme to offer to the members of the coalition for buying and selling energy. This problem is made more complex by the presence of several sources of uncertainty, like market prices, overall demand of the coalition and production from renewable systems. To the best of our knowledge, even if several contributions have been proposed to deal with the energy procurement and tariff definition problems separately, none of them has addressed the decision process as a whole. In this paper we propose a multiperiod 2-stage stochastic programming approach, which models the strict relations between the decisions to be made and controls risk exposure by a Mean-Risk objective function with the Conditional Value at Risk as risk measure.

During the same period a terrible and disruptive event has been emerging: Coronavirus pandemic. I was involved into an international Research Group promoted and scientifically coordinated by my Supervisor Prof. Dr. Massimiliano Ferrara. Two international projects were launched, by titles:

- *Dynamics of transmission and control of Covid-19: A new mathematical modelling and numerical simulation.*
- *Spectral solution for fractional dynamical models with non- singular kernel derivative under interval arithmetic: Applications to Applied Sciences.*

Therefore, two lines of research were defined giving a scientific curvature to the present thesis stimulating my interests on new fields as:

1. Dynamical Systems and new models to analyze the pandemic dynamics;
2. Artificial Intelligence and (in particular) Machine Learning & Deep Learning issues to develop new algorithms for Sentiment Analysis and Stage-wise Classification & Prediction.

In one year three papers were accepted and published on three high impact Journals as: *Advances on Difference Equations, Computers and Electrical Engineering and Expert Systems.*

In the first paper, a novel coronavirus infection system with a fuzzy fractional differential equation defined in Caputo's sense is developed. By using the fuzzy Laplace method coupled with Adomian decomposition transform, numerical results are obtained for better understanding of the dynamical structures of the physical behavior of Covid-19. Such behavior on the general properties of RNA in Covid-19 is also investigated for the governing model. The results demonstrate the efficiency of the proposed approach to address the uncertainty condition in the pandemic situation. The essence of the paper was organized into the **Chapter 4**.

In the second paper some Sentiment Analysis new issues were studied, in particular related to Fake news dynamics. "Fake news" refers to the misinformation presented about issues or events, such as Covid-19. Meanwhile, social media giants claimed to take Covid-19 related misinformation seriously, however, they have been ineffectual. This research uses the Information fusion to obtain real news data from News Broadcasting, Health, and Government websites, while Fake News data are collected from social media sites. 39 features were created from multimedia texts and used to detect fake news regarding Covid-19 using state-of-the-art deep learning models. Our model's fake news feature extraction improved accuracy from 59.20% to 86.12%. Overall high precision is 85% using the Recurrent Neural Network (RNN) model; our best recall and F1-Measure for fake news were 83% using the Gated Recurrent Units (GRU) model. Similarly, precision, recall, and F1-Measure for real news are 88%, 90%, and 88% using the GRU, RNN, and Long Short-Term Memory (LSTM) model, respectively. Our model outperformed standard machine learning algorithms.

As we all know, the Covid-19 pandemic has sparked global apprehensions over the increase in deaths and raised concerns about gaps in health infrastructure and accessibility around the world. Consequently, the importance of timely prediction and treatment of the disease to reduce transmission and mortality rates cannot be

emphasized enough. Various symptoms of the disease have been identified as it progresses from the time it is contracted. Covid-19 has been found to internally affect the lungs, and the four progressive stages of the infection can be categorized as mild, moderate, severe, and critical. An accurate analysis of the current stage of the disease that can help predict its progression has become critical. Therefore, in the third paper an effective screening procedure - X-ray imaging - is studied to predict the various phases of this epidemic.

Although many different approaches using machine learning, as well as deep learning were utilized to predict and classify diseases in general, till date, such an approach has not been used to predict the various stages of Covid-19 by using X-ray imaging to identify and classify those stages.

This research finds its place in **Chapter 8** of the thesis which describes the hybrid method and the tools used. The proposed hybrid model used three public datasets for its implementation. In this work, extensive images were used for the purposes of testing and training. The dataset-1 consists of 1200 Covid-19 as well as 1200 Non-Covid-19 images, while dataset-2 used 700 Covid-19 as well as 700 Non-Covid-19 images, and finally, dataset-III utilized 1900 Covid-19 as well as 1900 Non-Covid-19 images for purposes of testing and training. The proposed work undertook the task of pre-processing using textual and morphological features, while the segmentation and prediction of Covid-19 as well as Non-Covid-19 images were undertaken using VGG-16 with light GBM for better prediction and handing of huge datasets, and finally, the classification of the various stages of Covid-19 images was performed using Deep Belief Network. Furthermore, as will be illustrated in the same Chapter as well, the results of the proposed work were subjected to several iterations which were then compared using different parameters such as accuracy, specificity and sensitivity. In general, the prediction and grouping of the various stages of Covid-19 by using affected images were found to be 99.2%, 99.4%, and 99.5%, respectively. The bacterial pneumonia prediction rates were observed to be 98.5%, 99.4%, and 98.3 %, respectively. The average classification of the stages were found to be 98.1%, 98.6%, and 98.3%, while the combined multi-classification prediction rates were observed to be 98.6%, 99.1%, and 98.7%, respectively.

As previously stated, my Doctoral Studies were aimed at deepening the "Dynamic Modeling" and the role that uncertainty plays within each problem that is described in the various original works accompanying the thesis. In this regard, two scientific papers have been written (including one in the publication phase) in which the portfolio optimization problems are studied using new mathematical models and Machine Learning and Deep Learning algorithms. This study is conducted by referring not only to data concerning the equity and bond market but also to the market of virtual or digital currencies used in today's financial systems, namely Cryptocurrencies (bitcoin is the best known). In particular, in **Chapter 6**, we propose a tri-objective portfolio optimization model comprising three objectives, which in addition to return, risk, model the preferences of decision makers using a proposed composite index. In earlier studies, decision-maker preferences modelled using practical constraints; in

contrast, this paper modelled these preferences as constraints along with the proposed composite index based on three decision parameters. The first decision parameter represents an equilibrium between return and variance. The second decision parameter is related to the selection of proportional weights for sub-portfolios in the proposed optimization model. The third decision parameter is used for assigning an appropriate level for sustainable loss values in the portfolio selection problem. An upper bound is also formulated for this composite index using a weighted sum approach for decision parameters. The proposed third objective is suitable for an investor for choosing desired levels of the return, risk, amount of capital allocated to sub-portfolios along with an appropriate level of loss. To check the effectiveness of the proposed approach is tested on four Multi-Objective Evolutionary Algorithms i.e. NSGA-II, SPEA2, MOPSO, and MOEA/D. Finally, conclusions are drawn from the comparative study of these adapted MOEAs**.**

As previously mentioned, a further research study has affected the virtual currency market, in fact, Cryptocurrency represents a new type of digital asset that has evolved as a result of technological financial advances and has provided a significant research opportunity. There are many algorithms for price prediction for Cryptocurrencies like LSTM and ARIMA. However, the downside is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior is tough. In order to produce decent outcomes, rigorous hyperparameter adjustment is also essential.

Furthermore, Cryptocurrencies do not precisely adhere to past data, and patterns change fast, reducing the accuracy of predictions. Cryptocurrency price forecasting is difficult due to price volatility and dynamism. Because the data is dynamic and heavily influenced by various seasons, the ARIMA model is unable to handle seasonal data. In order to provide better price predictions for crypto traders, a new model is required. Therefore, the in-depth information on this issue is placed in the last chapter of the thesis.

The objective of the study is to apply Fbprophet model as the key model because it is superior in functionality as compared to LSTM and ARIMA additionally removing the pitfalls generated in LSTM and ARIMA model while analyzing the Cryptocurrency data. This study provides a methodology for predicting the future price of bitcoin that does not rely solely on past data due to seasonality in historical data.

So, after fitting the seasonality and smoothing, the model is constructed that can be useful for realworld use cases. In case of Cryptocurrencies where less historical data is available and it is hard to find pattern, proposed method can easily deal this type of problems. Overall difference between predicted and actual values is low as compared to other model even after seasonal data was available.

***

The scientific core of the present Ph.D. Dissertation is mainly a collection of the new findings contained in the following already published papers:

1. Ciano, T. *et al.* (2020). Fuzzy fractional-order model of the novel coronavirus. *Advances in difference equations*, *2020*(1), 1-17. **ANVUR Scientific Journal - Area 13. Scopus: s2.0-850902859151, WoS (Web of Science, Clarivate): 000569802600002.**

2. Ciano, T. *et al.* (2021). An integrated decision approach for energy procurement and tariff definition for prosumers aggregations. *Energy Economics*, *97*, 105034. **ANVUR Scientific Journal - Area 13, FASCIA A SSD 13/D4. Scopus: s2.0-85097039228, WoS: 000645464600023.**

3. Ciano, T. *et al.* (2021). A hybrid learning approach for the stage-wise classification and prediction of COVID-19 X-ray images. *EXPERT SYSTEMS*. **ANVUR Scientific Journal - Area 13. Scopus: s2.0-851209376604, WoS: 000729539200001**.

4. Ciano, T. *et al*. (2022). Covid-19 fake news sentiment analysis. *Computers and electrical engineering*, *101*, 107967. **ANVUR Scientific Journal - Area 13. Scopus: s2.0-85129284181, WoS: 000797942900001**.

5. Ciano T. *et al.* (2022). Real-world model for bitcoin price prediction. *Information Processing & Management*, *59*(4), 102968. **ANVUR Scientific Journal Area 13 - FASCIA A SSD 13/D4. Scopus to be assigned, WoS to be assigned.**

For that concerns the new findings not yet published, it was arranged into the following ongoing research:

6. Ciano T., *et. al.* (2022). A constrained portfolio optimization framework with a tri-objective design using multi-objective evolutionary algorithms and decision parameters.

These results were achieved thanks under the scientific guidance of ***Prof. Massimiliano Ferrara.***

Part I

Uncertainty and Dynamics: a mathematical background

CHAPTER 1

Fuzzy Fractional Differential Equations: some tools

1.1 Introduction

For a long time, differential difference and integral equations (in the sequel named as Systems), have been implemented to modeling some real world issues in which dynamics is the core.

Some mathematical model as SI, SIR, SEIR, HIV with logistic models have been used in studying epidemiological phenomena and Socio-economical Sciences. These Systems have been analysed for series, analytical, semi – analytical and numerical solution to evaluated their convergence and stability by specific mathematical techniques.

Since the begining of this century the field of calculus as arbitrary order differential (and anti-differential) calculus have gained a large interest by a lot of Scientists and Scholars.

By my Dissertation, I am going to study in a deep way, one sub – field of this interesting frame of differential/difference equations/systems, named as “Fractional Calculus” have more precise and concrete results in many and various fields like Physics Engineering, Biological Sciences and Economics.

Riemann and *Liouville*, *Eulero* and *Fourier* around 1750s introduced a milestone as analytical results into integer – order of differential and integral calculus. Starting from results whose began the field of Fractional Calculus and the focuced studies related to real world problems were modeled by using fractional – order differential and integral equation in a better way whit respect the integer order differential equations, partial differential equations and integer differential equation.

Practically, Fractional Calculus have reduced many of the errors wich normaly accured in calculus getting more precise mathematical models.

Some useful example of Fractional Calculus models can be in [1, 2, 4, 11-18]. The Researcher has studied Fractional Calculus very carefully. Therefore non-integer order derivative is an anti-derivative of a definite type, that is the summation of the spectrum or entire function, which make it globalized and generalized.

Scholars have obtained satisfactory results by analyzing the theory of existence, approximation, uniqueness results, optimization and stability analysis of the mathematical models. Thus, a differential operator of arbitrary order can be formulated in several ways.

A formula that has currently gained more interest, in 2016, is the non-integer derivative "ABC" defined by "*Atangana-Baleanu*" and "*Caputo* [19]. This arbitrary order derivative changed the “singular kernel” by “non-singular kernel” and because of this, it is studied on high-level [20, 21]. Several pertubation and decomposition techniques

were used to address problems for analytical results [22, 23]. Moreover, in the last few years generalized methods of the *Taylor* series and stands for *Runge Kutta* methods have been used, for the numerical results. Also, the ABC method (powerful, convergent and stable method, [see 26-27]) was used for the numerical results. Fractional Calculus has many applications in the fields where data are not precise [28 -31]. In 1965 *Zadeh* [32] introduced the concept of Fuzzy Set (that how to measure uncertainty present in certain phenomena), extended to many other branches like, Algebra, Topology, Fuzzy Logic, Analysis, Automata and many more that how to measure uncertainty present in some phenomena. The author, further elaborate the concept by defining fuzzy-function and control [33]. From these results, some scholars have extended this concept by introducing elementary Fuzzy Calculus [34-36]. Recently, the *Fuzzy Fractional Differential Equations* (FFDEs) and *Integral Equations* (FFIEs) have had much interest, in particular for their applicability in formulating real society phenomena. Some fundamental results of numerical analysis have been studied by Esmail *et al*. in [37], using the *Fuzzy Fractional Differential Equations* and the *Fuzzy Fractional Integral Equations* for these types of problems.

In fact, many Scientists consider it to analyze whole and Fractional Fuzzy Differential Equations to model uncertain phenomena in various branches, for example business, biological and physical sciences etc. [see 28–31]. To analyze FFDEs different techniques have been used such as the Fourier integral transform, Sumudu, Laplace and Natural transform etc. Laplace transforms are used to solve the differential and integral equations [see 38-43]. The Sumudu transform was introduced in 1993 and then extended to two variables by the same author in 2002 [44]. Both transforms also have many uses in the solutions of various types of FEDs. The natural transform was introduced in 2008, initially used for the evaluation of fluid flowing situations of *Maxwell*'s equations and subsequently also used in the solutions of ODEs [5-7, 45, 46]. Unlike the Laplace and Sumudu transforms, the NT has the ability to quickly converge only by varying the parameters.

Adomian decomposition and Laplace Adomian decomposition methods, Taylor's series (interative and series solution methods) they are also used to deal with such problems. Since the natural transform has different applications, in my thesis, I will evaluate the semi-analytical solution of the given non-integer two-dimensional fuzzy wave equation under the Caputo fractional derivative by repeated iterative numerical techniques applying NT together with perturbation techniques

$$\begin{aligned} &\mathcal{D}_t^{\theta}\tilde{\mathcal{U}}(x,y,t,r) = \mathcal{D}_x^2\tilde{\mathcal{U}}(x,y,t,r) + \mathcal{D}_y^2\tilde{\mathcal{U}}(x,y,t,r) + \tilde{\mathcal{F}}\,(x,y,t,r), \qquad 1 < \theta \le 2, \\ &\tilde{\mathcal{U}}(x,y,0) = \tilde{\mathcal{G}}_1(x,y), \\ &\tilde{\mathcal{U}}_t(x,y,0) = \tilde{\mathcal{G}}_2(x,y), \end{aligned} \tag{1}$$

here $\tilde{\mathcal{G}}_1, \tilde{\mathcal{G}}_2$ are continuous fuzzy-valued functions, D represent Caputo fractional

derivative and external source. Different methods like Laplace, Fourier, Samudu, and Natural transforms, etc. [47–59] have been used to investigate wave and various differential equation.
We consider this *case study* studied in "*A novel semi-analytical method for solutions of two dimensional fuzzy fractional wave equation using natural transform*" by *Ferrara et al.* [60] in which the authors involved some wave equations in terms of Caputo fractional derivative and in general fractional calculus. These aspects are crucial to better understand the new model discussed in the II part of my work.

1.2 Preliminars and tools

Definition 1. ([61, 62]) Let $\mu : \mathbb{R} \to [0,1]$ be a fuzzy set of the real line satisfying the following properties:

1. μ is normal (for any $a_0 \in \mathbb{R}$; $\mu(a_0) = 1$);
2. μ is upper semicontinuous on $\mathbb{R}\,(\forall \varepsilon > 0\, \exists\, \delta > 0\, |\mu(a) - \mu(a_0)| < \varepsilon, |a - a_0| < \delta)$;
3. μ is convex $\big(\mu(\kappa a + (1-\kappa)b) \geq \big(\mu(a) \wedge \mu(b)\big) \forall \kappa \in [0,1], a, b \in \mathbb{R}\big)$;
4. $cl\{a \in \mathbb{R}, \mu(a) > 0\}$ is compact.

Then it is called a fuzzy number.

Definition 2. ([61]) On a fuzzy number μ, the p-level set is defined by
$[\mu]^p = \{x \in \mathbb{R} : \mu(x) \geq p,$
where $p \in (0,1]$ and $x \in \mathbb{R}$.

Definition 3. ([61, 62]) Let $[\underline{\mu}(\vartheta), \underline{\bar{\mu}}(\vartheta)]$ be the parametric form of a fuzzy number μ, where $0 \leq \vartheta \leq 1$, which satisfies the following properties:

1. μ(ϑ) is left continuous, bounded, and increasing function over (0, 1], and right continuous 0.
2. $\bar{\mu}(\vartheta)$ is right continuous, bounded, and decreasing over [0, 1], and right continuous at 0.
3. μ(ϑ) ≤ $\bar{\mu}(\vartheta)$.

Also, if μ(ϑ)= $\bar{\mu}(\vartheta)$= 0, then ϑ is called a crisp number.

Definition 4. ([63]) Consider a mapping $\rho : E \times E \to \mathbb{R}$ and let $v = \big(\underline{v}(\vartheta), \bar{v}(\vartheta)\big)$ *e* $w = (\underline{w}(\vartheta), \bar{w}(\vartheta))$ be two fuzzy numbers in their parametric form. The Hausdorff distance between v and w is defined by

$$\rho(v,w) = \sup_{\vartheta\in[0,1]} \left[max\left\{\left|\underline{v}(\vartheta) - \overline{w}(\vartheta)\right|, \left|\overline{v}(\vartheta) - \overline{w}(\vartheta)\right|\right\}\right]$$

In E, the metric ρ has the following properties:

$$\rho\,(v\, +\, v, w\, +\, v) =\, \rho(v,w) \text{for all } v, v, w\, \in\, E$$

$$\rho(v\varrho\,, w\varrho)\, =\, |\varrho\,|\rho(v,w) \text{for all } v, w\, \in\, E, \varrho\, \in\, \mathbb{R};$$

$$\rho(v\, +\, \xi\,, w\, +\, \varsigma) \leq\, \rho(v,w) +\, \rho(\xi\,, \varsigma) \text{ for all } v, w, \xi\,, \varsigma\, \in\, E;$$

(E, ρ) is a complete metric space.

Definition 5. ([63]) Let $\tau_1, \tau_2 \in E$. If there exist $\tau_3 \in E$ such that $\tau_1 = \tau_2 + \tau_3$ then τ_3 is said to be the H-difference of $\tau_1\, and\, \tau_2$, denoted by $\tau_1 \ominus \tau_2$.

Definition 6. ([63]) Let $\Theta : \mathbb{R} \rightarrow E$ be a fuzzy mapping. Then Θ is called continuous if for any $\epsilon > 0\, \exists\, \delta > 0$ and a fixed value of di $\lambda_0 \in [\zeta_1, \zeta_2]$, we have

$\rho\, \Theta(\lambda), \Theta(\lambda_0) < \epsilon$ whenever $|\lambda - \lambda_0\,| < \delta$.

Definition 7. ([63, 64]) Let Φ be a continuous fuzzy function on $[0, b] \subseteq \mathbb{R}$, a fuzzy fractional integral in Riemann–Liouville sense corresponding to t is defined by

$$I^k\Phi(t) = \frac{1}{\Gamma(k)} \int_0^t (t-\zeta)^{k-1}\, \Phi(\zeta\,) d\zeta, \qquad \text{where } k\,, \zeta\, \in\, (0, \infty).$$

Further, if $\Phi\, \in\, C^F\,[0, b]\, \cap\, L^F\,[0, b]$, where $C^F[0, b]\, and\, L^F\,[0, b]$are the spaces of fuzzy continuous functions and fuzzy Lebesgue integrable functions, respectively, then fuzzy fractional integral is defined as

$$[I^k\phi(t)]_p = \left[I^k\underline{\phi}_p(t), I^k\bar{\phi}_p(t)\right],\ \ 0 \leq\, p \leq 1,$$

where

$$I^k\underline{\phi}_p(t) = \frac{1}{\Gamma(k)}\int_0^t (t-\zeta)^{k-1}\underline{\phi}_p(t)d\zeta, \quad k, \zeta \in (0, \infty)$$

$$I^k\bar{\phi}_p(t) = \frac{1}{\Gamma(k)}\int_0^t (t-\zeta)^{k-1}\bar{\phi}_p(t)d\zeta, \quad k, \zeta \in (0, \infty)$$

Definition 8. ([63]) If a fuzzy function $\Phi \in C^F[0,b] \cap L^F[0,b]$ is such that $\phi = [\underline{\phi}_p(t), \bar{\phi}_p(t)], 0 \leq p \leq 1$ and $t_1 \in (0,b)$, then the fuzzy fractional Caputo's derivative is defined as

where

$$D^{\beta}\underline{\phi}_p(t_0) = \frac{1}{\Gamma(n-\beta)}\left[\int_0^t (t-\zeta)^{n-\beta-1}\frac{d^n}{d\zeta^n}\underline{\phi}_p(\zeta)d\zeta\right]_{t=t_0}$$

$$D^{\beta}\bar{\phi}_P(t_0) = \frac{1}{\Gamma(n-\beta)}\left[\int_0^t (t-\zeta)^{n-\beta-1}\frac{d^n}{d\zeta^n}\bar{\phi}_P(\zeta)d\zeta\right]_{t=t_0}$$

whenever the integrals on the right-hand sides converge and $n = [\beta]$.

Definition 9. ([63, 65, 66]) Let Φ be a continuous fuzzy-valued function. Assume that $\Phi(\chi)\cdot e^{-s\chi}$ is an improper fuzzy Riemann-integrable on $[0,\infty)$, then its fuzzy Laplace transform is represented by

$$[\phi(\chi] = \int_0^{\infty}\phi(\chi)\cdot e^{-s\chi}\,d\chi$$

For $0 \leq r \leq 1$, the parametric form of $\Phi(\chi)$ is represented by

$$\int_0^{\infty}\phi(\chi,r)\cdot e^{-s\chi}d\chi = \left[\int_0^{\infty}\underline{\phi}(\chi,r)\cdot e^{-s\chi}d\chi, \int_0^{\infty}\bar{\phi}(\chi,r)\cdot e^{-s\chi}d\chi\right]$$

Hence,

$$L[\phi(\chi,r)] = \left[L\underline{\phi}(\chi,r), L\overline{\phi}(\chi,r)\right]$$

Theorem 1. ([63]) *Let* $\phi \in C^F[0,b] \cap [0,b]$ *then for* $0 \leq p \leq 1$, *and* $0 < \beta \leq 1$, *the Laplace transform of fuzzy fractional derivative in Caputo's sense is given by*

$$L\left[(D^{\beta}\phi(t))_p\right] = s^{\beta}L[\phi(t)] - s^{\beta-1}[\phi(0)].$$

CHAPTER 2

Stochastic and Multi-Objective Programming: an useful introduction

2.1 Introduction

Stochastic Programming handles complex mathematical optimization issues in which unknown variables create a series of available solutions. This may involve the adoption of a model through a series of phases, each of which can be influenced by separate variables. Mathematicians can apply it to problems related to decision-making, resource allocation and so on. It is also the subject of scientific study, in which researchers work on developing new and more effective Stochastic Programming models to apply to real-world situations.

Stochastic Programming must be flexible to reach the optimal solution, while imposing a certain order on decisions to allow for their quantification in a quantitative problem. The level of complexity may depend on the nature of the problem; some are simply arranged in two stages, while others may involve multiples. For each phase it is possible to determine the optimal solution and consider the impact it will have on the decision-making process along the line.

Therefore, Multi-Objective optimization is an area of multiple decision criteria that deals with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-Objective optimization has been applied in many fields of Science, including Engineering, Economics, and Logistics, where optimal decisions must be made when there are trade-offs between two or more conflicting objectives.

In particular, Multi-stage Stochastic Programming has attracted a lot of researchers in the last two decades, as a featured framework for stating decision making problems under uncertainty. “Uncertainty” may refer to the evolution of the demand for goods or services, temperature and rainfall patterns affecting consumption or production, interest rates affecting the burden of debt.

Under growing environmental stress, resource limitations, concentration of populations in cities, many believe that these applications can only get a higher societal impact in the future, and that even better quantitative methods for tackling them are needed, especially methods able to take into account a large number of constraints.

In this Chapter, we introduce the mathematical models of multi-stage stochastic programs, by analyzing the different risk measures and in particular the multi-period CVaR model. Finally, Multi-Objective Evolutionary Algorithms (MOEAs) are studied. Multi-Objective Evolutionary Algorithms were originally proposed in the mid-1980s, but it was until the mid-1990s that they began to attract researchers'

interest. MOEAs have attracted a lot of research efforts over the past 20 years and are still one of the hottest research areas in the field of evolutionary computing. Today, we have a wide variety of algorithms and research in this area has become highly specialized. Therefore, Multi-Objective Evolutionary Algorithms are any of the paradigms of evolutionary computation (e.g. genetic algorithms, evolutionary strategies, etc.) used to solve problems that require the optimization of two or more potentially conflicting objectives, without resorting to reduction objectives to a single objective by means of a weighted sum. In recent years, evolutionary algorithms have become a popular choice for solving highly complex multi-objective optimization problems. These algorithms will be analyzed in more detail in Chapter 6.
A further study on multi-period optimization models will be analyzed in Chapter 5 in which we propose a two-phase multi-period Stochastic Programming approach, which models the close relationships between decisions to be made and controls risk exposure through a medium risk objective function with the Conditional Value at Risk as a measure of risk.

2.2 Multi-stage stochastic programs

We first formulate a linear multi-stage stochastic program with an expectation in the objective function. We suppose that the model has random parameters in stage $t = 2, \dots, T$, denote $\xi_t = (c_t, A_t, B_t, b_t)$, which are governed by a known, or well-estimated, distribution. The parameters of the first stage, $\xi_1 = (c_1, A_1, b_1)$ are assumed to be known when we make decision x_1, but only a probability distribution governing future realizations, $\xi_2, \dots, \xi_T$ is known. The realization of ξ_2 is known when decisions x_2 must be made and so on up to stage T.
The term $B_t x_{t-1}$ captures the state of the system. For example, in an asset allocation model, the state could be the total value of the assets in our portfolio. The components $x_2, \dots, \xi_T$ of ξ and the decisions $x_2, \dots, x_T$ are assumed to be random vectors, not necessarily of the same dimension, defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, while ξ_1 is deterministic and x_1 is a nonrandom vector-valued variable. The sequence of decisions and observations is

$$x_1, \xi_2, x_2(x_1, \xi_2), \dots, x_T(x_{T-1}, \xi_2, \dots, \xi_T) \qquad (1)$$

The decision process is non-anticipative which means that decisions taken at any stage of the process depend neither on future realizations of stochastic data nor on future decisions, whereas the past information as well as the knowledge of the probability distribution of the data process can be exploited. In a mathematical way, let $\mathcal{F}_t \subseteq \mathcal{F}$ be the σ-field generated by the projection $\Pi_t \xi = \xi_{[t]} \coloneqq (\xi_1, \dots, \xi_t)$ of the stochastic data process ξ that includes data up to stage $t, \mathcal{F}_1 = \{\emptyset, \Omega\}$ is the trivial σ-field. The dependence of the t-th stage decision x_t only on the available information means that

x_t is $\mathcal{F}_t$ −measurable. Similarly we let $\Pi_t x = x_{[t]} \coloneqq (x_1, \dots, x_t)$ denote the sequence of decisions at stages $1, \dots, t, \mathbb{P}$ the probability distribution of ξ, $\mathbb{P}_t$ denotes the marginal probability distribution of ξ_t, and $\mathbb{P}_t[\cdot\,|\xi_{[t-1]}]$, $t = 2, , \dots, T$, its conditional probability distribution. An illustrative scheme of a scenario tree can be found in Figure 1.

The first stage decisions consist of all decisions that have to be selected before further information is revealed whereas the second stage decisions are allowed to adapt to this information, etc. In each of the stages, the decisions are limited by constraints that may depend only on the previous decisions and observations. Stages do not have to coincide with the observations of the random parameters, they should rather correspond to steps in the decision process. Besides that, time spans between stages do not have to be equal, which facilitates effective representation of problems with far horizon.

The decision process (1) has a random outcome, represented by cost $f(x, \xi)$, and the basic goal is to find a non-anticipative decision or policy $x(\xi)$ which minimizes the expectation $\mathbb{E}_{\mathbb{P}}[f(x, \xi)]$ and satisfies all prescribed constraints. The optimal decision will be denoted x^*.

If there are more optimal solutions, x^* is used to represent any member of the set of optimal solutions. The basic form of a multi-stage stochastic program is the multi-stage stochastic linear program (MSLP).

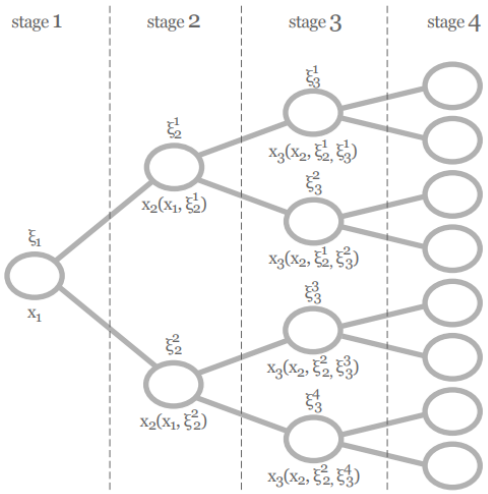


Figure 1: Scenario tree with associated decisions

Overall cost function:

$$\min_{x_1 \in \chi_1} c_1\, x_1 + \mathbb{E}_{\mathbb{P}}\big[Q_2\big(x_1, \xi_{[2]}\big)\big] \;\; with\; \chi_1 \coloneqq \{x_1 | A_1 x_1 = b_1, x_1 \geq 0\}, \tag{2}$$

and $Q_t(x_{t-1}, \xi_{[t]}), t = 2, \dots, T$, defined recursively as

$$Q_t(x_{t-1}, \xi_{[t]}) = \min_{x_t} c_t(\xi_{[t-1]})^T x_t + \mathbb{E}_{\mathbb{P}_{t+1}[\cdot|\xi_{[t]}]}[Q_{t+1}(x_t, \xi_{[t+1]}) \quad (3)$$

$$\text{s.t. } x_t \in \chi_t(x_{t-1}, \xi_{[t]}), \ e.g.$$

$$A_t(\xi_{[t-1]})x_t = b_t(\xi_{[t-1]}) - B_t(\xi_{[t-1]})x_{t-1}, \quad x_t \geq 0 \ \ a.s,$$

and $Q_{T+1}(\cdot)$ is explicitly given, e.g. $Q_{T+1}(\cdot) \equiv 0$.

Matrices A_t are of a fixed (m_t, m_t) type, and the remaining vectors and matrices are of consistent dimensions. For the first stage, known values of all elements of c_1, A_1, b_1 are assumed, and the main decision variable is c_1 that corresponds to the first stage. The first stage problem (2) has the form of the expectation-type stochastic program with the set of feasible decisions independent of P.
One can rewrite (2) – (3) briefly as

$$\min_{x_1} c_1^T x_1 + \mathbb{E}\left[\min_{x_t} c_2 (\xi_{[1]})^T x_2 + \mathbb{E}\left[\dots + \mathbb{E}\left[\min_{x_T} c_T(\xi_{[T-1]})^T x_T\right]\right]\right] \quad (4)$$

with corresponding conditional expectations as in (3) and subject to linear $x_t \in \chi_t(x_{t-1}, \xi_{[t]}, \ t = 1, \dots T$ on decision variables. The set of all feasible decisions $\chi(\xi)$ is given by

$$\{x \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_T} : x_1 \in \chi_1, x_2 \in \chi_2(x_1, \xi_{[2]}), \dots, x_T \in \chi_T(x_{T-1}, \xi_T\}.$$

Constraints involving random elements are supposed to hold almost surely and for simplicity we will assume that all optimal solutions exist. A common assumption of relatively complete recourse requires that for each decision of the stage t, there exists some feasible decision for the stage $t + 1$. Moreover, we will suppose that all conditional expectations exist. In the case of stage-wise independence the conditional probability distributions boil down to marginal distributions $\mathbb{P}_t$ of ξ_t.
In applications one mostly approximates the true probability distribution $\mathbb{P}$ of ξ by a discrete probability distribution carried by a finite number of atoms (scenarios), let me say, $\xi^1, \dots, \xi^K$. They are organized in form of a scenario tree and in principle, the optimal policy can be obtained by solving a large deterministic program. Every node of the tree is a root of a scenario subtree which does not contain any branches of other subtrees. Hence, the optimal solutions of a nodal subproblem do not reflect the future information carried by branches of the full tree that, from the point of view of the relevant nodal subproblem, cannot happen in the future. This observation is behind a

modified time consistency concept.

Remark 1. *At each state of the system, optimality of a decision policy should not involve states which cannot happen in the future.*

Hence, under modest assumptions, optimal solutions of risk-neutral scenario-based multi-stage stochastic programs possess both of these time consistency properties, whereas there are still open questions concerning time consistency notions for risk-averse multi-stage stochastic programs.

2.3 Risk-adverse multi-stage stochastic optimization

Maximization of expected gains or minimization of expected losses means to get decisions that are optimal on average while possible risks are neglected. This need not be an acceptable goal. The present tendency is to spell out explicitly the concern for risk monitoring and control. There are various types of risk and the choice of a suitable risk definition depends on the context, on the decision maker's attitude, the company goals, etc.

To reflect risks in the stochastic programming formulation, it is necessary to quantify them. Both in theoretical considerations and in applications, rational properties of risk measures are requested. A risk measure is a functional which assigns a real value to the random outcome $f(x,\xi)$. Similarly, as the risk-neutral expected value criterion, risk measures ρ should not depend on individual realizations of ξ, but they depend on decisions and probability distribution $\mathbb{P}$. Moreover, they should also reflect the structure of the filtration $\mathcal{F}_1 \subset \cdots \subset \mathcal{F}_t \ldots . \subseteq \mathcal{F}$.

Coherence of ρ (monotonicity, translation invariance, positive homogeneity and subadditivity) is mostly expected. Popular examples of risk measure ρ include Value at Risk (VaR), which is not coherent in general, and the Conditional Value at Risk (CVaR), which belongs to the class of coherent risk measures. Monotonicity with respect to the pointwise partial ordering and subadditivity are straightforward requirements, coming from the principles of risk quantification. Convexity is important to keep a manageable structure of the problem both for computational and theoretical purposes. Polyhedral property, cf. CVaR in Rockafellar & Uryasev [1], or polyhedral risk measures, cf. Eichhorn & Römisch [2], facilitate the application in scenario-based programs with linear constraints, because we can rely on linear programming techniques.

Whereas there are many suggestions of risk measures for static stochastic programs, which have performed well in numerical experiments and applications, the situation is much more involved for multi-stage problems. The first idea is to replace the expectation $\mathbb{E}_{\mathbb{P}}[f(x,\xi)]$ by a suitable risk measure ρ and to keep all constraints including non-anticipativity. Assigning a risk measure ρ to the final outcome $f(x,\xi)$

does not take into account the information structure given by the filtration. It corresponds to monitoring risk only at the horizon which need not be sufficient.
To include risk monitoring in individual stages, one may relate the risk measure to the partial outcomes $f_1(x_1), f_t\big(x_t, x_{[t-1]}, \xi_{[t]}\big), t = 2, \ldots., T$. Different risk measures ρ_t can be applied in individual stages.
As a result we may construct objective function of the form

$$\min_{x_1} c_1^T x_1 + \rho_2 \left(\min_{x_2} c_2 \, (\xi_{[1]})^T x_2 + \rho_3 \left(\ldots. + \rho_T \left(\min_{x_T} c_T \, (\xi_{[T-1]}) x_T \right) \right) \right) \qquad (5)$$

and use it in the place of (4). It is important to agree on acceptable properties of risk measures, usually at least convexity should be satisfied. The full formulation of the risk-averse stochastic program (5) has to include the non-anticipativity constraints. Depending on the risk-averse problem and on the applied solution technique, a form of time consistency of optimal solutions is desirable.

2.4 Mathematical formulation of the basic definition

We will model the risk by representing the loss which could be incurred in stages $1, \ldots, T$ by random functions $Z = (Z_1, \ldots, Z_T)$ that will be defined on a suitable linear space $\mathcal{Z}$. The notion of coherent risk measures was introduced in Artzner *et al*. [3] and is now widely accepted in static risk-averse optimization. Following definition, based on Artzner *et al.* [4] and Eichhorn and Römisch [5], extends this notion to the multi-stage case.

Definition 1. (Multi-period risk measures).
A functional ρ on $\mathcal{Z} = \times_{t=1}^{T} \mathcal{L}_p \, (\Omega, \mathcal{F}_t, \mathbb{P})$ with $p \in [1, \infty]$ is called a multi-period coherent risk measure if it satisfies the following:

1. $Z_t \geq \tilde{Z}_t$ a.s, $t = 1, \ldots, T \Rightarrow \rho(Z_1, \ldots\ldots, Z_T) \geq \rho(\tilde{Z}_1, \ldots\ldots, \tilde{Z}_T)$ (monotonicity);
2. $for\ each\ r \in \mathbb{R}$: $\rho\ (Z_1, \ldots\ldots, Z_T + r) = \rho(Z_1, \ldots\ldots, Z_T) + r$(translation invariance);
3. $\rho\ \big(\mu Z_1 + (1-\mu)\tilde{Z}_1, \ldots\ldots \mu Z_T + (1-\mu)\tilde{Z}_T\big) \leq \ \mu\rho(Z_1, \ldots\ldots, Z_T) + (1 - -\mu)\rho\big(\tilde{Z}_1, \ldots\ldots, \tilde{Z}_T\big) for\ \mu \in [0,1]$ (convexity);
4. $\rho\ (\mu Z_1 \ldots\ldots, \mu Z_T) = \mu\rho\ (Z_1, \ldots, Z_T)\ for\ \mu \geq 0$ (positive homogeneity)

Two special classes of multi-period risk measures have received a lot of attention, polyhedral risk measures and conditional risk mappings. Polyhedral risk measures are defined as the optimal value of a multi-stage stochastic program in the following way [5].

Definition 2. (Multi-period polyhedral risk measures).
A risk measure $\rho \times_{t=1}^{T} \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P})$ with $p \in [1, \infty]$ is called multi-period polyhedral if there are $k_t \in \mathbb{N}, c_t \in \mathbb{R}^{k_t}, t = 1, \ldots, T, w_{t,\tau} \in \mathbb{R}^{k_{t-\tau}}, t = 1, \ldots, T, \tau = 0, \ldots, t-1$, *a polyhedral set* $M_1 \subset R^{k_t}$ $t = 2, \ldots T$, *such that*

$$\rho(Z) = \inf \mathbb{E}\left[\sum_{t=1}^{T} c_t^T Y_t\right]$$

$$\begin{aligned} & s.t \quad Y_t \in \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P}) && \forall t\{1, \ldots, T\} \\ & Y_t \in M_t \; a.s. && \forall \in \{1, \ldots, T\} \\ & \sum_{\tau=0}^{t-1} w_{t,\tau}^T Y_{t-\tau} = Z_t \; a.s && \forall \in \{1, \ldots, T\} \end{aligned} \tag{6}$$

When replacing the expectation of the total outcome of a risk-neutral MSLP by the multi-period polyhedral risk measure it is possible to carry out the minimization with respect to the original decision variable x and minimization in (6) simultaneously.
Moreover, the scenario form of (6) and that of the combined problem is a linear program. Multi-period polyhedral risk measures are not coherent in general, but coherency can be obtained through a special choice of parameters.
The class of conditional risk mappings resembles the conditional expectations in (4).
Let $\mathcal{F} \subset \mathcal{F}'$ be σ-fields of subsets of Ω and $\mathcal{Z}$ and $\mathcal{Z}'$ be linear spaces of real-valued functions $f(w), w \in \Omega$ measurable with respect to $\mathcal{F}$ and $\mathcal{F}'$, respectively.
Following Ruszcyński and Shapiro [6] we define:

Definition 3. (Conditional risk mappings). *We say that mapping* $\rho : \mathcal{Z}' \to \mathcal{Z}$ *is a conditional risk mapping if the following properties hold:*

1. *Convexity: if* $\alpha \in [0, 1]$ *and X, Y* $\in \mathcal{Z}'$, *then*
$$\alpha\rho(X) + (1-\alpha)\rho(Y) \geq \rho(\alpha X + (1-\alpha)Y)$$
2. *Monotonicity: if* $Y \geq X$, *then* $\rho(Y) \geq \rho(X)$
3. *Predictable Translation Equivariance: if* $Y \in \mathcal{Z}$ *and* $X \in \mathcal{Z}'$, *then*
$$\rho(X, Y) = \rho(X) + Y$$

The inequalities in 1. and 2. Are understood component-wise, i.e., $Y \geq X$ *means that* $Y(w) \geq X(w)$ *for every* $w \in \Omega$.

Remark 2*. Predictable Translation Equivariance is similar to translation invariance from Definition 1, but in a more general setting. General definition of this assumption is still a subject of ongoing discussion and the two mentioned definitions are similar, but not equivalent. Based on our Predictable Translation Equivariance assumption,*

the addition of a real number in the first stage leads to following equation: for each $r \in \mathbb{R}$: $\rho\,(Z_1 + r, Z_2, \ldots., Z_T) = \rho(Z_1, \ldots., Z_T) + r$ *compare with Definition 1.*

For conditional risk mappings defined above we shall use notation $\rho\,(\cdot\,|\mathcal{F})$. Using Predictable Translation Equivariance, we can construct composite risk measures as follows: Consider conditional risk mappings $\rho_2, \ldots., \rho_T$ and a risk function $\rho : \mathcal{Z}_1 \times \ldots.\times \mathcal{Z}_T \to \mathbb{R}$ given by:

$$\rho(Z_1, \ldots., Z_T) = Z_1 + \rho_2(Z_2 + \cdots \rho_{T-1}\big(Z_{T-1} + \rho_T(Z_T)\big))$$

Using Predictable Translation Equivariance we get

$$\rho_{T-1}\big(Z_{T-1} + \rho_T(Z_T)\big) = \rho_{T-1} \circ\ \rho_T(Z_{T-1} + Z_T)$$

By continuing this process we end up with a *composite risk measure* $\bar{\rho}: = \rho_2 \circ \ldots.\circ\ \rho_T$.

It holds

$$\bar{\rho}(Z_1\ + \cdots + Z_T\,) = \rho(Z_1\ + \cdots + Z_T\,) \qquad (7)$$

Using notation of Definition 4 we continue by introducing a concept of dynamic or time-consistent conditional risk mappings [6].

Definition 4. (Time-consistent risk mappings). *A conditional risk mapping* $(\rho_t(\cdot\,|\mathcal{F}_t))_{t=1,\ldots T}$ *is called time-consistent if for all* $1 \le t_1 \le t_2 \le T$ *and* $X, Y \in \mathcal{L}_p(\Omega, \mathcal{F}, \mathbb{P})$*:*

$$\rho_{t_2}\big(X\big|\mathcal{F}_{t_2}\big) \le \rho_{t_2}(Y|\mathcal{F}_{t_2}) \Rightarrow \rho_{t_1}\big(X\big|\mathcal{F}_{t_1}\big) \le \rho_{t_1}(Y|\mathcal{F}_{t_1}$$

2.5 CVaR risk measures and stochastic multiphase programs

We formulate a multi-stage stochastic linear program with CVaR risk measure in various versions. We're going to state a multi-stage stochastic program with a multi-period CVaR risk measure and with a sum of CVaR risk measure.
All models have random parameters in stages $t = 2, \ldots, T$, e.g. $\xi_t = (c_t\big(\xi_{[t-1]}\big), A_t\big(\xi_{[t-1]}\big), B_t\big(\xi_{[t-1]}\big) b_t\big(\xi_{[t-1]}\big)$ in (3), which are governed by a known conditional distribution. All models can be also formulated in a more general convex form (which is solvable using the SDDP algorithm), but we have chosen the linear versions for easier presentation of our results. For simplicity of notation we will drop

the ($\xi_{[t-1]}$) arguments and denote the random parameters only by $\xi_t = (c_t, A_t, B_t, b_t)$. The parameters of the first stage, $\xi_1 = (c_1, A_1, b_1)$.are assumed to be known. Our models allow specification of different risk aversion coefficients, λ_t $[0, 1]$, and confidence levels, α_t $(0, 1)$, at each stage, $t = 1, \ldots, T$.

2.5.1 Multi-period CVaR model

The multi-period CVaR model is based on the following risk measure:

$$\rho^m(Z) = \sum_{t=2}^{T} \mu_t \mathbb{E}\left[CVaR_{\alpha t}[Z_t|\mathcal{F}_{t-1}]\right] \tag{8}$$

with $\sum_{t=2}^{T} \mu_t = 1,\ with\ \ \mu_t \geq 0\ \forall t$. The multi-period CVaR risk measure is time-consistent with respect to the Definition 6.
Considering the aim of the present study, we can introduce the following multi- period CVaR model:

$$\begin{gathered}
\min_{x_1, \ldots x_T} c_1^T x_1 + \mu_2 \rho_{2\xi_{[1]}}[c_2^T x_2] + \cdots + \mu_T \mathbb{E}[\rho_{T,\xi_{[T-1]}}[c_T^T x_T]] \\
\text{s.t. } A_1 x_1 = b_1 \\
A_2 x_2 = b_2 - B_2 x_1 \\
\cdots \\
A_T x_T = b_T - B_T x_{T-1} \\
x_t \geq 0, x_t \in \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P}), \quad t = 1, \ldots . T
\end{gathered} \tag{9}$$

We assume model (9) is feasible, has relatively complete recourse, and has a finite optimal value. While $\rho_{2\xi_{[1]}}$ is deterministic, $\rho_{T,\xi_{[T-1]}}, t = 3, \ldots, T$ are random variables and expectation is applied to get a meaningful model. In this context we give a reformulation by using the classical definition of conditional value at risk and auxiliary variables q_t to express its nonlinear term:

$$\begin{gathered}
\min_{x_1, u_t, q_t \forall_t} c_1^T x_1 + \sum_{t=1}^{T-1} \mu_{t+1} \mathbb{E}[\lambda_{t+1} u_t] + \sum_{t=2}^{T} \mu_t \mathbb{E}\left[(1 - \lambda_t) c_t^T x_t + \frac{1}{\alpha_t} \lambda_t q_t\right] \\
s.t \qquad A_1 x_1 = b_1 \\
A_t x_t = b_t - B_T x_{t-1}, \qquad\qquad t = 2, \ldots . T
\end{gathered} \tag{10}$$

$$q_t \geq c_t^T x_t - u_{t-1} \qquad t = 2, \ldots . T$$

$$q_t \geq 0, \qquad t = 2, \ldots . T$$

$$u_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), \qquad t = 1, \ldots . T-1$$

$$q_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), \qquad t = 2, \ldots . T$$

$$x_t \geq 0, x_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), \qquad t = 1, \ldots . T$$

As was shown by [7] the multi-period risk measure (8) is polyhedral.

Remark 3. *There is an inequality comparing the values of multi-period and nested* CVaR *risk measures. Under the assumption of* $\alpha = \alpha_t$ *we have that:* $\rho^n(Z) \leq \alpha^{-(T-2)}\rho^m(Z)$. *However, this bound can be loose for programs with large values of* T. *Besides that, it cannot be applied to the optimal values of our optimization problems, since they combine the value of the risk measure with the mean return.*

Similarly, as in the case with nested CVaR model we develop dynamic programming equations. Contrary to the nested model, CVaR is now evaluated in the stochastic program which determines the recourse value. In consequence, notation of the recourse value $Q_t\big(x_{t-1}, u_{t-1}, \xi_{[t]}\big)$ includes auxiliary variables u_t.

Using the interchangeability principle (see [8]) we get:

$$\begin{gathered} \min_{x_1, u_1} c_1^T x_1 + \mu_2 \lambda_2\, \mu_1 + Q_2(x_1, \mu_1, \xi_{[t1]}) \\ s.t.\ \ A_1 x_1 = b_1 \\ x_1 \geq 0 \end{gathered} \tag{11}$$

with the recourse value $Q_t\big(x_{t-1}, u_{t-1}, \xi_{[t]}\big)$ at stage $t = 2, \ldots, T$ given by:

$$\begin{aligned} Q_t\big(x_{t-1}, u_{t-1}, \xi_{[t]}\big) = \\ = \min_{x_t, u_t, q_t} \mu_t(1-\lambda_t) c_t^T x_t + \mu_{t+1}\lambda_{t+1} u_t + \mu_t \frac{1}{\alpha_t}\lambda_t q_t \\ + Q_{t+1}(x_t, u_t, \xi_{[1]}) \end{aligned} \tag{12}$$

$$\text{s.t. } A_t x_t = b_t - B_t x_{t-1}$$

$$q_t \geq c_t^T x_t - u_{t-1}$$

$$q_t \geq 0$$

$$x_t \geq 0$$

where

$$\mathcal{Q}_{t+1}(x_t, u_t, \xi_{[1]}) = \mathbb{E}_{\mathbb{P}_{t+1[\cdot|\xi_{[t]}]}}[Q_{t+1}(x_t, u_t, \xi_{[t+1]})] \tag{13}$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

2.5.2 Risk measures: Sum of the CVaR model

The weighted sum of CVaR model is based on the following risk measure:

$$\rho^s(Z) = \sum_{t=2}^{T} \mu_t CVaR_{\alpha t}[Z_t]$$

with $\sum_{t=2}^{T} \mu_t = 1, \mu_t \geq 0 \ \forall t$.

It can be shown that sum of CVaR is not a time-consistent risk measure with respect to the Definition 6. The sum of CVaR model does not include nesting of the recourse values. It can be deduced from the scalarization technique of the multi-objective optimization. Following this procedure, we obtain:

$$\min_{x_1, \dots x_T} c_1^T x_1 + \mu_2 \rho_{2\xi_{[1]}}[c_2^T x_2] + \cdots + \mu_T \rho_{T,\xi_{[1]}}[c_T^T x_T]$$

$$\begin{aligned} s.t \quad & A_1 x_1 = b_1 \\ & A_2 x_2 = b_2 - B_2 x_1 \\ & \cdots \\ & A_T x_T = b_T - B_T x_{T-1} \\ & x_t \geq 0, \\ & x_t \in \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P}), \ \ t = 1, \dots . T \end{aligned} \tag{14}$$

We assume again that model (14) is feasible, has relatively complete recourse, and has a finite optimal value. Please note that no nesting of the CVaR values is present and that we always condition the operator ρ with the first stage information $\xi_{[1]}$, i.e. $\rho_{t,\xi_{[1]}}$ is deterministic $\forall t = 2,\dots,T$.
Using mean-risk operator[1] and auxiliary variables q_t to express the non-linear term in $CVaR_\alpha[Z] = \min\left(u + \frac{1}{\alpha}\mathbb{E}\,[Z-u]_+\right)$ we can rewrite the model as the following multi-stage stochastic linear program:

$$\min_{x_t,u_t,q_t \dots \forall t} c_1^T x_1 + \sum_{t=1}^{T-1} \mu_{t+1}\,\lambda_{t+1} u_t + \sum_{t=2}^{T} \mu_t \mathbb{E}\left[(1-\lambda_t) c_t^T x_t + \frac{1}{\alpha_t}\lambda_t q_t\right]$$

$$\begin{aligned}
s.t \quad & A_1 x_1 = b_1 \\
& A_t x_t = b_t - B_T x_{t-1}, && t = 2, \dots . T \\
& q_t \geq c_t^T x_t - u_{t-1}, && t = 2, \dots , T \\
& q_t \geq 0, && t = 2, \dots , T \\
& u_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), && t = 1, \dots . T-1 \\
& q_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), && t = 2, \dots . T \\
& q_t \geq 0, && t = 2, \dots , T \\
& u_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), && t = 1, \dots . T-1 \\
& q_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), && t = 2, \dots . T \\
& x_t \geq 0, x_t \in \mathcal{L}_p\,(\Omega, \mathcal{F}_t, \mathbb{P}), && t = 1, \dots . T
\end{aligned} \tag{15}$$

It can be seen that the risk measure $\rho^s(Z)$ used in this linear program satisfies requirements of Definition 2 and is therefore polyhedral and the corresponding optimization model is time-consistent. (see for more [5]). We again develop dynamic programming equations using the interchangeability principle (see [8]):

[1] $\rho_{t,\xi[t-1]}[Z] = (1-\lambda_t)\,\mathbb{E}\big[Z \mid \xi_{[t-1]}\big] + \lambda_t\, CVaR_{\alpha t}\,\big[Z \mid \xi_{[t-1]}\big]$.

$$\min_{x_1,u_1,u_{T-1}} c_1^T x_1 + \sum_{t=1}^{T-1} \mu_{t+1}\,\lambda_{t+1} u_t + Q_2(x_1, \mu_1, \dots\dots, \mu_{T-1}, \xi_{[1]}) \tag{16}$$

$$s.t \quad A_1 x_1 = b_1$$
$$x_t \geq 0$$

with recourse value $Q_t(x_{t-1}, \mu_{t-1}, \dots\dots, \mu_{T-1}, \xi_{[1]})$ at stage $t = 2, \dots\dots, T$ given by:

$$Q_t(\dots) = \min_{x_t, q_t} \mu_t(1-\lambda_t) c_t^T x_t + \mu_t \frac{1}{\alpha t} \lambda_t q_t + Q_{t+1}\,(x_t, \mu_t, \dots\dots, \mu_{T-1}, \xi_{[t]})$$

$$s.t \quad A_t x_t = b_t - B_T x_{t-1} \tag{17}$$

$$q_t \geq c_t^T x_t - u_{t-1}$$

$$q_t \geq 0,$$

$$x_t \geq 0$$

where:

$$Q_{t+1}\left(x_t, \mu_t, \dots\dots, \mu_{T-1}, \xi_{[t]}\right) = \mathbb{E}_{\mathbb{P}_{t+1[\cdot|\xi[t]}}\left[Q_{t+1}\left(x_t, \mu_t, \dots\dots, \mu_{T-1}, \xi_{[t]}\right)\right] \tag{18}$$

We take $Q_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

2.6 Stocastic Dual Dynamic Programming

In applying SDDP[2] algorithm, we have to assume $\xi_{[t]}\, t = 2, \dots, T$, to be stage-wise independent. It may seem that such assumption is overly restrictive, but this is not the case. The stage dependence is central in the frame of Stochastic Programming in particular for what concerns the multi-level one. Moreover, for tractability and storage reasons, fully general models cannot be solved in practice for large number of stages.

The assumption of stage-wise independence further simplifies formulations of the presented models. The dynamic programming equations are given by:

[2] Stochastic Dual Dynamic Programming.

$$\min_{x_1,u_1} c_1^T x_1 + \lambda_2\,\mu_1 + Q_2(x_1,\mu_1)$$

$$s.t.\ A_1 x_1 = b_1 \tag{19}$$

$$x_1 \geq 0$$

with the recourse value $Q_t(x_{t-1},\xi_{[t]})$ at stage $t = 2,\dots T$ given by:

$$Q_t(x_{t-1},\xi_{[t]}) = \min_{x_1,u_1} c_1^T x_1 + \lambda_{t+1}\,\mu_t + Q_{t+1}(x_t,\mu_t)$$

$$s.t \quad A_t x_t = b_t - B_T x_{t-1} \tag{20}$$

$$x_t \geq 0$$

where

$$Q_{t+1}(x_{t,}u_t) = \mathbb{E}\left[(1-\lambda_{t+1})Q_{t+1}(x_{t,}\xi_{t+1}) + \frac{\lambda_{t+1}}{\alpha_{t+1}}[Q_{t+1}(x_{t,}\xi_{t+1}) - u_t]_+\right] \tag{21}$$

Similar development applies to the function $Q_{t+1}\ (x_t,\mu_t,\xi_{[t]})$ from equation (13) and the function $Q_{t+1}\ (x_t,\mu_t,\dots\dots,\mu_{T-1},\xi_{[t]})$ from equation (18) which will be written as $Q_{t+1}\ (x_t,\mu_t,\dots\dots,\mu_{T-1})$ in the stage independent case.

2.6.1 A multi-period asset allocation model

In addition, we include computational results for a second model, which is based on multi-period CVaR risk measure. The decisions, random parameters and notation remain the same as in the case of a nested model. We also suppose the same behavior of transaction costs, relative to the volume of traded assets. Under the assumption of stage independence, first stage of model (11) specializes to:

$$\min_{x_1,u_1}\ \lambda_2\,\mu_1 + Q_2(x_1,\mu_1)$$

$$1^T x_1 = 1 \tag{22}$$

$$x_t \geq 0$$

with the recourse value $Q_t(x_{t-1},\mu_{t-1},\xi_t)$ at stage $t = 2,\dots,T$ given by:

$$\mathcal{Q}_t(x_{t-1}, \mu_{t-1}, \xi_t) = \min_{x_t, o_t, u_t, q_t} (1-\lambda_t) c_t^T x_t + \lambda_{t+1}\mu_t + \frac{1}{\alpha_t}\lambda_t q_t + \mathcal{Q}_{t+1}(x_t, u_t)$$

$$\begin{aligned} s.t. \quad & 1^T x_t = f_t 1^T o_t = r_t^T x_{t-1} \\ & o_t - x_t \geq -x_{t-1} \\ & o_t + x_t \geq x_{t-1} \\ & q_t \geq c_t^T x_t - u_{t-1} \\ & q_t \geq 0, \\ & x_t \geq 0 \end{aligned} \tag{23}$$

where:

$$\mathcal{Q}_{t+1}(x_t, \mu_t) = \mathbb{E}[Q_{t+1}(x_t, \mu_t, \xi_{t+1})] \tag{24}$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

2.7 Multi-objective programming: an essential introduction

Some concerns analyzed in this work are connected with following the multi-objective programming problem:

$$(VP): \quad \min f(x), g(x) \leq 0, \qquad x \in X_0$$

Where VP stands for Vector Programming and X_0 is a nonvoid open subset of $\mathbb{R}^n$ $f = (f_1, f_2, \dots, f_p): X_0 \to \mathbb{R}^p$, $g = (g_1, g_2, \dots, g_m): X_0 \to \mathbb{R}^m, f_i, i \in \{1, 2, \dots, p\}$ and $g_j, j \in \{1, 2, \dots, m\}$ are assumed to be differentiable on X_0. The symbol "min" is used with the generic meaning of finding solutions of one of the types defined below.
Let *X* be the set of all feasible solutions of (VP),

$$X = \{x \in X_0 | g(x) \leq 0\}$$

Definition 5. *$a \in X$ is said to be a weakly efficient solution of (VP) if there is no $a \in X$ such that $f(x) < f(a)$.*

Definition 6. *$a \in X$ is said to be an efficient solution of (VP) if there is no $a \in X$ such that*

$$f(x) \leq f(a), f(x) \neq f(a).$$

Definition 7. *$a \in X$ is said to be a properly efficient solution of VP if it is efficient and there exists a positive constant K such that for each $x \in X$ and for each $i \in \{1, 2, \dots p\}$ satisfying $f_i(x) < f_i(a)$ there exists at least one $j \in \{1, 2, \dots p\}$ such that $f_i(x) < f_i(a)$ and $f_i(a) - f_i(x) \leq K\left(f_j(x) - f_j(a)\right)$.*

Denoting by $WE\ (VP), E(VP)$, and $PE\ (VP)$ the sets of all weakly efficient, efficient, respectively, properly efficient solutions of (VP), we have:

$$WE(VP) \supseteq E(VP) \supseteq PE(VP)$$

2.8 MOEAs (Multi-Objective Evolutionary Algorithms)

A general framework of a multi-objective optimization problem - useful for the study of the problem presented in the next chapter 6 - with conflicting objectives is described as follows:

$$Minimize\ F(x) = \left(f_1(x), \dots, f_m(x)\right) \tag{25}$$

$$s.t.\ \ x = (x_1, x_2, \dots., x_N) \in X \tag{26}$$

where, $x = (x_1, x_2, \dots., x_N)$ presents a vector of decision variables, and X is the number of possible solutions and $F(x)$ is an objective function vector that contains values of m objectives maps the possible set X into the set F which represents all possible values of the objective functions. All objective functions may be maximized, minimized, and be in a mixed manner. The general procedure in multi-objective is to search the entire Pareto optimal and non-dominated solutions, i.e., any solution we cannot improve on one objective function without worsening on another.
A decision variable x is mentioned as the dominating variable over decision variable y if the following conditions are satisfied,

a) $f_i(x) \leq f_i(y)$ it holds $\quad \forall i \in \{1, \dots, m\}$

b) $f_i(x) < f_i(y)$ is valid for a single value of $i\ \forall i \in \{1, \dots, m\}$

It is written as $F(y) \leq F(y)$. Now, a set of Pareto optimal solutions is described as

$$P = \{x \in X | \nexists\, y \in X, for\ which\ F(x) \leq F(y)\} \quad (27)$$

The multi-objective optimization problem can be converted into a single objective optimization problem using the parametric approaches, of which the weighted sum approach is one of them. A single run of the algorithm does not generate the set of optimum trade-offs amidst objectives. Moreover, in the case of the non-convex problem, the best approximate solutions of objectives are considered as Pareto optimal set. The searching techniques are employed in a MOEAs are generally using the concept of the population which is framed for obtaining a set of best approximate optimum solutions of trade-offs. Since no assumptions are made regarding the framework of the problem's objectives and constraints, it allows them to solve complex optimization problems to estimate a set of solutions near the Pareto front. We handled the CCPO with a Composite index along with its upper bound (tri-objective portfolio optimization problem) using MOEAs in this study.

CHAPTER 3

Deep Learning, Machine Learning and Sentiment Analysis

3.1 Introduction

Nowdays the analysis of textual data represents, for the community, one of the most important horizons, both in terms of volume and relevance of the information obtainable.
With the increasing availability and popularity of opinion-rich resources such as social media and review sites, new opportunities and challenges arise as it is possible to proactively use information technologies to seek and understand the opinions of others. The reason why opinion mining (or Sentiment Analysis) has recently received great interest from the academic community and commercial companies derives from its ability to provide a series of tools suitable for analyzing public opinion on various topics. The discipline, the subject of discussion in part of this work, focuses on the development of methods capable of automatically detecting the information contained in a text and determining the polarity of opinion towards a specific entity. Since we are immersed in an immense amount of textual data to handle, artificial intelligence can be used with good results to do much of the decryption work. In particular, Artificial Intelligence is used to break down the textual content into component parts (nouns, verbs, sentimental words, etc.) in order to develop an understanding of the author's sentiment. For many text analysis activities, but in particular for large-scale ones where the text requires relatively standardized analysis, Machine Learning can prove to be an excellent approach because it is able to build analytical models independently. In fact, the Machine Learning area dedicated to the meaning of the written word is known as Natural Language Processing (from now on NLP). Therefore, Sentiment Analysis is one of the most active research areas of NLP and is widely studied and applied in the fields of data mining, web mining and text mining. This is because the NPL algorithms allow the computer to "read" and analyze textual information, recognizing similar concepts, even if expressed in different ways. The development of algorithms capable of improving the behavior of the machine (understood as the ability to act and make decisions) by learning through experience has certainly represented one of the main steps forward in the history of Artificial Intelligence (AI). Building algorithms capable of learning from their mistakes is in fact fundamental to create intelligent systems that operate in contexts for which programmers cannot predict all the development possibilities a priori. What characterizes AI, from a technological and methodological point of view, is precisely the method of learning with which intelligence becomes skilled in a task or action.

In particular, the two learning models discussed in this study are:

- **Machine learning:** branch of AI that concerns the study, construction and implementation of mathematical algorithms through which a machine is allowed to learn independently and, starting from a set of input data, to build predictive models by reducing the weight of errors at the end of each learning process. In this way the machine can carry out a specific activity without being previously programmed [1]. In particular, we distinguish three approaches: supervised machine learning, where the labels are created by the trainer to make the machine capable of discovering relationships between inputs and labels; unsupervised machine learning where, on the other hand, labels are not available and the machine is simply asked to find clusters within the data and, finally, reinforcement learning that creates algorithms capable of learning and adapting to changes in the environment.
- **Deep learning**: learning model inspired by the structure and functioning of the biological brain. If Machine Learning trains AI, deep learning represents the algorithm that allows you to emulate the mind of man. In this case, however, deep learning requires both algorithms and artificial neural networks designed ad hoc (Deep Artificial Neural Networks) and a computational capacity capable of supporting different layers of calculation and analysis [2]. Focusing on Artificial Neural Networks (ANN), systems in use today in pattern recognition, speech and image recognition and in Natural Language Processing systems, they are mathematical-computer computing models that try to simulate biological neural networks, that is systems made up of thousands of interconnections between neurons (synapses) that allow us to reason and manage every function of the body [3]. Furthermore, they are defined as non-linear structures of statistical data organized as modeling tools: they receive external signals on a layer of input nodes (which represents the processing unit, the processor); each of these input nodes is connected to various internal nodes of the network which, typically, are organized at several levels so that each single node can process the signals received by transmitting the result of its processing to the subsequent levels (therefore more advanced information, detailed).

In particular, the nodes are located on levels which can be of three types [4]:

1. **Input Layer**: level designed to receive information from the outside in order to learn to recognize and process such information.
2. **Hidden Layer**: connects the input layer with the output layer and helps the neural network to learn the complex relationships analyzed by the data. There are often more than one hidden levels.
3. **Output Layer**: final level that shows the result of what the program has managed to learn.

A weight is associated with each connection between neurons which determines the importance of the input value. The starting weights are set randomly. Each neuron, on the other hand, has an activation function that allows to define an output given a series of input data analyzed by the neuron itself and once the input data has crossed all the layers of the neural network, it returns the data of output through the output level. More specifically, neural networks are programmed to learn in the same way and use an algorithm called Backpropagation. This algorithm involves comparing the result obtained from a network with the output that you actually want to obtain and, using the difference between the two results, it plans to change the weights of the connections between the levels of the network starting from the output level [5]. Going backwards, the algorithm modifies the weights of the hidden levels and finally those of the input levels. Going backwards, the algorithm modifies the weights of the hidden levels and finally those of the input levels. To do this, develop a cost function appropriate to the problem to be solved. Ultimately, from a mathematical point of view, a neural network can be defined as a composite function, that is, dependent on other functions which in turn can be defined differently according to further functions on which they depend. As previously mentioned, neural networks have gained increasing popularity in the NLP community in recent years and in addition to feed forward neural networks and convolutional neural networks (CNNs), recursive neural networks (RNNs) and their variants are most common neural elements used in NLP, due to their natural ability to manage sequences. In recent years, two major innovations in deep learning have been introduced into NLP; sequence-to-sequence learning [6] and the mechanism of attention [7]. Reinforcement learning and generative models are also gaining in popularity [8]. A sophisticated type of generative neural network that uses an unsupervised machine learning model to produce results is the Deep Belief Network.
These types of networks are made up of various smaller unsupervised neural networks. One of the common characteristics of a Deep Belief Network is that although the layers have connections to each other, the network does not include connections between units in a single layer.
In this Chapter, therefore, we will analyze the tools and approaches of Machine Learning and Deep Learning applied to the pandemic phenomenon from Covid-19. In particular, through the use of neural networks and therefore Deep Learning techniques, the sentiment of individuals on social media regarding Covid-19 is studied. The models, techniques and tools used will be analyzed in more detail in Chapter 7. Furthermore, we will see how the above approaches can be used to identify the various stages of Covid-19 disease. Indeed, X-ray imaging has proved to be an effective screening procedure for predicting the various stages of the pandemic as will be described in Chapter 8.
A further line of research - which will be analyzed in Chapter 9 - concerns a forecast model for the price of Cryptocurrencies, or Bitcoins. The goal is to build a model to predict Bitcoin's closing price along with Bitcoin's opening price, Bitcoin's highest price of the day, Bitcoin's lowest price of the day, Bitcoin's volume of the day and the market capitalization of Bitcoin on a given day using Deep Learning algorithms and

various machine learning concepts, which can find hidden patterns in the data, combine them and make considerably more accurate predictions.

3.2 Machine Learning and Deep Learning approaches to analyse the Fake News on Social media

Propaganda, half-truth, fraud, and erroneous, misleading, sensational headlines, "fake news" content is not a new occurrence. What makes the difference is that technology has advanced so much that it now aids social media in the rapid spread of disinformation. It is so fast that a couple can cough and sneeze in their bedroom, put it on the internet, and within a short period, such action can go viral with millions of people who have no idea what the couple's original intention might be. The Covid-19 pandemic has again shown the negative part of digital technology and the internet of things (IoT) that promotes all kinds of conspiracy theories and people who took actions such as applying fake coronavirus cures, making false claims. As a result of these detrimental health instructions, some people were exposed to more dreaded sicknesses than the virus itself. We have witnessed confusion, fear, and distrust from politicians, stakeholders, and so on. Enemies have been created, and others have revolted against compliance with state orders and policy to stay at home. Reports gathered from different multimedia services, including Facebook, show approximately 50 million contents in regards to Covid-19 disinformation and misinformation were removed in April 2020. Twitter, on the other hand, questioned about 1.5 million of their users for dissemination of fake news and displaying what they called false "manipulative behaviors" in the same period of time. Google is not left out with YouTube and Gmail, where about 18 million scam emails were flagged and blocked while many uploaded videos on YouTube with misinformation about coronavirus were brought down [9, 10]. The authors in [11] concur with the idea that multimedia services have become a citadel of rumor mongers spreading fake news, a joint of unfiltered news with the motive of defrauding, confusing, and misleading other users. Information was gathered from over 500 websites with fake news, and close to 1000 fake stories were discovered on Twitter and Facebook, so a model was proposed that performs identification and secures the subject of fake news in social networks. Figure 1 shows what is currently obtainable in regard to fake news on the internet and the mind-boggling scenarios created for internet users.

No one will dispute the fact that idleness and the stay-at-home policy promoted the increasing usage of social media and thus endorsed the explosion of fake news argued by the author in [12], proposed a framework to solve it. This framework combines the domains of the frequency and pixels and engages them as visual information to detect fake news in the network. The authors have used a CNN-based network in their design. The captured images features are then extracted

after making use of the multi-branch convolutional neural networks-recurrent neural networks (CNN-RNN) model. Further analysis on a real-world dataset demonstrated that their model performs better than other existing state-of-the-art models with a 9.2% in accuracy and performance detection increased by over 5.2%.

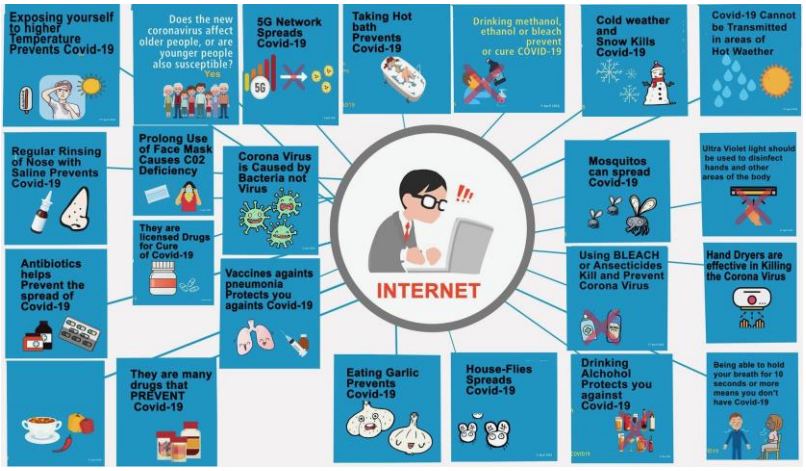


Figure 1: Fake news realities

The contributions of this study are such:

- The research uses the process of Information fusion to obtain real news data from various multimedia services: the New York Times, Health Harvard, Centers for Disease Control and Prevention (CDC), World Health Organization (WHO), and Global Health Now. While Fake News data is collected from Facebook, YouTube, and other social media sites.

- The dataset we used consists of only text and class labels, and its performance with deep learning classifiers like Gated recurrent units (GRU), Long short-term memory (LSTM), and recurrent neural network (RNN) is very low. In order to improve the accuracy, precision, recall, and F1-Measure, we have proposed 39 novel features for text, which has not been used before for fake news to the best of our knowledge.

- We have proposed sentiment features, linguistic features, and name entity base features. After extracting features from the text, our new features detect fake news in Covid-19 with an accuracy of 86.12%. Thus, accuracy is increased by 20% with novel features.

A quantitative analysis of themes that emerged from the twitter account of Prime Minister Trudeau's of Canada and President Trump of USA during the climax of

the Covid-19 pandemic shows how different their sentiments and perspectives of the virus are. Politics is the emerging theme of one, while the public health and policy has been the focus of the other, according to authors in [13]. Network science was made use of, which considers interactions between systems in formation called "co-occurrence networks". It links keywords that are present in the two tweets. For example, keywords like "covid19" and "pandemic" were linked if in the same tweet. The top 100 keywords from @realDon-aldTrump and @JustinTrudeau were gotten based on how many times they were used. The analysis conducted on the network proposes that politicians' social media messages play a role in moulding the views of the public concerning the pandemic and the attempts to regulate it based on the expressed sentiments.

With the emergence of various deep neural network models, sentiment analysis tasks have once again made significant progress. However, these neural network models could not accurately capture sentiment information on sentiment analysis tasks, which leads to their instability [14]. Aspects category or aspect terms have been combined to form a final sentiment [15]. Emotions and sentiment of software developers can largely influence the software productivity and quality. However, existing work on emotion mining and sentiment analysis is still in the early stage in software engineering in terms of accuracy, the size of datasets used and the specificity of the analysis [16].

False information littered about the Covid-19 pandemic is a very important aspect that could greatly influence the response to it [9]. The infographic encourages people to verify information about the pandemic by checking various multimedia sources such as the official site of the WHO, reading the full article instead of just the headline, as well as checking out the credentials and qualifications of the author. It has been shown recently that visual information such as images and videos almost always accompanies fake news and is quite useful in deciphering which is fake news.

There have been several studies on fake news identification. The Multi-domain Visual Neural Network (MVNN) proposed in [12], ascertains fake news by aligning the semantic information in the pixel domains and their frequency, i.e. it studies the visual content accompanying a news post to determine if the news is fake or not. It is a well-known fact that news channels, usually sponsored by the government, are fond of distributing fake news around social media in order to divert the populace's attention from crucial matters such as healthcare, education and the state of the country's economy. India was used as a case study to prove that main stream news channels are deliberately spreading fake news to incite unhealthy nationalism, hatred and division between citizens of various communities with variant religious and political beliefs, in order to distract them from the real issues at hand [17]. In concurrence with this, Qi *et al.* [18] describe how some writers produce fake news articles to influence the results of an election, as well as cause harm by triggering the negative emotions of people. Most of such articles are quite a resemblance of the real articles but include information that is

usually exaggerated. Also, these articles possess misleading headlines, known as clickbait in order to lure people to view them. Often, people do not care to read the articles; they absorb the misleading headlines and distribute them.

In recent times, society has become very dependent on social media for the provision of information. And while it has proven useful with its quick and cheap access to information, like any other thing, it has its disadvantages. This includes the rapid dissemination of fake news without people verifying the authenticity of the news. More can be done in the detection of these fake news. In [19], an idea was proposed to detect fake news by maintaining properties that explains the structure made for various type of news. Machine learning approaches as well as other detection methods such as linguistics basics, clustering, and predictive modelling and so on, have proven to be effective in detecting fake news from real news [20]. Another machine learning system, Bayesian, was developed using specific toolkits such as SciPy to examine the probability that a news post being fake. This fresh technique is called influence mining and was propounded in [21]. It is important to note that time is of the essence in differentiating between real and fake news. Detection of fake news while still in the early stages of its dissemination will go a long way in avoiding massive misin formation and misleading [22]. Some recent advancements in fake news detectors, no work has intricately proposed early detection of the fake news.

Bhoir [23] also used various machine learning models such as Gated Recurrent Unit Network (GRU), Naïve Bayes, Decision Tree to determine whether the news is fake based on previously witnessed fake or real news. The solution to the problem of fake news is simply a very effective and efficient way of identifying the authenticity of news. The authors of [22] explored various classification methods and suggested a Natural Language Processing method which determines whether the news is fake based purely on its content. In [24], an automatic inference model that could detect the credibility of fake news was introduced. This 'FAKEDETECTOR' creates a unit model called the GDU which accepts and studies various sources such as articles and creators simultaneously and fuses them to create an output. In 2019, a survey was conducted to obtain the public opinion on whether false information being disseminated was being manipulated by government officials of the Thailand government [25]. 291 respondents participated in the survey, and most of them reported that they saw fake news several times a month 29% reported that they chanced upon fake news every day. A majority of the interviewees claimed to not put trust in the media and blamed the government for tampering with the dissemination of authentic information. An ensemble technique which uses GRU and LSTM (some neural networks) as well as android software to determine the veracity of the news article has been proposed [26]. The app developed from this study connects to the Java Web Server and verifies from various sources on the Internet whether the news is "fake" before displaying the result. The experimental results of the proposed model look propitious and may further be experimented on.

While discussing the spread of misinformation, it is important to decipher which is satire and which is fake news. While fake news is often meant to mislead people, satire holds up human vices to ridicule them. Its purpose is usually more amusing than misleading. The authors in [27] proposed a model to detect satiric works using subjective concepts and a supervised learning approach which showed 82.5% accuracy when combined with text data. A different line of research was carried out in [28] to determine how long fake news stay on the internet. They posit that fake news typically vanishes from the internet after a certain period of time (probably after it has achieved its goal), while real news stay on the internet forever. In [29], a framework referred to as the MCE for detecting falsified news was proposed. It does this by learning the latent space real news vectors' vastness be- comes greater than the fake news' own. The authors of [30] tried to solve the problem of fake news distribution using a computational model gotten from machine learning models probabilistic and geometric. They stated that content, headline, statistics of the engagement and the news source may assist in fake news classification.

The issue of fake news is more of a menace than we give it credit for and should be seriously looked into. As at 2011, it was recorded that 32.8% of the earth's population made use of the internet. It's a no-brainer that the number has surely increased by now, and the risk of distributing fake news is greater now more than ever. Bhutani *et al.* [31] dwelled on this growing risk, and noted that while various researches have been conducted to find solutions to the problem, the proposed models are still in the early stages of development and are not ready to be implemented. They also discussed how concepts like freedom of expression, the neutrality of the internet and archaic digital laws make it challenging to solve the issue of fake news. They, however, proposed solutions such as creating websites that check information accuracy and merge them with teams that monitor media. The authors of [32] analyzed the different text pre-processing techniques and used sentiment to accelerate the accuracy of detecting fake news. They also stated that focus will be made on neural network algorithms to further enhance the accuracy, as well as expanding their dataset to visual content. Sear *et al.* [33] propounded that the agents of fake news are malevolent software (bots) that socialize with natural users. They described credulous users as those who have a higher number of these 'bots' as friends than their social friends. The results of an experiment carried out proved that credulous users post a larger amount of fake news than their counter parts. They offered that the study of these credulous users could aid in the understanding of misinformation, and could be turned into a weapon to battle fake news.

The Covid-19 pandemic has made room for another rift between people who have now formed groups such as the 'anti-vax community'; people that are against vaccination and the 'pro-vax community'; the people who are for it. A recent study shows that the anti-vax community online have more distinct and broader conversations about Covid-19, thereby encouraging new users to join their

community [34]. The lack of a vaccine and the hindrance of creation of one is disturbing because then, there is nothing to stop this pandemic. Their papers tries to measure the content of Covid-19 information among the various communities' posts of health guidance concerning the vaccinations. It also tries to provide strategies to solve the issues of users going through a large amount of misleading information about their health. It has been discovered that a quick diagnosis of Covid-19 aids in the successful medical treatment of it. In [35], a learning framework for classifying information on Covid-19 and lesion localization was developed. This framework can be applied clinically to get a quick and accurate Covid-19 diagnosis. As a result of the rapid misinformation spreading at such a crucial time, Rustam *et al.* [36] provided an encyclopedic review of the key aspects of the pandemic. A lot of countries' economy has been in shambles as a result of Covid-19, so the paper also explored the use of technologies such as the Blockchain, 5G, Artificial Intelligence (AI) in alleviating the impact of the outbreak. The authors of [37] used their study to attempt to predict the amount of patients who may contract the Covid-19 in the future, using the machine learning models. Various forecasting models were used in the research taking into account the number of deaths, newly infected cases and recoveries. This forecast can assist governments of various countries to make informed and quick decisions on how to regulate the crisis.

Cui, Wang & Lee [38, 39] employed a machine learning model for fake news detection at a premature stage with-out depending on external information which may prevent it from causing public uproar. They make use of two kinds of models, the bag-of-words model and the neural networks.

Evidence was presented that fake news online have caused harm, with a reference made to the 2016 US Presidential Election. The authors of [40, 41] try to face the problem of identifying fake news using models of Dirichlet Compound Multinomial (EDCM) distributions. They developed a Bayesian approach to learn the mixture models. Another study [42, 43] attempts to compare different approaches in alleviating the issue of fake news distribution. Some of the approaches used include the hybrid CNN, RNN, as well as Naïve Bayes.

The paper chooses a more effective method from either the machine or deep learning method for providing a solution to the issue of the balance of accuracy and falsity. The authors in [44, 45] strove to identify, review and compare various methods and tools current research has dug up as a panacea to the challenge of fake news, specifically taking into consideration algorithms of machine learning and natural language. It is not unheard of for people's social media accounts to be hacked, especially those with a large number of followers.

These cyber-attacks allow fake news to be distributed to a large amount of people using the hacked account. Recently, the Twitter accounts of some millionaires and celebrities were hacked, with the scammers offering to double the Bitcoin sent to their Blockchain wallet from the followers. A lot of people fell prey to this scam and thousands of dollars were lost on that one fateful night. Iwendi *et al.* [46, 47]

proposed a Zero code based Watermark detection approach called the KeySplitWatermark that provides software against cyber-attacks. When the software is tampered with to a certain extent, the original software code is restored, making it strong enough to withhold further attacks. The phenomenon of fake news is not unfamiliar; relevant authorities should consider this study and take calculated and measured steps to mitigate it.
Therefore, we have proposed a model using the fake news propaganda around Covid-19 to detect, mitigate and provide a quick response.

3.3 Prediction of Covid-19 X-ray images: new findings by AI

Covid-19 is known to be a transmittable infection, and is found to spread at a rapid rate and affect billions of people around the world. This has ensured that the numbers of affected as well as recovering peoples are increasing day by day. The various symptoms associated with the disease have also undergone changes from its initial days of infecting humans to the present times, and have been seen in many cases as becoming more aggressive. The lung is usually seen to be the most affected organ in Covid-19, as is the case with common bacterial or viral infections. Therefore, it is important to predict the stage of lung disease in a patient once infected by Covid-19 and in this study different technology, especially machine learning techniques, have been introduced in order to predict this, which has demonstrated a fair degree of success and accuracy. The four stages of Covid-19 that were considered in these predictions to determine the degree of progression of lung disease [48] are:

i. Phase-1, where mild cell incursion and viral duplication are observed in the nose and lungs.

ii. Phase-2, where moderate amount replication is seen to affect the lungs and the immune system.

iii. The more severe Phase 3, where the replication is consolidated across all sections of the lungs threatening a collapse of the lungs as they struggle to keep the alveoli open and close.

iv. Phase 4, where the patient is critically affected with an onset of failure of multiple organs. All these various stages of Covid-19 along with the training images used and the related symptoms are shown in Table 1. It is therefore, evident that lung-based initial prediction of the disease and its accurate staging can go a long way in arresting the progression of Covid-19 and facilitating effective treatment.

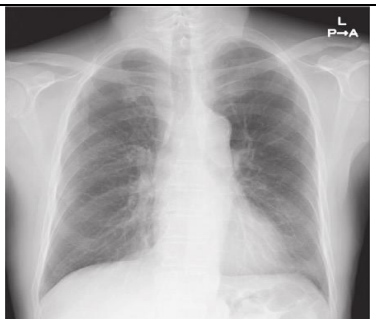
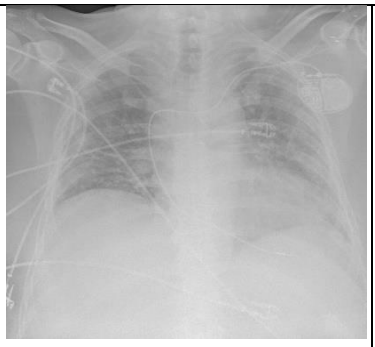
S. No	Stages	Symptoms	COVID-19 Images
1	Stage-1	Invading and replicating in the lungs and nose. [48-50]	
2	Stage-2	Replicating moderately and affecting immune system as well as lungs. [48-50]	
3	Stage-3	Consolidated replication leading to collapse of parts of the lungs. [48-50]	
4	Stage-4	Severe damage to lungs and criticality accompanied by onset of multiple-organ failure. [48-50]	

Table 1: Various Stages of Covid-19

In recent times, several researchers have been seen to use AI-based methodologies, especially machine learning and deep learning, for classifying and predicting images in healthcare where, X-ray, histopathology as well as CT scan images are usually the

subject of predictions. Deep Learning has been hailed as a potent tool that can learn cognitive and complex problems. As a result, there has been a surge in the usage, evaluation, and prediction of diseases using deep learning methods. This study undertook the prediction of Covid-19 stages by utilizing X-ray lung images aided by machine learning as well as deep learning methods. Previous predictions based on deep learning encountered various issues and challenges that this work has addressed while undertaking to make predictions regarding the progression of Covid-19 by using X-ray lung images are shown below:

i. The previous predictions [51, 52, 53, 54] used fewer images for training, testing and, validation, therefore negatively impacting prediction accuracy rates.

ii. There was, consequently, a to improve the efficiency and accuracy of prediction, while using larger datasets.

iii. The previous works [51, 52, 53, 54] used fewer features for prediction, whereas significant features are required for improved accuracy.

iv. The previous predictions did not consider the various stages depicted by the Covid-19 images [48,49] even though incorporating the stages or phases in the prediction would help to improve the treatment significantly.

This work addresses all the above-mentioned issues with its proposed methodology that enabled the contributions.

Computer-aided diagnosing systems were introduced in 1980, and have been applied to different scenarios of disease prediction and diagnosis. Initially, such image-based diagnosing systems were time-consuming and inefficient. However, with the invention of artificial intelligence and graphic processor units in decision support systems, the prediction rates witnessed a rapid improvement [51]. Deep learning models were applied to diseases such as breast cancer, lung cancer, and various other diseases for the purposes of detection and diagnosis. This work is specifically focused on prediction of Covid-19 stages by utilizing X-ray images with a machine learning and deep learning methodology. Various researchers have undertaken similar work with respect to predictions based on the lung X-rays.

The authors of [51] proposed a hybrid model using VGG, spatial transform, and data argumentation. But the accuracy (73%) of the prediction was relatively lesser since fewer images were utilized for the purposes of training and testing. The authors of [55] collected various images of Covid-19 and created datasets. But it had only 123 frontal views of images. The authors of [52] proposed a model for classifying Covid-19 lung images using multiple levels of thresholds and SVM. The authors of [56] made a survey of Covid-19 datasets and presented various datasets such as images, text, and speech. The authors of [57] proposed a model for classifying X-ray images to relevant Covid-19 stages using different augmentations of images. Using this method, the Covid-19 as well as Non-Covid-19 images were classified. The authors of [58]

proposed a model for predicting Covid-19 using X-ray images and deep learning. The ensemble and feature classification techniques such as SVM and bagging classifier were used for prediction. The authors of [59] proposed a framework based on deep learning for predicting Covid-19 by utilizing X-ray images. The authors of [53] proposed a deep Covid model for predicting Covid-19 using X-ray images. In this model, testing and training of 184 images were used for prediction. The authors of [54] similarly proposed the detection of Covid-19 by utilizing a deep learning model. The images were predicted using Xceplm, V3 and ResNet. The accuracy rate of the prediction was 97.9%. The authors of [60] presented images of Covid-19 and introduced chest CT view of scans. The authors of [61] proposed a machine learning based model utilizing multiple-parameters to predict Covid-19. The authors of [62] proposed a model for prediction using chest X-rays. The texture and morphological features used to predict utilized the deep learning method. Table 2 presents a summary of the various existing techniques for prediction as well as the number of images used for testing, training and prediction rates.

S. No	Techniques	Prediction Rate	Images for Training and Testing
1.	VDSNet [51]	Accuracy (73%)	Limited Number of Images used.
2.	Multi-Level Thresholding and Support Vector Machine [52]	Accuracy (95.76%)	25 Normal Images and 25 COVID affected images used.
3	COVID-Net [63]	Accuracy (92.4%)	13,975 and 13,870 patient images used.
4.	Deep COVID [53]	Sensitivity (98%) Specificity (92%)	Training- 84 images, Testing- 100 images.
5.	Xception and ResNext [54]	Accuracy (97.5)	COVID affected image: 86.
6.	Deep Learning and Various Classification Techniques [62]	Multi-classification Accuracy (97.5%) Prediction Accuracy (79.52)	Texture and Morphological features were used to predict. The number of images used was limited.
7.	CNN [64]	Accuracy (98.7)	Two datasets were used to predict but the number of images were limited.

Table 2: Various Methods: Prediction Rates and Number of Images

Part 2

My own results and new findings

CHAPTER 4

Fuzzy Fractional-Order Model of the Novel Coronavirus

4.1 Introduction

Recently, the whole globe has been suffering from a novel coronavirus pandemic, which was named “2019 novel coronavirus”, abbreviated by “2019-nCoV”, and claimed to out-break for the first time in Wuhan City, central China [1]. It has been observed that 2019-nCoV is transmitted from animal to human; as many infected claimed that they had been infected due to a local fish and wild animal market in Wuhan as early as 28 November [2]. Soon after, some researchers confirmed that the transmission also happens from a person to a person [3]. According to the data reported by WHO (World Health Organization), on March 21, 2020, the reported laboratory confirmed human infections in 187 countries, territories, or areas around the world have reached more than 292,142, including 12,784 death cases [4]. Even in some countries, like Italy and Spain, the death rate was as high as almost 0.066. This verifies the severity and high infectivity of 2019-nCoV. It is confirmed that most people infected with 2019-nCoV will experience mild to moderate respiratory illness, such as breath difficulty, low fever, sick, cough, and other symptoms. However, other symptoms such as gastroenteritis and neurological diseases of varying severity have also been reported [5]. The 2019-nCoV is transmited mainly through droplets from the nose when an infected person coughs or sneezes. Once a person breaths the droplets from infected people in the air, he/she will be exposed to the danger of getting the infection. As a result, the best way to prevent the virus is to avoid meetings and touching other people. For this purpose, the Chinese government decided to lock down Wuhan city and cut or limit the transportation system of the country, including airplanes, trains, buses, and private cars, etc., to control population flow and movement. People were required to stay at home and get body temperature taken each day. Respirators were advocated to be worn if people had to go out. With the transmission and outbreak of 2019-nCoV around the world, more governments joined the antivirus battle by following the Chinese government. It was heard that more and more countries started to release regulations to ban international travel, close schools, shopping malls, and companies. The 2019-nCoV pandemic has lead to a serious economic damage in the whole world, and it has also been a great ordeal for the administrations of countries and even to all human beings. A great number of doctors and researchers also devoted themselves to the antipandemic war and did researches based on their expertise.

They looked into 2019-nCoV from various points of view, such as virology, infectious diseases, microbiology, public environmental occupational health, veterinary sciences, sociology, media studies, political economics, etc. China, USA, and Korea are the leading countries on the 2019-nCoV research because the early outbreak of virus urged them to start relevant research immediately. A group of researchers studied the origin of 2019-nCoV. Initially, it was said that bats are the origin of 2019-nCoV, which is similar to SARS (Severe Acute Respiratory Syndrome), an epidemic which broke out in China and other regions of the world in 2003 [6, 7]. Then some researchers compared 2019-nCoV with SARS and MERS (Middle East Respiratory Syndrome) from 2012 to prove the possibilities to learn lessons from the two pandemics happened before in the human history. According to Lu, SARS-CoV, MERS-CoV, and 2019-nCoV all belongs to the same family of Beta coronavirus genus [8]. According to Zhou, previous research indicates that 2019-nCoV has high similarity to SARS-CoV, which is supported by the full-length genome phylogeny analysis, and therefore, has the putative similar cell entry mechanism and human cell receptor usage [9]. Xiaolong and Mose also considered the high identity of RBD (Receptor Binding Domain) in 2019-nCoV and SARS-CoV, and raised the idea that the SARS-CoV specific human antibody, CR3022, could bind potently with 2019-nCoV RBD, KD of 6.3 nM, which indicates that the difference within the RBD of SARS-CoV and 2019-nCoV incorporates a crucial influence on the cross-reactivity of neutralizing antibodies, which is still necessary to develop novel monoclonal antibodies that would bind specifically to 2019-nCoV RBD [10]. Based on the previous studies on SARS- Cov immunological system and structures, Syed *et al.* determined SARS-CoV-derived B lymphocyte epitopes and T cell epitopes experimentally, and located that they are similar and comprise no mutation within the available 2019-nCoV sequences, which is critical to narrow down the hunt for potent targets for an efficient vaccine against the 2019-nCoV. Some researchers put their focus on the transmission of 2019-nCoV virus among humans and its identification. It's well accepted that human-to-human transmission is leading to the rapid growth of infections. Ahmed claimed that viral strains from the infected people of the area have been sequenced; but only little genetic variation was found, implying that they have descended from a common ancestor [11]. On the other hand, Zhou argued that sequences of the seven conserved viral replicase domains in ORF 1ab show 94.6% similarity in 2019-nCoV and SARS-CoV [9]. Chaudhury *et al.* proved that computational protein–protein docking with accurate, physics-based energy functions is able to reveal the native-like, low-energy protein–protein complex from the unbound structures of two individual, interacting protein components [12]. In our work we try to investigate 2019-nCoV infection system mathematically. The fuzzy Laplace transform based on Adomian decomposition is employed to obtain the numerical results which can be helpful for the understanding of the dynamical structures of the physical behavior of 2019nCoV. We define the system of six equations illustrating the outbreak of the coronavirus in the form of nonlinear fractional order

differential equations (FODEs), involving the susceptible people $S_k(t)$, the exposed population $E_k(t)$, total infected strength $I_k(t)$, asymptotically infected population $A_k(t)$, the total number of humans recovered $R_k(t)$, reservoir $M_k(t)$, and corresponding interaction, which are presented as follows [13]:

$$\begin{gathered} D_t^k S_k(t) = n_k - m_k S_k + b_k S_k(I_k + kA_k) - b_k S_k M_k \\ D_t^\gamma E_k(t) = b_k S_k(I_k + \delta A_k) + b_l S_k M - (1-\delta_k) w_k E_k - \delta_k w_k' E_k - m_k E_k \\ D_t^\gamma I_k(t) = (1-\delta_k) w_k E_k - (\gamma_k + m_k) I_k \\ D_t^\gamma A_k(t) = \delta_k w_k' E_k - (\gamma_k' + m_k) A_k \\ D_t^\gamma R_k(t) = \gamma_k I_k + \gamma_k' A_k - m_k R_k \\ D_t^\gamma M_k(t) = \xi I_k + \eta A_k - \upsilon M_k \end{gathered} \tag{1}$$

where n_k represents the rate of birth, m_k represents the death of infected population, b_k represents the transmission coefficient, $b_l$represents disease transmission coefficient, k is transmissibility multiple, w_k and w_k' denote signified incubation period, γ_k and γ_k' represent the recovery rate of I_k and A_k, respectively, ξ and η denote the influence of the virus from I_ke and A_k to M_k, and υ represents the rate of eliminating the virus from M_k. The parameters are explained in Table 1.

In the last few years, modern calculus and DEs have been extended to fuzzy calculus and FODEs [14–18], respectively. Then FODEs were extended to fuzzy FODEs [19–21]. FODEs and fuzzy integral equations have been studied by many researchers to establish the existence and uniqueness theory of solutions [22–27]. When dealing with fuzzy FODEs, it is really tedious to compute more precise solutions to every fuzzy FODE. A lot of efforts have been made by mathematicians in solving fuzzy FODEs by using various methods like perturbation method, integral transform methods, as well as spectral techniques [29–33]. Some researchers performed stability analysis of fuzzy DEs [30]. Here, we are going to investigate model (1) with a fuzzy fractional-order derivative where the uncertainty lies in the initial data. For $0 < \gamma \leq 1$,

$$\begin{gathered} D_t^\gamma \mathrm{Y}_k(t) = \tilde{n}_k - \tilde{m}_k \mathrm{Y}_k - \tilde{b}_k \mathrm{Y}_k\left(I_k + \tilde{k}\mathcal{A}_k\right) - \tilde{b}_k \mathrm{Y}_k \mathcal{M}_k \\ D_t^\gamma \mathrm{V}_k(t) = \tilde{b}_k \mathrm{Y}_k\left(I_k + \tilde{\delta}\mathcal{A}_k\right) + \tilde{b}_l \mathrm{Y}_k \mathcal{M} - \left(1-\tilde{\delta}_k\right)\tilde{w}_k \mathrm{V}_k - \tilde{\delta}_k \tilde{w}_k' \mathrm{V}_k - \tilde{m}_k \mathrm{V}_k \\ D_t^\gamma I_k(t) = \left(1-\tilde{\delta}_k\right)\tilde{w}_k \mathrm{V}_k - (\tilde{\gamma}_k + \tilde{m}_k) I_k \\ D_t^\gamma \mathcal{A}_k(t) = \tilde{\delta}_k \tilde{w}_k' \mathrm{V}_k - (\tilde{\gamma}_k' + \tilde{m}_k)\mathcal{A}_k \\ D_t^\gamma \mathcal{R}_k(t) = \tilde{\gamma}_k I_k + \tilde{\gamma}_k' \mathcal{A}_k - \tilde{m}_k \mathcal{R}_k \\ D_t^\gamma \mathcal{M}_k(t) = \tilde{\xi}_k I_k + \tilde{\eta}\mathcal{A}_k - \tilde{\upsilon}\mathcal{M}_k \end{gathered} \tag{2}$$

associated to fuzzy initial condition, for $\alpha \in [0,1]$,

$$\tilde{Y}(0,\alpha) = (\underline{\mathrm{Y}}(0,\alpha), \overline{\mathrm{Y}}(0,\alpha)$$

$$\widetilde{\mathrm{V}}(0,\alpha) = (\underline{\mathrm{Y}}(0,\alpha), \overline{\mathrm{V}}(0,\alpha)$$
$$\tilde{I}(0,\alpha) = (\underline{I}(0,\alpha), \bar{I}(0,\alpha)$$
$$\tilde{\mathcal{A}}(0,\alpha) = (\underline{\mathcal{A}}(0,\alpha), \bar{\mathcal{A}}(0,\alpha)$$
$$\tilde{\mathcal{R}}(0,\alpha) = (\underline{\mathcal{A}}(0,\alpha), \bar{\mathcal{A}}(0,\alpha)$$
$$\widetilde{\mathcal{M}}(0,\alpha) = (\underline{\mathcal{A}}(0,\alpha), \bar{\mathcal{A}}(0,\alpha)$$

Regarding the above explanations and to address the current uncertain situation, we were motivated to propose a novel coronavirus infection system under fuzzy fractional calculus. In fact, considering the proposed model which also enhances the physical behavior of such an infection system, we ensure that the model is closer to the real behavior of a system evolving the general properties of RNA in Covid-19.

4.2 Main results

In the following section, the existence and uniqueness of solution to the subsequent fuzzy fractional model are discussed; and we provide the procedure for finding a semianalytic solution of model (2) by using fuzzy Laplace transform.

4.2.1 Existence and uniqueness

In this section, by the use of fixed point theory, the existence and uniqueness of the sub- sequent fuzzy fractional model is discussed. Consider the right-hand sides of model (2):

$$\begin{aligned}
&\psi\big(t, Y_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \\
&\qquad = \tilde{n}_k - \tilde{m}_k \mathrm{Y}_k - \tilde{b}_k \mathrm{Y}_k\big(I_k + \tilde{k}\mathcal{A}_k\big) - \tilde{b}_k \mathrm{Y}_k \mathcal{M}_k \\
&\Xi(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\
&\qquad = \big\{\tilde{b}_k \mathrm{Y}_k(I_k + \tilde{\delta}\mathcal{A}_k + \tilde{b}_l \mathcal{M}_k - \big(1 - \tilde{\delta}_k\big)\widetilde{w}_k \mathrm{V}_k - \tilde{\delta}_k \widetilde{w}'_k \mathrm{V}_k \\
&\qquad - \widetilde{m}_k \mathrm{V}_k\big\} \\
&f(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) = \big(1 - \tilde{\delta}_k\big)\widetilde{w}_k \mathrm{V}_k - (\tilde{\gamma}_k + \widetilde{m}_k) I_k \\
&g(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) = \tilde{\delta}_k \widetilde{w}'_k \mathrm{V}_k - (\tilde{\gamma}'_k + \widetilde{m}_k)\mathcal{A}_k \\
&h\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) = \tilde{\gamma}_k I_k + \tilde{\gamma}'_k \mathcal{A}_k - \widetilde{m}_k \mathcal{R}_k \\
&y\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) = \tilde{\xi}_k I_k + \tilde{\eta}'_k \mathcal{A}_k - \tilde{v}_k \mathcal{M}_k
\end{aligned}$$

where $\Psi, \Xi, f, g, h \text{ and } y$, are fuzzy functions. Thus, for $0 < \gamma \leq 1$, the given model (2) can be written as:

$$\begin{cases} D_t^{\gamma} \mathrm{Y}_k(t) = \psi\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \\ D_t^{\gamma} \mathrm{V}_k(t) = \Xi(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ D_t^{\gamma} I_k(t) = f(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ D_t^{\gamma} \mathcal{A}_k(t) = g(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ D_t^{\gamma} \mathcal{R}_k(t) = h\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \\ D_t^{\gamma} \mathcal{M}_k(t) = y\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \end{cases} \tag{3}$$

With fuzzy initial conditions

$$\begin{gathered} \tilde{Y}(0,\alpha) = (\underline{\mathrm{Y}}(0,\alpha), \overline{\mathrm{Y}}(0,\alpha) \\ \tilde{\mathrm{V}}(0,\alpha) = (\underline{\mathrm{V}}(0,\alpha), \overline{\mathrm{V}}(0,\alpha) \\ \tilde{I}(0,\alpha) = (\underline{I}(0,\alpha), \bar{I}(0,\alpha) \\ \tilde{\mathcal{A}}(0,\alpha) = (\underline{\mathcal{A}}(0,\alpha), \bar{\mathcal{A}}(0,\alpha) \\ \tilde{\mathcal{R}}(0,\alpha) = (\underline{\mathcal{R}}(0,\alpha), \bar{\mathcal{R}}(0,\alpha) \\ \tilde{\mathcal{M}}(0,\alpha) = (\underline{\mathcal{M}}(0,\alpha), \bar{\mathcal{M}}(0,\alpha) \end{gathered}$$

Now applying fuzzy fractional integral I^r and using initial conditions, we get

$$\begin{cases} \mathrm{Y}_k(t) = \tilde{Y}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} \psi\big(s, \mathrm{Y}_k(s), \mathrm{V}_k(s), I_k(s), \mathcal{A}_k(s), \mathcal{R}_k(s), \mathcal{M}_k(s)\big) ds \\ \mathrm{V}_k(t) = \tilde{\mathrm{V}}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} \Xi\big(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) ds \\ I_k(t) = \tilde{I}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} f\big(s, \mathrm{Y}_k(s), \mathrm{V}_k(s), I_k(s), \mathcal{A}_k(s), \mathcal{R}_k(s), \mathcal{M}_k(s)\big) ds \\ \mathcal{A}_k(t) = \tilde{\mathcal{A}}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} g\big(s, \mathrm{Y}_k(s), \mathrm{V}_k(s), I_k(s), \mathcal{A}_k(s), \mathcal{R}_k(s), \mathcal{M}_k(s)\big) ds \\ \mathcal{R}_k(t) = \tilde{\mathcal{R}}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} h\big(s, \mathrm{Y}_k(s), \mathrm{V}_k(s), I_k(s), \mathcal{A}_k(s), \mathcal{R}_k(s), \mathcal{M}_k(s)\big) ds \\ \mathcal{M}_k(t) = \tilde{\mathcal{M}}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-s)^{\gamma-1} y\big(s, \mathrm{Y}_k(s), \mathrm{V}_k(s), I_k(s), \mathcal{A}_k(s), \mathcal{R}_k(s), \mathcal{M}_k(s)\big) ds \end{cases} \tag{4}$$

Let us define a Banach space as $\mathbb{B} = \mathbb{B}_1 \times \mathbb{B}_2$ under the fuzzy norm:

$$\big\|\big(\mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big\|$$

$$\max_{t\in[0,T]} [|(\mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)|]$$

One can write equation (4) as

$$\tilde{\aleph}_k(t) = \tilde{\aleph}(0,\alpha) + \frac{1}{\Gamma(\gamma)} \int_0^t (t-\wedge)^{\gamma-1} \Theta(\wedge, \tilde{\aleph}_k(\wedge)) d\wedge \tag{5}$$

where

$$\widetilde{\aleph}_k(t) = \begin{cases} \Upsilon_k(t) \\ \mathrm{V}_k(t) \\ I_k(t) \\ \mathcal{A}_k(t) \\ \mathcal{R}_k(t) \\ \mathcal{M}_k(t) \end{cases} \qquad \widetilde{\aleph}_k(0,\alpha) = \begin{cases} \tilde{\Upsilon}(0,\alpha) \\ \tilde{\mathrm{V}}(0,\alpha) \\ \tilde{I}(0,\alpha) \\ \tilde{\mathcal{A}}(0,\alpha) \\ \tilde{\mathcal{R}}(0,\alpha) \\ \tilde{\mathcal{M}}(0,\alpha) \end{cases} \qquad \text{and}$$

$$\Theta\left(t, \widetilde{\aleph}_k(t)\right) = \begin{cases} \psi\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \\ \Xi(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ f(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ g(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)) \\ h\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \\ y\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big) \end{cases}$$

We make several assumptions on the nonlinear function $\Theta : \mathbb{B} \to \mathbb{B}$ as follows:
(C-1) There exists constant $K_{\aleph} > 0$ such that for each $\widetilde{\aleph}_{k_1}(t), \widetilde{\aleph}_{k_2}(t) \in \mathbb{B}$,

$$|\Theta\left(t, \widetilde{\aleph}_{k1}(t)\right) - \Theta\left(t, \widetilde{\aleph}_{k2}(t)\right)| \leq K_{\aleph}|\widetilde{\aleph}_{k1}(t) - \widetilde{\aleph}_{k2}(t)|$$

(C-2) There exist constants $M_{\aleph} > 0$ *and* $N_{\aleph} > 0$ t such that

$$\left|\Theta\left(t, \widetilde{\aleph}_k(t)\right)\right| \leq \mathcal{M}_{\aleph}|\widetilde{\aleph}_k(t)| + N_{\aleph}$$

Theorem 1. *Under Assumption (C-2), the considered model (3) has at least one solution.*

Proof. Let $\mathcal{A} = \{\widetilde{\aleph}_k(t) \in \mathbb{B} : \|\widetilde{\aleph}_k(t)\| \leq r\} \subset \mathbb{B}$ be a closed and convex fuzzy set, and $\psi : \mathcal{A} \to \mathcal{A}$ be a mapping defined as

$$\psi\left(\widetilde{\aleph}_k(t)\right) = \widetilde{\aleph}_k(0,\alpha) + \frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\Theta(\wedge, \widetilde{\aleph}_k(\wedge))d\wedge \tag{6}$$

For any $\widetilde{\aleph}_k(t) \in \mathcal{A}$,, we have

$$\left\|\psi\left(\widetilde{\aleph}_k(t)\right)\right\| = \max_{t\in[0,T]}\left|\widetilde{\aleph}_k(0,\alpha) + \frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\Theta(\wedge, \widetilde{\aleph}_k(\wedge))d\wedge\right|$$
$$\leq \left|\widetilde{\aleph}_k(0,\alpha) + \frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\Theta(\wedge, \widetilde{\aleph}_k(\wedge))\right| d\wedge$$

$$\leq \left|\widetilde{\aleph}_k(0,\alpha)\right| + \frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\left[\mathcal{M}_{\aleph}\left|\widetilde{\aleph}_k(t)\right| + N_{\aleph}\right], d\wedge$$

$$\leq \left|\widetilde{\aleph}_k(0,\alpha)\right| + \frac{\tau^{\gamma}}{\Gamma(\gamma+1)}\left[\mathcal{M}_{\aleph}\left|\widetilde{\aleph}_k(t)\right| + N_{\aleph}\right]$$

From the last inequality, we have $\psi(\mathcal{A}) \subset \mathcal{A}$, which implies that the operator ψ is bounded. Next we show that the operator ψ is completely continuous. For this, let $\varphi_1, \varphi_2 \in [0,T]$ be such that $\varphi_1 < \varphi_2$, then

$$\left\|\psi\left(\widetilde{\aleph}_k(t)\right)(\phi_2) - \psi(\widetilde{\aleph}_k(t)(\phi_1)\right\|$$

$$= \left|\frac{1}{\Gamma(\gamma)}\int_0^{\phi_2} (\phi_2-\wedge)^{\gamma-1}\Theta\left(\wedge,\widetilde{\aleph}_k(\wedge)\right)d\wedge - \frac{1}{\Gamma(\gamma)}\int_0^{\phi_1} (\phi_1-\wedge)^{\gamma-1}\Theta\left(\wedge,\widetilde{\aleph}_k(\wedge)\right)d\wedge\right|$$

$$\leq \left[\phi_2^{\gamma} - \phi_1^{\gamma}\right]\frac{\left[\mathcal{M}_{\aleph}\left|\widetilde{\aleph}_k(t)\right| + N_{\aleph}\right]}{\Gamma(\gamma+1)}$$

From the last inequality, we see that the right-hand side goes to zero as $\varphi_1 \to \varphi_2$. Hence,

$$\left\|\psi\left(\widetilde{\aleph}_k(t)\right)(\phi_2) - \psi\left(\widetilde{\aleph}_k(t)\right)(\phi_1)\right\| \to 0 \quad come\ \phi_2 \to \phi_1$$

Thus, the operator ψ is equicontinuous. By Arzela–Ascoli theorem, the operator ψ is completely continuous, also ψ is bounded as proved earlier. Therefore, system (3) has at least one solution by Schauder's fixed point theorem.

Theorem 2. *If Assumption (C-1) holds, then the considered system (3) has a unique solution if* $\tau^{\gamma}K_{\aleph} < \Gamma(\gamma+1)$.

Proof. Let $\widetilde{\aleph}_{k1}(t), \widetilde{\aleph}_{k2}(t) \in \mathbb{B}$ then

$$\left\|\psi\left(\widetilde{\aleph}_{k1}(t)\right) - \psi\left(\widetilde{\aleph}_{k2}(t)\right)\right\| \max_{t\in[0,T]}\left|\frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\Theta\left(\wedge,\widetilde{\aleph}_{k1}(\wedge)\right)d\right.$$

$$\left.\wedge - \frac{1}{\Gamma(\gamma)}\int_0^t (t-\wedge)^{\gamma-1}\Theta\left(\wedge,\widetilde{\aleph}_{k2}(\wedge)\right)d\wedge\right|$$

$$\leq \frac{\tau^{\gamma}}{\Gamma(\gamma+1)}\,K_{\aleph}\left|,\widetilde{\aleph}_{k1}(t) - \widetilde{\aleph}_{k2}(t)\right|$$

Hence ψ is a contraction. Hence, by Banach contraction theorem, system (3) has a unique solution.

4.2.2 Procedure for solution

Here a general method is provided in order to find the solution of the considered system by the fuzzy Laplace transform.
Taking fuzzy Laplace transform of (3) and using initial conditions, we get:

$$\left[D_t^{\gamma}[\mathrm{Y}_k(t)]\right] = L\left[\psi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L\left[D_t^{\gamma}[\mathrm{V}_k(t)]\right] = L\left[\Xi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L\left[D_t^{\gamma}[I_k(t)]\right] = L\left[f\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L\left[D_t^{\gamma}[\mathcal{A}_k(t)]\right] = L\left[g\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L\left[D_t^{\gamma}[\mathcal{R}_k(t)]\right] = L\left[h\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L\left[D_t^{\gamma}[\mathcal{M}_k(t)]\right] = L\left[y\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[\mathrm{Y}_k(t)] = s^{\gamma-1}\,\widetilde{\mathrm{Y}}(0,\alpha) + L\left[\psi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[\mathrm{V}_k(t)] = s^{\gamma-1}\,\widetilde{\mathrm{V}}(0,\alpha) + L\left[\Xi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[I_k(t)] = s^{\gamma-1}\,\tilde{I}(0,\alpha) + L\left[f\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[\mathcal{A}_k(t)] = s^{\gamma-1}\tilde{\mathcal{A}}(0,\alpha) + L\left[g\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[\mathcal{R}_k(t)] = s^{\gamma-1}\,\widetilde{\mathcal{R}}(0,\alpha) + L\left[h\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$s^{\gamma}L[\mathcal{M}_k(t)] = s^{\gamma-1}\widetilde{\mathcal{R}}(0,\alpha) + L\left[y\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L[\mathrm{Y}_k(t)] = \frac{1}{s}\,\widetilde{\mathrm{Y}}(0,\alpha) + \frac{1}{s^{\gamma}}L\left[\psi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L[\mathrm{V}_k(t)] = \frac{1}{s}\,\widetilde{\mathrm{V}}(0,\alpha) + \frac{1}{s^{\gamma}}L\left[\Xi\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L[I_k(t)] = \frac{1}{s}\,\tilde{I}(0,\alpha) + \frac{1}{s^{\gamma}}L\left[f\left(t, \mathrm{Y}_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\right)\right]$$

$$L[\mathcal{A}_k(t)] = \frac{1}{s}\,\tilde{\mathcal{A}}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[g\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

$$L[\mathcal{R}_k(t)] = \frac{1}{s}\,\widetilde{\mathcal{R}}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[h\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

$$[\mathcal{M}_k(t)] = \frac{1}{s}\,\widetilde{\mathcal{M}}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[y\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

The infinite series solution is given by:

$$\Upsilon_k(t) = \sum_{n=0}^{\infty} \Upsilon_{k_n}(t) \qquad \mathrm{V}_k(t) = \sum_{n=0}^{\infty} \mathrm{V}_{k_n}(t)$$

$$I_k(t) = \sum_{n=0}^{\infty} I_{k_n}(t) \qquad A_k(t) = \sum_{n=0}^{\infty} A_{k_n}(t)$$

$$\mathcal{R}_k(t) = \sum_{n=0}^{\infty} \mathcal{R}_{k_n}(t) \qquad \mathcal{M}_k(t) = \sum_{n=0}^{\infty} \mathcal{M}_{k_n}(t)$$

$$\Upsilon_k(t) I_k(t) = \sum_{n=0}^{\infty} \mathcal{Z}_{1,n}$$

$$\Upsilon_k(t) \mathcal{A}_k(t) = \sum_{n=0}^{\infty} \mathcal{Z}_{2,n}$$

$$\Upsilon_k(t) \mathcal{M}_k(t) = \sum_{n=0}^{\infty} \mathcal{Z}_{3,n}$$

where $\mathcal{Z}_{1_n}$, $\mathcal{Z}_{2_n}$ and $\mathcal{Z}_{3_n}$ are Adomian polynomials, representing nonlinear terms. So the last equation becomes

$$L\left[\sum_{n=0}^{\infty} \Upsilon_{k_n}(t)\right] = \frac{1}{s}\,\tilde{\Upsilon}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[\psi\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

$$L\left[\sum_{n=0}^{\infty} \mathrm{V}_{k_n}(t)\right] = \frac{1}{s}\,\tilde{\mathrm{V}}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[\Xi\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

$$\mathrm{L}\big[\textstyle\sum_{n=0}^{\infty} I_{k_n}(t)\big] = \frac{1}{s}\,\tilde{I}(0,\alpha) + \frac{1}{s^{\gamma}} L\big[f\big(t, \Upsilon_k(t), \mathrm{V}_k(t), I_k(t), \mathcal{A}_k(t), \mathcal{R}_k(t), \mathcal{M}_k(t)\big)\big]$$

$$\mathrm{L}\left[\sum_{n=0}^{\infty}\mathcal{A}_{k_n}(t)\right]=\frac{1}{s}\,\tilde{\mathcal{A}}(0,\alpha)+\frac{1}{s^{\gamma}}L\left[g\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]$$

$$\mathrm{L}\left[\sum_{n=0}^{\infty}\mathcal{R}_{k_n}(t)\right]=\frac{1}{s}\,\widetilde{\mathcal{R}}(0,\alpha)+\frac{1}{s^{\gamma}}L\left[h\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]$$

$$\left[\sum_{n=0}^{\infty}\mathcal{M}_{k_n}(t)\right]=\frac{1}{s}\,\widetilde{\mathcal{M}}(0,\alpha)+\frac{1}{s^{\gamma}}L\left[y\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]$$

Taking the inverse Laplace transform, we have

$$\sum_{n=0}^{\infty}\mathrm{Y}_{k_n}(t)=\tilde{\mathrm{Y}}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[\psi\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

$$\sum_{n=0}^{\infty}\mathrm{V}_{k_n}(t)=\tilde{\mathrm{V}}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[\Xi\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

$$\sum_{n=0}^{\infty}I_{k_n}(t)=\tilde{I}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[f\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

$$\sum_{n=0}^{\infty}\mathcal{A}_{k_n}(t)=\tilde{\mathcal{A}}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[g\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

$$\sum_{n=0}^{\infty}\mathcal{R}_{k_n}(t)=\tilde{\mathcal{R}}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[h\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

$$\sum_{n=0}^{\infty}\mathcal{M}_{k_n}(t)=\tilde{\mathcal{M}}(0,\alpha)+L^{-1}\left[\frac{1}{s^{\gamma}}L\left[y\left(t,\Upsilon_k(t),\mathrm{V}_k(t),I_k(t),\mathcal{A}_k(t),\mathcal{R}_k(t),\mathcal{M}_k(t)\right)\right]\right]$$

Comparing the terms on both sides, we consider the first two terms of the series

$$\begin{cases}\underline{\mathrm{Y}}_{k0}(t)=\underline{\mathrm{Y}}(0,\alpha), & \bar{S}_{k0}(t)=\overline{\mathrm{Y}}(0,\alpha),\\ \underline{\mathrm{V}}_{k0}(t)=\underline{\mathrm{V}}(0,\alpha), & \overline{\mathrm{V}}_{k0}(t)=\overline{\mathrm{V}}(0,\alpha),\\ \underline{I}_{k0}(t)=\underline{I}(0,\alpha), & \bar{I}_{k0}(t)=\overline{\mathrm{V}}(0,\alpha),\\ \underline{\mathcal{A}}_{k0}(t)=\underline{\mathcal{A}}(0,\alpha), & \bar{\mathcal{A}}_{k0}(t)=\bar{\mathcal{A}}(0,\alpha),\\ \underline{\mathcal{R}}_{k0}(t)=\underline{\mathcal{R}}(0,\alpha), & \bar{\mathcal{R}}_{k0}(t)=\bar{\mathcal{R}}(0,\alpha),\\ \mathcal{M}_{k0}(t)=\underline{\mathcal{M}}(0,\alpha), & \overline{\mathcal{M}}_{k0}(t)=\overline{\mathcal{M}}(0,\alpha),\end{cases} \tag{7}$$

$$\begin{cases} \underline{\mathrm{Y}}_{k1}(t) = L^{-1}\left[\frac{1}{s^{\gamma}} L\left[\widetilde{n}_k - \widetilde{m}_k \underline{\mathrm{Y}}_{k_0} - \widetilde{b}_k \underline{\mathrm{Y}}_{k_0}\left(\underline{I}_{k_0} + \widetilde{k}\underline{\mathcal{A}}_{k_0}\right) - \widetilde{b}_k \underline{\mathrm{Y}}_{k_0} \underline{\mathcal{M}}_{k_0}\right]\right] \\ \overline{\mathrm{Y}}_{k1}(t) = L^{-1}\left[\frac{1}{s^{\gamma}} L\left[\widetilde{n}_k - \widetilde{m}_k \overline{\mathrm{Y}}_{k_1} - \widetilde{b}_k \overline{\mathrm{Y}}_{k_0}\left(\bar{I}_{k_1} + \widetilde{k}\bar{\mathcal{A}}_{k_0}\right) - \widetilde{b}_k \overline{\mathrm{Y}}_{k_0} \overline{\mathcal{M}}_{k_0}\right]\right] \end{cases} \tag{8}$$

Similarly, we can find the other terms. Hence, the series solution of the considered system is given by

$$\begin{gathered} \underline{\mathrm{Y}}_k(t) = \underline{\mathrm{Y}}_{k_0}(t) + \underline{\mathrm{Y}}_{k_1}(t) + \underline{\mathrm{Y}}_{k_2}(t) + \cdots, \\ \overline{\mathrm{Y}}_k(t) = \overline{\mathrm{Y}}_{k_0}(t) + \overline{\mathrm{Y}}_{k_1}(t) + \overline{\mathrm{Y}}_{k_2}(t) + \cdots, \\ \underline{\mathrm{V}}_k(t) = \underline{\mathrm{V}}_{k_0}(t) + \underline{\mathrm{V}}_{k_1}(t) + \underline{\mathrm{V}}_{k_2}(t) + \cdots, \\ \overline{\mathrm{V}}_k(t) = \overline{\mathrm{V}}_{k_0}(t) + \overline{\mathrm{V}}_{k_1}(t) + \overline{\mathrm{V}}_{k_2}(t) + \cdots, \\ \underline{I}_k(t) = \underline{I}_{k_0}(t) + \underline{I}_{k_1}(t) + \underline{I}_{k_2}(t) + \cdots, \\ \bar{I}_k(t) = \bar{I}_{k_0}(t) + \bar{I}_{k_1}(t) + \bar{I}_{k_2}(t) + \cdots, \\ \underline{\mathcal{A}}_k(t) = \underline{\mathcal{A}}_{k_0}(t) + \underline{\mathcal{A}}_{k_1}(t) + \underline{\mathcal{A}}_{k_2}(t) + \cdots, \\ \bar{\mathcal{A}}_k(t) = \bar{\mathcal{A}}_{k_0}(t) + \bar{\mathcal{A}}_{k_1}(t) + \bar{\mathcal{A}}_{k_2}(t) + \cdots, \\ \underline{\mathcal{R}}_k(t) = \underline{\mathcal{R}}_{k_0}(t) + \underline{\mathcal{R}}_{k_1}(t) + \underline{\mathcal{R}}_{k_2}(t) + \cdots, \\ \overline{\mathcal{R}}_k(t) = \overline{\mathcal{R}}_{k_0}(t) + \overline{\mathcal{R}}_{k_1}(t) + \overline{\mathcal{R}}_{k_2}(t) + \cdots, \\ \underline{\mathcal{M}}_k(t) = \underline{\mathcal{M}}_{k_0}(t) + \underline{\mathcal{M}}_{k_1}(t) + \underline{\mathcal{M}}_{k_2}(t) + \cdots, \\ \overline{\mathcal{M}}_k(t) = \overline{\mathcal{M}}_{k_0}(t) + \overline{\mathcal{M}}_{k_1}(t) + \overline{\mathcal{M}}_{k_2}(t) + \cdots, \end{gathered} \tag{9}$$

4.3 Numerical results and discussion

We consider a table corresponding to the parameters involved in the model. Consider the proposed model (2) with initial conditions as given in Table 1:

$$\begin{aligned} \widetilde{\mathrm{Y}}(0,\alpha) &= (2\alpha - 1, 1 - 2\alpha) & \widetilde{\mathrm{V}}(0,\alpha) &= (2\alpha - 1, 1 - 2\alpha) \\ \tilde{\mathrm{I}}(0,\alpha) &= (2\alpha - 1, 1 - 2\alpha) & \tilde{\mathcal{A}}(0,\alpha) &= (2\alpha - 1, 1 - 2\alpha) \end{aligned}$$

$$\tilde{\mathcal{R}}(0,\alpha) = (2\alpha - 1, 1 - 2\alpha) \qquad \tilde{\mathcal{M}}(0,\alpha) = (2\alpha - 1, 1 - 2\alpha)$$

Notation	Parameters description	Numerical value
$\tilde{n}_k$	Birth rate	1
$\tilde{m}_k$	Death rate of infected population	$\frac{1}{76.79 \times 365}$
$\tilde{b}_k$	Transmission coefficient	0.05
$\tilde{b}_l$	Disease transmission coefficient	0.001231
$\tilde{w}_k, \tilde{w}'_k$	Signified incubation period	0.001243, 0.05
$\tilde{\gamma}_k, \tilde{\gamma}'_k$	Recovery rate of $I_k, \mathcal{A}_k$	0.09871, 0.0854302
$\tilde{\xi}, \tilde{\eta}$	Influence of virus from I_k and $\mathcal{A}_k$ to $\mathcal{M}_k$	0.0398, 0.01
$\tilde{v}$	Amount of asymptotic infection	0.1243
$\tilde{k}$	Transmissibility multiple	0.02
$\tilde{\eta}$	Elimination rate of virus from $\mathcal{M}_k$	0.01
Y_0	Initial value of susceptible	220 milioni
I_0	Initial value of infected	0.015 milioni
V_0	Initial value of exposed	100 milioni
$\mathcal{A}_0$	Initial value of asymptotically infected	0.60 milioni
$\mathcal{R}_0$	Initial value of recovered	0 milioni
$\mathcal{M}_0$	Initial value of reservoir	0.1 milioni

Table 1. Description of the parameters used in model (2)

Applying the proposed procedure to (2) and using initial conditions, we have

$$\begin{aligned}
\underline{\mathrm{Y}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathrm{Y}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha,\\
\underline{\mathrm{V}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathrm{V}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha,\\
\underline{\mathrm{I}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathrm{I}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha,\\
\underline{\mathcal{A}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathcal{A}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha,\\
\underline{\mathcal{R}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathcal{R}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha,\\
\underline{\mathcal{M}}_{\mathrm{k0}}(\mathrm{t},\alpha) &= 2\alpha - 1, \quad \overline{\mathcal{M}}_{\mathrm{p0}}(\mathrm{t},\alpha) = 1 - 2\alpha.
\end{aligned}$$

The second term of series solution is

$$\begin{cases}\underline{Y}_{k_1}(t,\alpha)=\left[\tilde{n}_k-\tilde{m}_k(2\alpha-1)-\tilde{b}_k(2\alpha-1)^2-k\tilde{b}_k(2\alpha-1)^2-\tilde{b}_k(2\alpha-1)^2\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \overline{Y}_{k_1}(t,\alpha)=\left[\tilde{n}_k-\tilde{m}_k(1-2\alpha)-\tilde{b}_k(1-2\alpha)^2-k\tilde{b}_k(1-2\alpha)^2-\tilde{b}_k(1-2\alpha)^2\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (10)$$

$$\begin{cases}\underline{V}_{k_1}(t,\alpha)=\begin{bmatrix}\tilde{b}_k-\tilde{m}_k(2\alpha-1)^2+k\tilde{b}_k(2\alpha-1)^2+\tilde{b}_k(2\alpha-1)^2-\\(1-\tilde{\delta}_k)\tilde{w}_k(2\alpha-1)-\tilde{\delta}_k w_k'(2\alpha-1)-\tilde{m}_k(2\alpha-1)\end{bmatrix}\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \overline{V}_{k_1}(t,\alpha)=\begin{bmatrix}\tilde{b}_k-\tilde{m}_k(1-2\alpha)^2+k\tilde{b}_k(1-2\alpha)^2+\tilde{b}_k(1-2\alpha)^2\\-(1-\tilde{\delta}_k)\tilde{w}_k(1-2\alpha)-\tilde{\delta}_k w_k'(1-2\alpha)-\tilde{m}_k(1-2\alpha)\end{bmatrix}\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (11)$$

$$\begin{cases}\underline{I}_{k_1}(t,\alpha)=\left[(1-\tilde{\delta}_k)\tilde{w}_k(2\alpha-1)-(\hat{Y}_p+\tilde{m}_k)(2\alpha-1)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \bar{I}_{k_1}(t,\alpha)=\left[(1-\tilde{\delta}_k)\tilde{w}_k(1-2\alpha)-(\hat{Y}_p+\tilde{m}_k)(1-2\alpha)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (12)$$

$$\begin{cases}\underline{\mathcal{A}}_{k_1}(t,\alpha)=\left[\tilde{\delta}_k\tilde{w}_k'(2\alpha-1)-(\tilde{Y}_k'+\tilde{m}_k)(2\alpha-1)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \bar{\mathcal{A}}_{k_1}(t,\alpha)=\left[\tilde{\delta}_k\tilde{w}_k'(1-2\alpha)-(\tilde{Y}_k'+\tilde{m}_k)(1-2\alpha)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (13)$$

$$\begin{cases}\underline{\mathcal{R}}_{k_1}(t,\alpha)=(2\alpha-1)\left[(\hat{Y}_k'+\hat{Y}_k'-\tilde{m}_k)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \overline{\mathcal{R}}_{k_1}(t,\alpha)=(2\alpha-1)\left[(\hat{Y}_k'+\hat{Y}_k'-\tilde{m}_k)\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (14)$$

$$\begin{cases}\underline{\mathcal{M}}_{k_1}(t,\alpha)=(2\alpha-1)\left[(\tilde{\xi}+\tilde{\eta}-\tilde{v})\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\\ \overline{\mathcal{M}}_{k_1}(t,\alpha)=(1-2\alpha)\left[(\tilde{\xi}+\tilde{\eta}-\tilde{v})\right]\frac{t^\gamma}{\Gamma(\gamma+1)}\end{cases} \quad (15)$$

For the sake of simplicity, assume that

$\underline{C}_1=\tilde{n}_k-\tilde{m}_k(2\alpha-1)-\tilde{b}_k(2\alpha-1)^2-k\tilde{b}_k(2\alpha-1)^2-\tilde{b}_k(2\alpha-1)^2,$
$\bar{C}_1=\tilde{n}_k-\tilde{m}_k(1-2\alpha)-\tilde{b}_k(1-2\alpha)^2-k\tilde{b}_k(1-2\alpha)^2-\tilde{b}_k(1-2\alpha),$
$\underline{C}_2=\tilde{b}_k(2\alpha-1)^2+k\tilde{b}_k(2\alpha-1)^2+\tilde{b}_k(2\alpha-1)^2-\left(1-\tilde{\delta}_k\right)\tilde{w}_k(2\alpha-1)-\tilde{\delta}_k w_k'(2\alpha-1)-\tilde{m}_k(2\alpha-1)$
$\bar{C}_2=\tilde{b}_k(1-2\alpha)^2+k\tilde{b}_k(1-2\alpha)^2+\tilde{b}_k(1-2\alpha)^2-\left(1-\tilde{\delta}_k\right)\tilde{w}_k(1-2\alpha)-\tilde{\delta}_k w_k'(1-2\alpha)-\tilde{m}_k(1-2\alpha)$
$\underline{C}_3=(1-\tilde{\delta}_k)\tilde{w}_k(2\alpha-1)-(\hat{Y}_p+\tilde{m}_k)(2\alpha-1)$
$\bar{C}_3=(1-\tilde{\delta}_k)\tilde{w}_k(1-2\alpha)-(\hat{Y}_p+\tilde{m}_k)(1-2\alpha)$
$\underline{C}_4=\tilde{\delta}_k\tilde{w}_k'(2\alpha-1)-(\tilde{Y}_k'+\tilde{m}_k)(2\alpha-1)$
$\bar{C}_4=\tilde{\delta}_k\tilde{w}_k'(1-2\alpha)-(\tilde{Y}_k'+\tilde{m}_k)(1-2\alpha)$
$\underline{C}_5=(2\alpha-1)\left[(\hat{Y}_k'+\hat{Y}_k'-\tilde{m}_k)\right]$
$\bar{C}_5=(2\alpha-1)\left[(\hat{Y}_k'+\hat{Y}_k'-\tilde{m}_k)\right]$
$\underline{C}_6=(2\alpha-1)\left[\tilde{\xi}+\tilde{\eta}-\tilde{v}\right]$
$\bar{C}_6=(1-2\alpha)\left[\tilde{\xi}+\tilde{\eta}-\tilde{v}\right]$

Now the third term of the series is

$$
\begin{cases}
\underline{Y}_{k_2}(t,\alpha) = \tilde{n}_k \dfrac{t^{\gamma}}{\Gamma(\gamma+1)} - \tilde{m}_k \underline{C}_1 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \left[\tilde{b}_k(2\alpha-1)(\underline{C}_3+\underline{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - \left[\tilde{b}_k(2\alpha-1)(\underline{C}_6+\underline{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{Y}_{k_2}(t,\alpha) = \tilde{n}_k \dfrac{t^{\gamma}}{\Gamma(\gamma+1)} - \tilde{m}_k \bar{C}_1 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \left[\tilde{b}_k(1-2\alpha)\bar{C}_3+\bar{C}_1\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - \left[\tilde{b}_k(1-2\alpha)(\bar{C}_6+\bar{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\underline{V}_{k_2}(t,\alpha) = \left[\tilde{b}_k(2\alpha-1)(\underline{C}_3+\underline{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} + \left[[\tilde{b}_k k(2\alpha-1)(\underline{C}_4+\underline{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - \underline{C}_2\left[(1-\tilde{\delta}_k) - \tilde{\delta}_k w_k' - \tilde{m}_k)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{V}_{k_2}(t,\alpha) = \left[\tilde{b}_k(1-2\alpha)(\bar{C}_3+\bar{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} + \left[[\tilde{b}_k k(2\alpha-1)(\bar{C}_4+\bar{C}_1)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - \bar{C}_2\left[(1-\tilde{\delta}_k) - \tilde{\delta}_k w_k' - \tilde{m}_k)\right] \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\underline{I}_{k_2}(t,\alpha) = (1-\tilde{\delta}_k)\tilde{w}_k \underline{C}_2 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - (\hat{Y}_k + \tilde{m}_k)\underline{C}_3 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{I}_{k_2}(t,\alpha) = (1-\tilde{\delta}_k)\tilde{w}_k \bar{C}_2 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - (\hat{Y}_k + \tilde{m}_k)\bar{C}_3 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\underline{\mathcal{A}}_{k_2}(t,\alpha) = \tilde{\delta}_k \tilde{w}_k' \underline{C}_2 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - (\hat{Y}_k' + \tilde{m}_k)\underline{C}_4 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{\mathcal{A}}_{k_2}(t,\alpha) = \tilde{\delta}_k \tilde{w}_k' \bar{C}_2 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} - (\hat{Y}_k' + \tilde{m}_k)\bar{C}_4 \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\underline{\mathcal{R}}_{k_2}(t,\alpha) = (\hat{Y}_k \underline{C}_3 - \hat{Y}_k' \underline{C}_4 - \tilde{m}_k \underline{C}_5) \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{\mathcal{R}}_{k_2}(t,\alpha) = (\hat{Y}_k \bar{C}_3 - \hat{Y}_k' \bar{C}_4 - \tilde{m}_k \bar{C}_5) \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\underline{\mathcal{M}}_{k_2}(t,\alpha) = (\tilde{\xi}\underline{C}_3 - \tilde{\eta}\underline{C}_4 - \tilde{v}\underline{C}_6) \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)} \\
\bar{\mathcal{M}}_{k_2}(t,\alpha) = (\tilde{\xi}\bar{C}_3 - \tilde{\eta}\bar{C}_4 - \tilde{v}\bar{C}_6) \dfrac{t^{2\gamma}}{\Gamma(2\gamma+1)}
\end{cases} \tag{16}
$$

In Figures 1–6 we presented comparisons of approximate fuzzy and approximate normal solutions for the considered model at the given uncertainty against various fractional order. We see that as the susceptible class value is decreasing, the exposed papulation increases and hence infection spreads with different rate due to various fractional order. Similarly, the death cases are increasing so the recovered class also grows and the asymptotically infected class also increases, and hence the population of virus in the reservoir is growing. From the figures we observe that fuzzyness along with fractional calculus provides global dynamics to such a kind of nonlinear problems where uncertainty lies in the data.

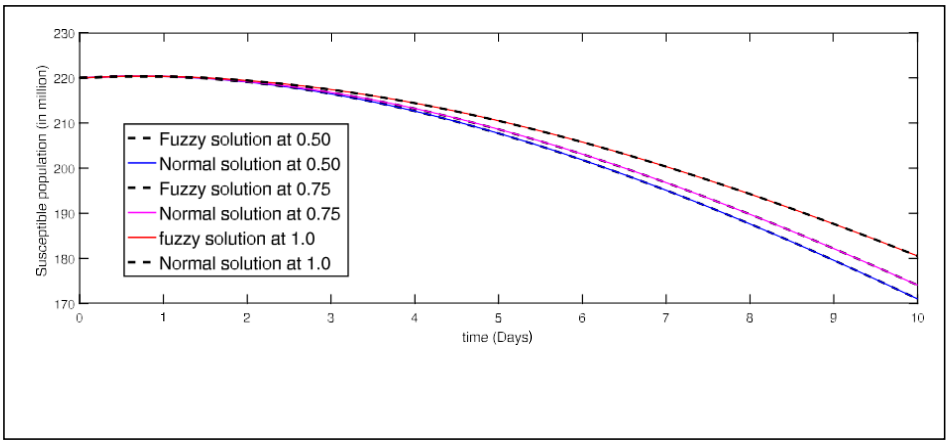


Figure 1: Comparison of approximate fuzzy and normal solution for susceptible compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order.

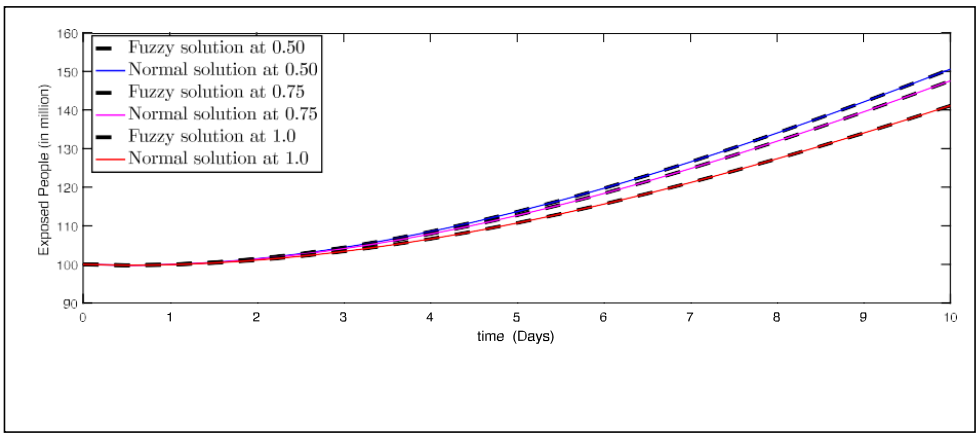


Figure 2: Comparison of approximate fuzzy and normal solution for exposed compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order.

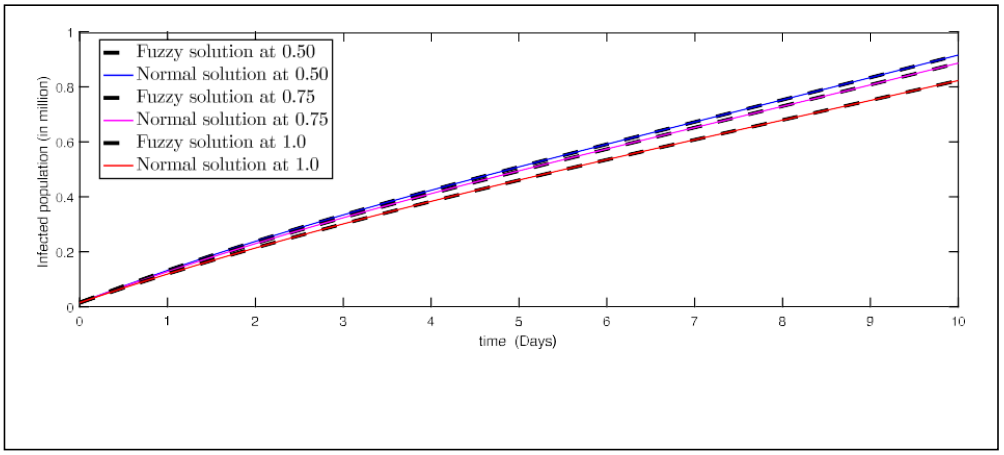


Figure 3: Comparison of approximate fuzzy and normal solution for infected compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order.

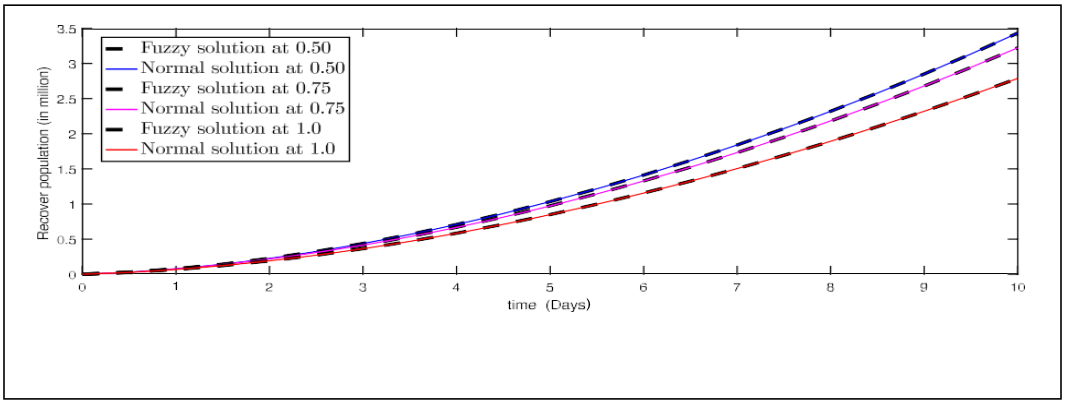


Figure 4: Comparison of approximate fuzzy and normal solution for recovered compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order.

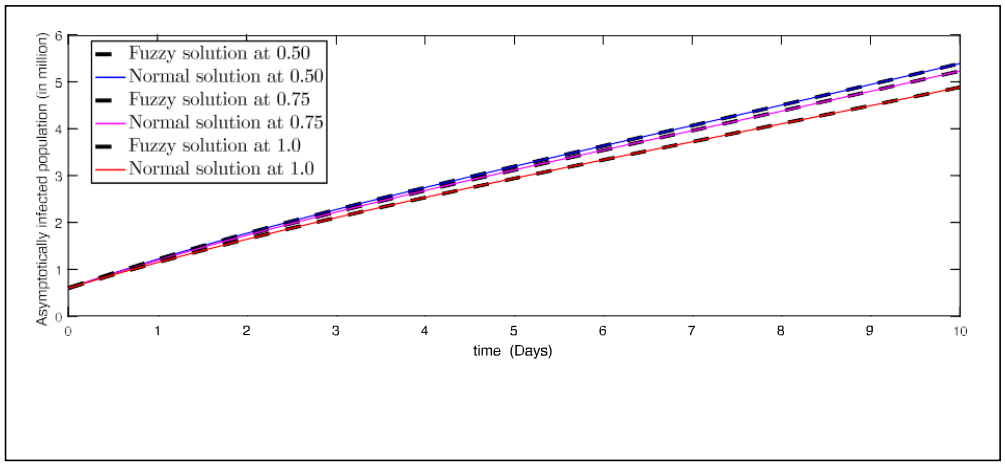


Figure 5: Comparison of approximate fuzzy and normal solution for asymptotically infected compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order.

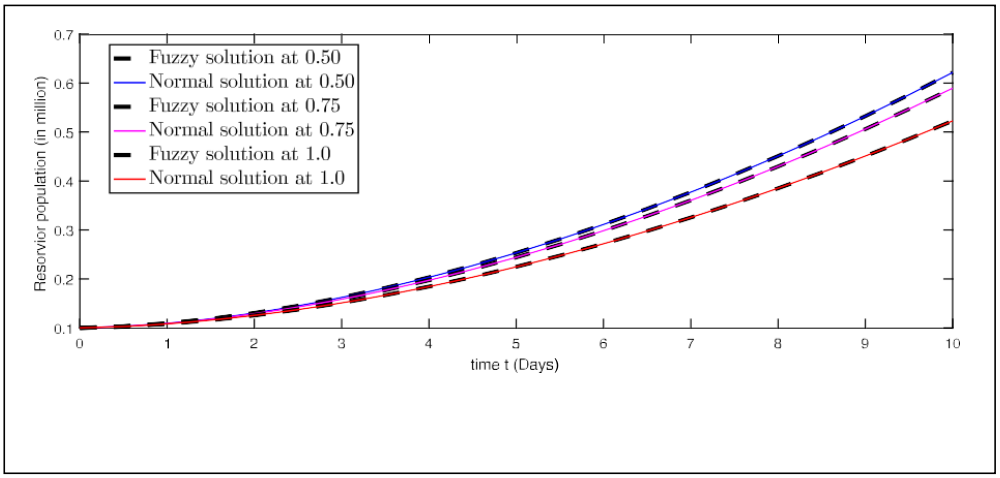


Figure 6: Comparison of approximate fuzzy and normal solution for reservoir compartment for three terms at the given uncertainty values $\alpha \in [0, 1]$ against various fractional order

Remark 1. *Regarding the provided results, it is clear that the lower bound is an increasing set-valued function and an upper bound is a decreasing one, which proves that the solutions are fuzzy numbers. Also, it is worthy to mention that for general cases, similar results can be obtained under fuzzy differentiability.*

Remark 2. *Considering the fact that stochastic and random parameters are more complex to address, the uncertainty can lead to an increase in the computation cost, so employing fuzzy concepts for modeling such real-world systems can be the most suitable choice.*

4.4 Conclusion

In this Chapter, we have demonstrated the existence and uniqueness of the solution to the fuzzy fractional order model of Covid-19 infection using the Banach fixed point theorem. We also established a proper procedure for the fuzzy Laplace transform coupled with Adomian decomposition method to obtain an approximate solution for the proposed model. We have presented comparisons between fuzzy and normal results up to three terms to depict the efficiency of this approach. We observed that fuzzyness coupled with fractional calculus approach excellently produced global dynamics of those problems where uncertainty lies in the data.

CHAPTER 5

An Integrated Decision Approach for Energy Procurement and Tariff Definition for Prosumers Aggregations

5.1 Introduction

In recent liberalized electric energy markets, the role of end-users is rapidly changing, also because of the availability of production and storage resources at an affordable cost, and they are becoming "prosumers", that is consumers and producers at the same time. In order to allow prosumers to have a more active role and to benefit from market opportunities, many coalitions are emerging in several countries. These users groupings, also called microgrids [1] or Virtual Power Plants (VPP) [2] according to the availability of resources, aim at creating a sort of cooperative system, aggregating a number of prosumers to act as a single user w.r.t. power market and to fully exploit the available resources [3], usually coordinated by a single entity, the "aggregator" in the following. According to this setting, single users can buy and sell energy by interacting just with the aggregator, which is responsible of offering these users more convenient tariffs w.r.t. those they could find otherwise and to make sustainable from an economic point of view the aggregation as a whole. The effective energy management for the entire coalition imposes the definition of several decisions both at a strategic/tactical and operational level. As regards the former, the main decision processes refer to the energy procurement planning considering the different available sources (bilateral contracts, self-production and day-ahead electricity market or DAEM) and the definition of tariff structures for buying and selling energy by prosumers within the coalition. However, the context in which these decisions are set is characterized by a high level of uncertainty. The overall energy demand is typically difficult to predict exactly since it refers to future needs. Market prices are known only after all producers and consumers submit their selling and bidding curves [4], and, thus, they are unknown in advance. Furthermore, the power generation from renewable resources cannot be accurately predicted because it can depend, for example, on the weather conditions.

To the best of our knowledge, the decision problem briefly described above has not been investigated as a whole, but recent contributions have been proposed to handle separately the energy procurement and the tariff definition, with different methodological approaches. As far as the medium-long term energy procurement, many contributions, like for example in [5] and [6], propose deterministic approaches. Among these, in [7] authors solve a deterministic procurement problem over a medium-term time horizon. The model considers a set of bilateral contracts, hourly changing spot prices and the possibility of self-producing energy.

More recently, in many contributions the uncertainty is explicitly modelled. For example, in [8] the authors extend the previous contribution by addressing a similar problem for a shorter time horizon and accounting for cost volatility by an estimation of the covariance of the spot price. In [9] the authors consider the perspective of a large consumer that owns a limited self-production facility and propose a stochastic programming model for the energy procurement plan.
They include a risk measure in the objective function in order to find a trade-off between risk and expected cost but without considering the uncertainty which affects energy demand. A similar trade-off is also investigated in [10] by the use of the information gap theory with the aim of evaluating the robustness of the solutions against high spot prices or high procurement costs. In [11] the authors propose a recourse two-stage formulation for the procurement problem over a short-term planning horizon. The problem is solved in a rolling-horizon fashion using each time the more updated information. More recently, Beraldi *et al.* [12] analyse the procurement problem from the perspective of an aggregator and propose a multi-period two-stage stochastic programming formulation. In [13] and [14] authors deal with the procurement problem under uncertainty by adopting the paradigm of joint chance constraints to define reliable plans that are feasible with a high probability level. Risk exposure has been also controlled by means of a mean- risk objective function, with the Conditional Value at Risk (CVaR) as risk measure. In [15] authors address the optimal operation of power-intensive plants by proposing a stochastic multi-stage mixed-integer linear programming model that considers uncertain product demands and equipment breakdowns to determine a solution to the integrated electricity procurement and production scheduling problem.
The optimal tariff structure definition problem has been addressed mainly by the point of view of retailers, which usually act as intermediary between the market and the end-users. Also this problem, however, is highly affected by uncertain factors, like market prices and energy demand. For example, Carrin *et al.* [16] provide a stochastic programming model that allows an electricity retailer to engage in medium-term forward contracts and to optimally set selling prices to customers, with the final aim of maximizing the expected profit given a pre-specified risk level on profit volatility. In [17] authors propose a dynamic and flexible tariff structure for a distribution company that protects customers against the excessive fluctuations of the wholesale market prices, by means of a two-stage pricing scheme with a static and a dynamic component. In [18], the selling price problem for a retail energy provider (REP) is addressed by the robust optimization approach.
Since the definition of tariff for users' coalition is a quite recent problem, the number of scientific contribution is still limited. In Nojavan *et al.* [19] a scenario based approach for a smart grid where the selling tariff is determined on the basis of a real-time pricing is proposed. In [20] the authors have carried out a comparison among 12 different tariffs schemes to propose to different residential

microgrids, analysing the impact of some relevant tariff features. More recently, In [21] a bilevel programming approach to electricity tariff optimization for demand response management (DRM) in smart grids, using a generic game-theoretic model, is proposed. In particular, the author presents a primal-dual reformulation for convert the bilevel optimization problem into a single-level quadratically constrained quadratic program (QCQP), with a linear programming algorithm as solution method. In [22] authors propose a stochastic programming model based on the paradigm of integrated chance constraints for the tariff definition problem for a coalition of prosumers, but considering already set the procurement plan.

In this work, we propose a decision approach for the energy procurement and the tariff definition in an integrated fashion. We have adopted the stochastic programming framework in order to effectively manage the uncertainty, which characterizes the overall decision process. Starting from the results proposed in [12] and [22], a first contribution consists in the modelling of the decision process as a whole, exploiting the inherent relations between the decisions that the aggregator has to deal with, that is how much energy to procure from each available source and the tariff scheme to offer to prosumers within the coalition, and considering the same uncertainty representation. Another innovative issue is related to the definition of "quasi-dynamic" tariff structures, which present some components related to the observation of uncertain parameters (overall demand and production from renewable systems, market prices). This feature can be a very useful stimulus for an efficient behaviour of prosumers, as a sort of demand responsiveness strategy. The economic convenience for prosumers to stay within the coalition is also explicitly imposed, so to guarantee the definition of tariffs which can be attractive w.r.t. other opportunities. A modern risk measure is considered in the mathematical model as well, in order to control the overall sustainability of the coalition from a financial standpoint.

The rest of the chapter is organized as follows: the following Section 5.2 introduces the decision problem characteristics and the mathematical formulation. Section 5.3 presents and discusses the numerical experience that has been carried out in order to evaluate the effectiveness of the proposed model. Concluding remarks and future research developments are discussed in Section 5.4.

5.2 An integrated decision model

As already stated, we consider the problem faced by the aggregator of a medium size prosumers' aggregation and a time lapse from several months to one year that can be considered a "usual" planning horizon when the procurement and management plan for a coalition at a strategic level is defined. The energy procurement regards the choice of the optimal mix to procure from the available sources, that is from bilateral contracts, production from both conventional and

renewable systems, and the day-ahead market (DAEM). The tariff definition is related to the design of price schemes to apply to different prosumers' groups for both selling and buying energy within the coalition. For each month of the planning horizon, the hours of the different days are articulated into a set F of time-of-use (TOU) blocks (e.g. peak, intermediate and off-peak). Under this assumption the elementary time period is the [month-TOU block] pair, indexed with $[t, f]$. This choice is coherent with a medium-long term planning, since many of the decisions related to [month-TOU block] pairs will be an input for the operative energy management, which is performed with a day-by-day frequency. However, other choices can be set without any impact on the model consistency.

We assume that the coalition energy needs can be covered by bilateral contracts, self-production from renewable and/or no renewable systems and by the day-ahead market, and that the coalition can have prosumers with different characteristics (for example, residential or industrial users), which can be clustered in a certain set of groups H. As already stated, decisions refer to both the procurement plan and the tariff structures to offer to the prosumers. More in detail, as regards the former, the amount of energy to procure from each selected bilateral contract, the energy to produce from traditional systems and the quantity to buy and sell on the day-ahead market. For the tariff structure, the unit price of energy bought and sold for each group of prosumers within the coalition.

Let N be the set of bilateral contracts to be evaluated, assuming that at most N^{max} can be accepted. For each contract $i \in N$, we denote by LB_{itf} and UB_{itf} the lower and upper bound for energy assumed for the TOU block f of month t, if i is selected, and by PB_{itf} the unit price for purchasing energy for the TOU block f of month t. We also consider a fixed component FB_i that accounts for administrative costs for contract acceptance. The aggregator is assumed to have a set of traditional production systems, which he can use for producing energy for each TOU block f of each month t, at a production cost G and with an upper bound Q_f^{max} for each TOU block.

In order to explicitly address the inherent stochastic nature of the decision problem, related to demand, production from renewable sources and market prices, we have adopted the 2-stage multiperiod stochastic programming framework. Here, the uncertain parameters are modelled as random variables defined on a given probability space $(\Omega, \mathcal{F}, \mathcal{P})$. Under the assumption of discrete distributions, the future uncertain evolution of DAEM prices, electricity demands and renewable production levels is represented by a set S of scenarios, each occurring with probability π_s. The eventual correlation between these random variables is not explicitly modelled within the mathematical model we present, but it can be eventually represented in the scenario generation phase. We denote by D_{tfh}^s the uncertain overall demand of customers of group h and by P_{tf}^s the unitary price from purchasing energy from the market at month t and TOU block f under scenario s. Moreover, we indicate with R_{tf}^s the overall production from renewable

systems. Since our model also considers the possibility to sell energy in excess to the demand, let B_{tf}^{s} be the selling price for the market zone in which the coalition is located. We observe that demand and supply prices could be different, according to the specific rules of several national markets like the Italian one [4].
As regards tariff structures, our idea is to define a fixed and a variable component for both the buying and the selling side, with the second one related to the specific real-life evolution of market prices. This approach is motivated by two issues. On the one hand, the need to keep the coalition sustainable from a financial standpoint imposes to link as much as possible tariffs with the specific evolution of unpredictable demand, production and market dynamics. On the other hand, this articulation can stimulate prosumers to better schedule energy consumption in order to have economic benefits: if a great part of the overall demand can be satisfied by internal production, preferably by renewable systems, and/or by bilateral contracts market operations can be reduced and also tariffs can be more convenient for prosumers. An evidence of this effect will be shown in Section 5.3.
According with the two-stage framework, in this case first-stage variables are related to the procurement plan and to the fixed tariff components, whereas second-stage decisions stand for the variable tariff components and corrective actions that guarantee the fulfilment of the energy needs by drawing energy from the balance market. More in detail, as regards first-stage variables, let k_i be the binary decision variable associated to the acceptance of bilateral contract i and x_{itf} the amount of electricity to purchase through contract i in TOU block f of month t. As regards market operations, we indicate with y_{tf} and w_{tf} the amount to buy and to sell on the DAEM for month t and TOU block f, respectively. Q_{tf} is the amount to produce at time *t* and TOU block f from controllable production units. For the tariff scheme we denote with $TC1_{tfh}$, the unit base price for prosumers in group h for the amount of consumed energy for month t and TOU block f with α_h, that is the percentage of the deviation of the market price from the average value adopted for consumers group h. Similarly, let $TP1_{tf}$ be the unit base price for energy sold by a prosumer and μ the percentage of deviation of market price for the selling side. As regards second-stage variables, let Δ_{tf}^{s-} and Δ_{tf}^{s+} represent the amounts of energy required to balance (excess/shortage) the aggregated needs under scenario *s*, traded on the secondary market. Moreover, as regards tariff components, $TC2_{tfh}^{s}$ unit variable price for energy bought by prosumers in group h and $TP2_{tf}^{s}$ the unit variable price on the selling side, for month t TOU block f.

5.2.1 Constraints

The operative conditions that limit the decision process have been modelled by means of the constraints reported below.

$$\sum_{i\in N} x_{itf} + y_{tf} + Q_{tf} + R_{tf}^{s+} + \Delta_{tf}^{s+} = D_{tfh}^{s} + w_{tf} + \Delta_{tf}^{s-} \quad \forall t, \forall f, \forall s \tag{1}$$

$$w_{tf} \leq Q_{tf} + R_{tf}^{s} \quad \forall t, \forall f, \forall s \tag{2}$$

$$Q_{tf} \leq Q_{f}^{max} \quad \forall t, \forall f \tag{3}$$

$$LB_{itf} k_i \leq x_{itf} \leq UB_{itf} k_i \quad \forall t, \forall f, \forall i \tag{4}$$

$$\sum_{i\in N} k_i \leq N^{max} \tag{5}$$

$$\sum_{t\in T}\sum_{f\in F}\left(TC1_{tfh} + TC2_{tfh}^{s}\right) D_{thf}^{s} \leq (1+\Theta^{B}) \sum_{t\in T}\sum_{f\in F} P_{tf}^{s} D_{tfh}^{s} \quad \forall h, \forall s \tag{6}$$

$$\sum_{t\in T}\sum_{f\in F}\left(TP1_{tf} + TP2_{tf}^{s}\right) R_{tf}^{s} \geq (1+\Theta^{S}) \sum_{t\in T}\sum_{f\in F} B_{tf}^{s} R_{tf}^{s} \quad \forall s \tag{7}$$

$$\Upsilon_{h}^{LB} \sum_{s\in S} \pi_s P_{tf}^{s} \leq TC1_{tfh} + \sum_{s\in S} \pi_s TC2_{tfh}^{s} \leq \Upsilon_{h}^{UB} \sum_{s\in S} \pi_s P_{tf}^{s} \quad \forall t, \forall f, \forall h \tag{8}$$

$$\eta_{h}^{LB} \sum_{s\in S} \pi_s B_{tf}^{s} \leq TP1_{tf} + \sum_{s\in S} \pi_s TP2_{tf}^{s} \leq \eta_{h}^{UB} \sum_{s\in S} \pi_s B_{tf}^{s} \quad \forall t, \forall f \tag{9}$$

$$TC2_{tfh}^{s} = \alpha_h (P_{tf}^{s} - \sum_{r\in S}\left(\pi_r P_{tf}^{r}\right)) \quad \forall t, \forall f, \forall h, \forall s \tag{10}$$

$$\alpha_h \leq \tau_h \quad \forall h \tag{11}$$

$$TP2_{tf}^{s} = \mu (B_{tf}^{s} - \sum_{r\in S}\left(\pi_r B_{tf}^{r}\right)) \quad \forall t, \forall f, \forall s \tag{12}$$

$$\mu \leq \varphi \tag{13}$$

$$x_{itf} \geq 0 \quad \forall t, \forall f, \forall i \tag{14}$$

$$k_i \in \{0,1\} \quad \forall i \tag{15}$$

$$y_{tf} \geq 0 \qquad \forall t, \forall f \tag{16}$$

$$w_{tf} \geq 0 \qquad \forall t, \forall f \tag{17}$$

$$Q_{tf} \geq 0 \qquad \forall t, \forall f \tag{18}$$

$$TC1_{tfh} \geq 0 \quad \forall t, \forall f, \forall h \tag{19}$$

$$TC2^{s}_{tfh} \geq 0 \quad \forall t, \forall f, \forall h, \forall s \tag{20}$$

$$\alpha_h \geq 0 \qquad \forall h \tag{21}$$

$$TP1_{tf} \geq 0 \quad \forall t, \forall f \tag{22}$$

$$TP2^{s}_{tf} \geq 0 \quad \forall t, \forall f, \forall s \tag{23}$$

$$\mu \geq 0 \tag{24}$$

$$\Delta^{s-}_{tf}, \Delta^{s+}_{tf} \geq 0 \quad \forall t, \forall f, \forall s \tag{25}$$

Equation (1) represents the energy balance for each period and under each scenario between overall demand and procurement sources, considering also the possibility to sell on the market the energy in excess. Condition (2) limits to the production level the amount that can be sold on the market, so to avoid a too speculative attitude. Constraint (3) represents a technological upper bound on the amount that can be produced by controllable systems in each period, while condition (4) limits the amount that can be absorbed by each active bilateral contract. In order to avoid a too big administrative effort, we limit the number of active bilateral contracts by means of (5).

Conditions (6) and (7) impose the economic convenience for buying and selling energy within the coalition. In many energy systems individual prosumers cannot operate directly on the market, so they interface with other operators, like distribution companies on the buying side. For this reason the convenience of prosumers to stay within the coalition has been represented considering the mark-up for these operators. According to this assumption, on the buying side, (6) states that the overall cost for prosumers of group h has to be lower than the cost that they should pay to a distribution company, considering as unit price the market price plus a certain percentage mark-up Θ^B. Similarly, the convenience on the selling side modelled by (7) states that the overall revenue for energy sold should

be greater than the revenue obtained outside, considering as unit price the market price minus a certain percentage mark-up Θ^S. With (8) we define the range for the buying side overall tariff, with bounds that are proportional to the expected value of the market price by means of a percentage lower (Υ_h^{LB}) and upper bound (Υ_h^{UB}). Similar restrictions are imposed to the overall selling side tariff by means of (9), with η^{LB} and η^{UB} representing the percentage lower and upper bound for selling side overall tariff.
Equation (10) defines the variable unit price for the buying side for each prosumer group as a percentage of the deviation from the expected value of the observed market price for each time-block of each month. The percentage α_h is constrained to be lower than a threshold τ_h (11), that can be defined according to the nature of the prosumer group. For example, residential users can be subject to a lower variability w.r.t. commercial ones. Similar conditions are defined also for the selling-side variable component by means of (12) and (13), with μ limited by a certain percentage threshold φ. Finally, constraints (14) - (25) define the nature of decision variables.

5.2.2 Objective function

The final aim of the aggregator is to define a procurement plan and a tariff scheme that guarantee the economic sustainability of the coalition as a whole and the convenience of each prosumer at the same time, also considering the uncertainty which characterizes the decision process. We assume that he can act like an investor which aims at both maximizing profits, which in this case represents the economic sustainability, and reducing risk expo- sure. Since these two objectives can be potentially conflicting in a scenario based formulation, we have adopted a mean-risk structure for the objective function:

$$max\ \left[(1\ -\ \lambda)E\ [\Upsilon] +\ \lambda CVaR_{\beta}\right] \tag{26}$$

where the first term is the expected value of the overall profit of the aggregator (i.e. of the coalition) and the second one is the Conditional Value at Risk (CVaR), a modern risk measure widely adopted in several domains [23]. The parameter λ, ranging in $[0, 1]$, represents the risk-aversion attitude of the decision-maker. A high value of λ stands for a "conservative" planning, while if $\lambda\ =\ 0$ a risk-neutral position is modelled. The overall profit under each scenario *s* is given by the sum of different components:

$$\Upsilon^s =\ V_{TC}^s + V_{MKT}^s +\ C_{TP}^s -\ C_{BC} - C_{Prod} - C_{MKT}^s - C_{Err}^s \quad \forall s \tag{27}$$

$$V_{TC}^{s} = \sum_{t\in T}\sum_{f\in F}\sum_{h\in H}\left(TC1_{tfh} + TC2_{tfh}^{s}\right) D_{tfh}^{s} \qquad \forall s \qquad (28)$$

$$V_{MKT}^{s} = \sum_{t\in T}\sum_{f\in F} B_{tf}^{s}\, w_{tf} \qquad \forall s \qquad (29)$$

$$C_{TP}^{s} = \sum_{t\in T}\sum_{f\in F}\left(TP1_{tf} + TP2_{tf}^{s}\right) R_{tf}^{s} \qquad \forall s \qquad (30)$$

$$C_{BC} = \sum_{i\in N}\left[FB_i k_i + \sum_{t\in T}\sum_{f\in F} PB_{itf} x_{itf}\right] \qquad (31)$$

$$C_{Prod} = \sum_{t\in T}\sum_{f\in F}\left(GQ_{tf}\right) \qquad (32)$$

$$C_{MKT}^{s} = \sum_{t\in T}\sum_{f\in F} P_{tf}^{s}\, y_{tf} \qquad \forall s \qquad (32)$$

$$C_{MKT}^{s} = \sum_{t\in T}\sum_{f\in F} P_{tf}^{s}\, y_{tf} \qquad \forall s \qquad (33)$$

$$C_{Err}^{s} = \sum_{t\in T}\sum_{f\in F}\left(Z_{tf}^{s+}\Delta_{tf}^{s+} - Z_{tf}^{s-}\Delta_{tf}^{s-}\right) R_{tf}^{s} \qquad \forall s \qquad (34)$$

Here, C_{BC} is the cost related to procurement form bilateral contracts, and is made of both a variable and a fixed amount, as already stated, while C_{Prod} represents the cost of the energy produced by conventional production systems. These cost components are deterministic, since they do not depend on the outcomes of uncertain parameters. On the contrary, all the other terms are scenario dependent. V_{TC}^{s} and C_{TP}^{s} represent the revenue and the cost due to the energy sale and purchase by prosumers within the coalition, while V_{MKT}^{s} and C_{MKT}^{s} are the revenue and the cost for market operations under scenario s, that is for a specific evolution of market prices. Finally, in (34)C_{Err}^{s} is the cost for the energy balance on the secondary market, with Z_{tf}^{s+} and representing the unit price on this market for the buying and selling side respectively.

As already stated, we have adopted a risk measure the CVaR, which allows to control the "tail" risk, that is the expected value of the losses exceeding the Value at Risk. Moreover, CVaR has a lot of useful properties, mainly its coherency and consistency with the second order stochastic dominance [25], and is easily

tractable from a computational standpoint. In the following, rather than considering the distribution of the losses, we deal with the distribution of the overall profit. In this case, for a given confidence level β $(0,1)$, the VaR is the $(1-\beta)$-quantile of the profit distribution, whereas the CVaR measures the expected worst-case profits less than VaR:

$$CVaR_{\beta} = E\left[Y | Y \leq VaR_{\beta}\right] \tag{35}$$

where Y denotes the average profit for the entire planning horizon. Under the assumption of discrete distribution function, the risk measure can be reformulated as follows:

$$CVaR_{\beta} = VaR - \frac{1}{1-\beta}\sum_{s \in S} \pi^{s} \left[VaR - Y^{s}\right] + \tag{36}$$

where Y^{s} is the profit under scenario *s* and the operator $[.]+$ is used to denote the maximum between 0 and $(VaR - Y^{s})$. This last term can be easily linearised by the introduction of support non negative variables σ^{s} satisfying the following constraints:

$$\begin{aligned} &\sigma^{s} && \\ &\geq \ VaR - \gamma^{s} && \forall s \end{aligned} \tag{37}$$

where the VaR represents a free decision variable.
As already stated, the resulting model belongs to the class of mixed- integer multiperiod two-stage stochastic programming problems. It is worth- while noting that the binary variables are just related to the selection of bilateral contracts, thus their number is limited for real-life decision problems and do not impact too much on the solution process. A complete list of symbols used to represent parameters and variables is reported in Appendix A.

5.3 Computational experience

In this section we report on the computational experience carried out in order to validate the effectiveness of the proposed decision approach. The model has been implemented by using GAMS 24.7.1[3] as algebraic modelling system, with CPLEX 12.6.1[4] as solver for mixed integer linear problems, and MATLAB R2015a[5] for the scenario generation and parameters set-up phases. As testbed we have

[3] www.gams.com
[4] https://www.ibm.com/it-it/analytics/cplex-optimizer
[5] www.mathworks.com

considered a "virtual" coalition, made up by 3 prosumers groups (residential, commercial, public utility), with a set of small photovoltaic plants with an overall capacity of 2 MWp and a set of conventional production systems with the same nominal capacity. Starting from the available data, which refer to a limited number of prosumers of each group, the expected values of the aggregated demand and production from renewable systems for each TOU block and each month have been calculated by scaling up the values calculated on the basis of historical series. In Appendix B the expected values of demand and production for the coalition as a whole are reported. We are considering a planning horizon of 12 months and 3 TOU blocks, F1 (peak), F2 (intermediate) and F3 (off-peak), according to the Italian electricity market. As already stated, a different choice for the planning horizon can be suitable as well. We have also considered a set of 10 bilateral contracts with different characteristics, the same adopted in [12]. In Appendix C we have reported the price components for each bilateral contract considered. We have fixed to 5 the maximum number of contracts that can be activated.

As regards the uncertainty representation, the scenario set for each test case has been generated by using a mean-reverting process for the market prices (see [26]) and by adopting random increment or decrement to the expected values for overall demand and production from renew- able systems. We have considered overall demand of the coalition, renewable production and market prices as independent random variables, that is the aggregation acts like a price-taker operator. According to this assumption, the whole scenario set has been generated by merging the scenario sets obtained for each random variable independently through Cartesian product and then by adopting a scenario reduction technique like the one proposed in [24]. For most test instances we have considered 500 scenarios along the time horizon.

Several computational experiments have been carried out in order to assess the effectiveness of the proposed approach and to evaluate the impact of different issues on the decision process.

5.3.1 Solution analysis

First of all, the solution of a single instance of the problem provides managerial insights about the optimal procurement mix and the tariff structure for the entire coalition. In Figure 1 we report the amount of energy to pro- cure from the different available sources for each TOU block of each month, obtained by solving the problem with an intermediate risk-aversion attitude parameter value ($\lambda = 0.5$). As evident, the optimal procurement mix is quite variable over the time horizon, with for example the percentage of energy bought from bilateral contracts that for some months is higher than the quantity purchased from the market also the amount procured from each bilateral contract is quite variable, as reported in Figure 2, and this is mainly due to the different unit cost proposed by each contract

for each [month-TOU block] pair.

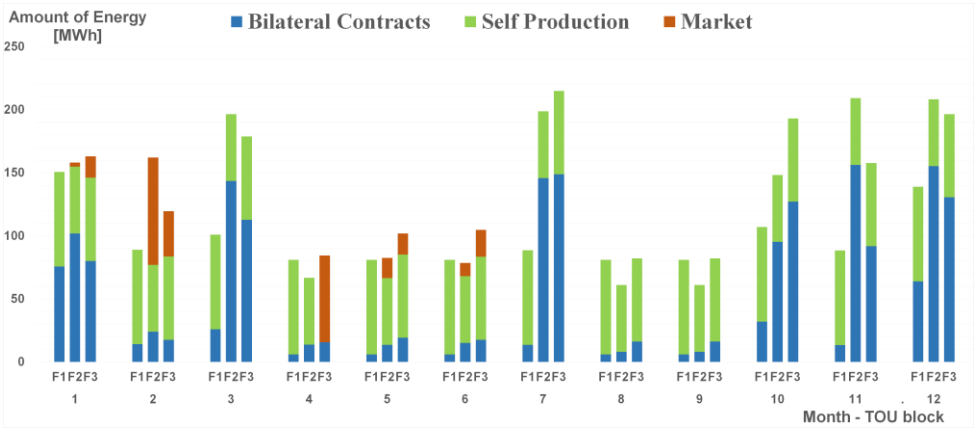


Figure 1: Optimal procurement plan

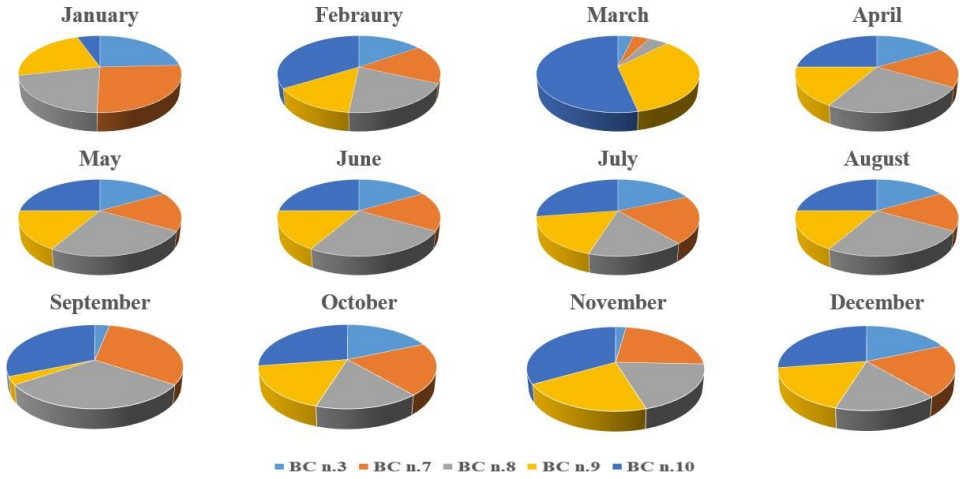


Figure 2: Energy procured from bilateral contracts for $\lambda = 0.5$

The average values of overall unit tariffs related to TOU block F1 for both the buying and the selling side are described in Table 1 and Table 2, where we report also the market alternative, that is the cost (for the buying side) and the revenue (for the selling side) that a single prosumer would have outside the coalition. These values have been calculated by considering the mark-up of the retailer equal to 35% both for the buying and the selling side.

Month	Residential	Commercial	Public Util.	Market
Jan	70.15	72.95	75.16	75.55
Feb	64.89	67.48	70.08	70.59
Mar	57.56	59.86	62.16	62.01
Apr	52.68	54.79	56.89	57.25
May	54.40	56.58	58.06	58.82
Jun	58.01	60.33	60.65	62.67
Jul	65.36	67.97	68.59	70.47
Aug	59.77	62.16	64.56	64.67
Sep	64.54	67.12	67.70	70.23
Oct	68.13	70.86	73.58	74.20
Nov	78.02	81.14	81.26	84.04
Dec	78.99	82.15	82.31	85.06

Table 1: Buying side average tariffs in F1 (e/MWh)

Month	Coalition	Market
Jan	59.19	38.36
Feb	55.42	35.84
Mar	47.79	30.99
Apr	44.67	29.23
May	47.77	31.29
Jun	50.84	33.11
Jul	57.95	37.75
Aug	53.04	34.32
Sep	57.15	36.81
Oct	53.20	34.48
Nov	65.76	42.30
Dec	62.16	40.95

Table 2: Selling-side average tariff in F1 (e/MWh)

Even if these values have been obtained with an in-sample solution, the economic benefit for both the buying and selling side is clear, thus confirming the convenience for each prosumer to stay within the aggregation. We outline also that on the buying side the cost components for the considered prosumers' groups are quite different. We have imposed different variability range for the 3 groups, in particular preserving residential users by a high tariff volatility.

5.3.2 Efficient Frontier

We have also analysed the impact of the risk-aversion attitude of the aggregator on the energy management plan for the entire coalition. Figure 3 reports the efficient frontier, that is the set of optimal planning obtained for different values of the risk-aversion parameter.

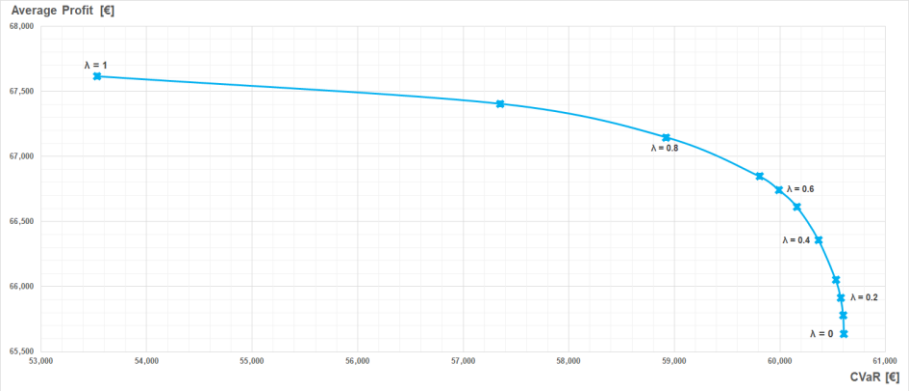


Figure 3: Efficient Frontier

The lower the value of λ the more profitable, but also the more risk (in terms of worst-case profit), the planning. This result shows how the proposed model can be a useful tool to implement different policies or to find the best trade-off between risk and economic convenience.
The risk-aversion attitude has also an effect on the optimal procurement mix, as confirmed by Table 3.

λ	Bilateral contracts	DAEM	Self-production	Total
0.01	2217.5	2058.9	262	4538.40
0.5	2542	1712	284.40	4538.40
0.99	2754	1458.40	326	4538.40

Table 3: Energy procurement mix (MWh) for different values of λ

Again, a more conservative attitude of the aggregator, that is high levels of λ, reduces the energy amount to buy from the market, preferring to procure energy from sources with a price known in advance.

5.3.3 Value of Stochastic Solution

Another set of computational experiments has been carried out in order to evaluate the benefit obtained by explicitly modelling uncertainty in the decision process. We have compared the stochastic solution provided by the proposed model with its deterministic counterpart, by means of the value of stochastic solution (VSS) (see [27]). Since we have adopted a mean-risk objective function, the VSS can be formulated as the weighted sum of the difference of the values of the two objectives:

$$VSS = \lambda(\bar{H}^{Sto} - \bar{H}^{Det}) + (1-\lambda)(CVaR^{Sto} - CVaR^{Det}) \qquad (38)$$

Here, $CVaR^{Sto}$ and $\bar{H}^{Sto}$ are the values obtained by solving the stochastic programming model, whereas $CVaR^{Det}$ and $\bar{H}^{Det}$ are determined by solving the same stochastic problem with the first-stage variables fixed to the values of the optimal solution of the deterministic (with the expected values) problem (also known as EEV).
The following Table 4 reports the relative VSS expressed in percentage (with the respected to EEV) for different values of risk aversion parameter λ.

λ	**VSS %**	λ	**VSS %**
0.01	33.9	0.6	26.25
0.1	32.64	0.7	25.27
0.2	30.77	0.8	24.66
0.3	28.92	0.9	22.75
0.4	28.25	0.99	21.70
0.5	27.34		

Table 4: Value of stochastic solution

The results show that the VSS values are higher when a risk neutral position is considered (low values of λ). In this case, accounting for uncertainty can lead to more effective solutions.

5.3.4 Solution Stability

In order to evaluate the stability of the solutions different experiments have been carried out by varying the size of the scenarios set. Preliminary tests have shown that the adopted scenario generation technique exhibits good in-sample stability

properties. These results are summarized in Figure 4 where we report the expected terminal profit and CVaR values obtained by solving different problem instances by varying the size of the scenario set.

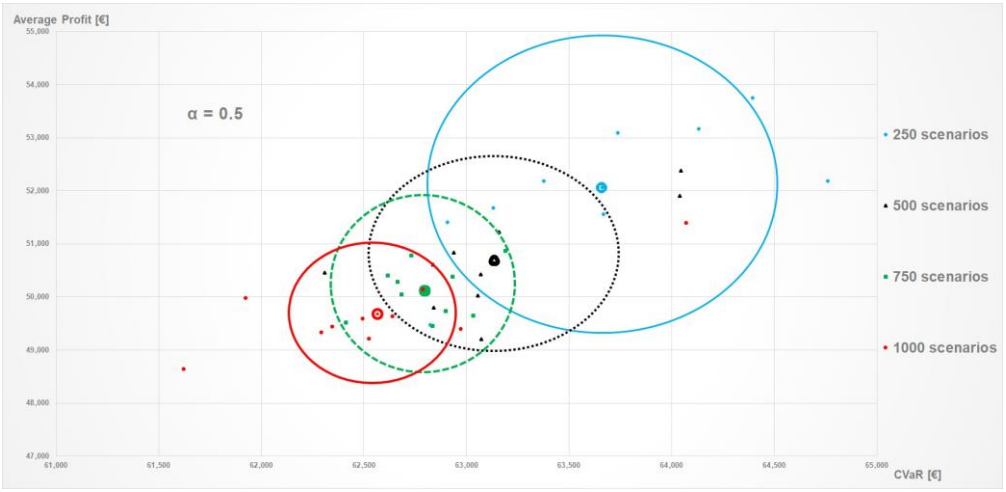


Figure 4: Solution stability

More specifically, for each scenario set ($S = 250, 500, 750, 1000$), several stochastic programming problems have been iteratively generated and solved from randomly generated scenario samples to test the solution stability with respect to each set of problems. As evident from Figure 4, the radius of the circle measured by the standard deviation is sufficiently small so to guarantee robust solutions for sets also with 500 scenarios. The advantage of adopting larger scenario sets is not so relevant, considering also the computational drawback due to the increment of the problem size.

5.3.5 Out-of-Sample analysis

A last set of computational experiments have been devoted to assess the effectiveness of the approach with an out-of-sample analysis. We have evaluated the economic performance of the overall aggregation energy planning by considering the really observed values of the uncertain parameters each TOU block of 12 months starting from January 2019. Moreover, we have compared the solution provided by the proposed approach with other possible decision policies. The first one is the deterministic counterpart, that is with the solution of the model with the value of each uncertain parameters sets to the corresponding expected value calculated on the historical series. The second benchmark we have considered is a sort of "decoupled" approach, in which the aggregator solves a first

procurement planning problem and then, on the basis of the solution obtained, defines the tariff structure which can fit with the procurement plan and guarantee the economic convenience for prosumers. Figure 5 depicts the profit/loss evolution of the solutions obtained with the proposed "integrated" model and the two considered benchmarks for a test case with $\lambda = 0.5$ and $\beta = 0.95$.

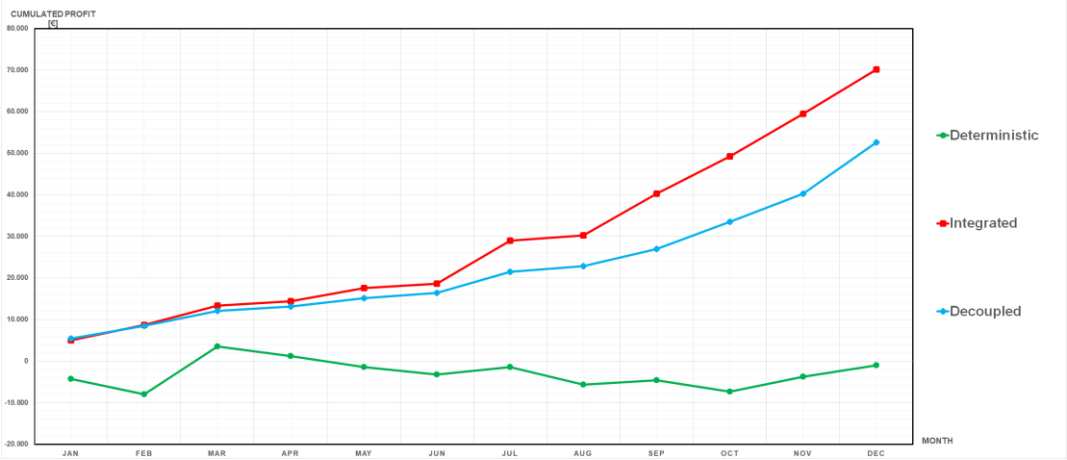


Figure 5: Out-of-sample solutions comparison

As we can see, the planning obtained with our model provides more benefits, making the overall coalition much more globally sustainable, in particular w.r.t. the deterministic counterpart where no uncertainty management is implemented. The benefit of about 33% of the integrated approach w.r.t. the decoupled one is due to the more flexibility guaranteed by the possibility to define in an integrated fashion decisions that are closely related.

In order to assess in an out-of-sample fashion also the economic convenience for the prosumers, in the following we report the actual tariff components, obtained after the observation of the real values of market prices. Table 5 contains the buying and selling side final tariffs for residential prosumers in F1 and the corresponding market alternative prices with a 35% mark-up. As we can see, the advantage for the prosumers is evident for both the buying and selling side, so to ensure the attractiveness of the coalition.

	Buy		**Sell**	
Month	**Coalition**	**Market**	**Coalition**	**Market**
Jan	73.38	79.33	58.60	37.98
Feb	63.59	69.18	60.96	41.22
Mar	64.47	71.31	43.97	34.71
Apr	46.88	50.95	46.01	30.69

May	52.22	56.47	48.73	32.23
Jun	63.81	68.94	54.91	35.76
Jul	60.13	64.83	56.21	36.62
Aug	63.95	69.20	55.16	35.69
Sep	56.79	61.80	45.72	29.45
Oct	71.54	77.91	50.54	32.76
Nov	93.62	100.85	59.18	38.07
Dec	81.36	87.61	68.89	43.4

Table 5: Buying and Selling-side real tariff for residential prosumers in F1 (e/MWh)

Similar results have been obtained for the other prosumers group (on the buying side) and for all the TOU blocks.

5.4 Conclusion

In this chapter we have addressed the complex decision problems related to the energy procurement and the tariff definition for a coalition of prosumers in an integrated fashion. The different nature of the decisions to be made and the inherent uncertain nature of many parameters make the problem very complex to deal with. To the best of our knowledge, the problems have always be faced separately, also because prosumers aggregations are raising just in the last few years. Our contribution aims at dealing with decisions that can appear to be independent at a first sight, but are strictly related for an effective planning and management of a coalition. We have proposed a mathematical model based on the multiperiod 2-stage stochastic programming framework, with a mean-risk objective function with the CVaR as risk measure. As regards the tariff definition we have designed price schemes with a fixed component and a variable one, in order to give the prosumers an active role for their economic savings. Moreover, we have included the possibility to have different tariffs on the basis of the nature of the prosumers, thus allowing the aggregator to implement various policies. The computational experience, carried out on a coalition built starting from real-life data, has shown that the proposed approach can be a useful and articulated decision support tool for the economic sustainability of a prosumers aggregation.
The comparison with two decision benchmark in an out-of-sample fashion has further demonstrated the effectiveness of our integrated approach to the problem. As future research we aim at including more complex tariff schemes, oriented towards individual plans, in order to enhance the responsiveness of single prosumers. Another possible enhancement is related to the inclusion in the medium-long term energy procurement of “seasonal” storage systems, which can implemented also by means of clusters of commercial batteries, in order to

improve the flexibility in the management. Finally, our goal is to integrate the model with support tools for other decision problems, like for example the selection and scheduling of storage systems and the day-by-day energy management for the coalition as a whole.

Appendix A. Model Notation

The following Tables A.6 and A.7 report all the parameters and variables introduced in the mathematical model.

Table A.6: Model Notation part 1

Sets	
T	Time horizon articulated in elementary time periods (e.g. months)
F	Set of time-of-use blocks, in which hours are classified
H	Set of prosumers groups (e.g. residential, commercial, public)
N	Set of bilateral contracts available for the coalition
S	Scenario set
Parameters	
N^{max}	Maximum number of bilateral contracts tah can be accepted
$LB_{itf,}UB_{itf}$ $[kWh]$	Lower and upper bound for energy bought from bilateral contract i (if selected) in TOU block f of time period t
FB_i[€]	Fixed cost for the acceptance of bilateral contract i
PB_{itf}[€/kWh]	Unit price for energy from bilateral contract i in TOU block f of time period t
G[€/kWh]	Unit production cost from traditional production systems
Q_f^{max}[kWh]	Upper bound on the energy from traditional production systems in TOU block f of each time period
τ_h	Upper bound on variable α_h
φ	Upper bound on variable μ
Θ^B	Mark-up percentage on the purchasing unit price out of the coalition
Θ^S	Reduction percentage on the selling unit price out of the coalition
$\gamma_h^{LB}, \gamma_h^{UB}$	Percentage lower and upper bound on the overall unit purchasing tariff for group h w.r.t. the average value of the unit purchasing price on the DAEM
η^{LB}, η^{UB}	Percentage lower and upper bound on the overall unit selling tariff w.r.t. the average value of the unit selling price on the DAEM
λ	Risk aversion parameter
β	Confidence level for CVaR
π_s	Probability of occurrence of scenario s
D_{tfh}^s[kWh]	Overall demand of prosumers of group h in TOU block f of time period t under scenario s
P_{tf}^s[€/kWh]	Unit purchasing price from the DAEM under scenario s in TOU block f of time period t

R_{tf}^{s}[kWh]	Overall production from renewable systems of the coalition in TOU block f of time period t under scenario s
B_{tf}^{s}[€/kWh]	Unit selling price on the DAEM under scenario s in TOU block f of time period t
Z_{tf}^{s+}[€/kWh]	Unit purchasing price on the balance market
Z_{tf}^{s-}[€/kWh]	Unit selling price on the balance market

Table A.7: Model Notation part 2

First-stage Decision Variables	
k_i	Acceptance of bilateral contract i (binary)
x_{itf}[kWh]	Energy to buy by bilateral contract i in TOU block f of time period t
Q_{tf} [kWh]	Energy to produce with traditional production systems in TOU block f of time period t
y_{tf} [kWh]	Energy to buy from the DAEM in TOU block f of time period t
w_{tf} [kWh]	Energy to sell on the DAEM in TOU block f of time period t
$TC1_{tfh}$ [€/kWh]	Unit base purchasing tariff for prosumers in group h for TOU block f of time period t
α_h	percentage of deviation of the real market purchasing price from the historical average value to adopt for the variable purchasing tariff for prosumers in group h
$TP1_{tf}$[€/kWh]	Unit base selling tariff for all the prosumers for TOU block f of time period t
μ	percentage of deviation of the real market selling price from the historical average value to adopt for the variable selling tariff for all the prosumers
Second-stage Decision Variables	
Δ_{tf}^{s+}[kWh]	Energy to buy on the Balance Market in TOU block f of time period t under scenario s
Δ_{tf}^{s-} [kWh]	Energy to sell on the Balance Market in TOU block f of time period t under scenario s
$TC2_{tfh}^{s}$ [€/kWh]	Unit variable purchasing tariff for prosumers in group h for TOU block f of time period t under scenario s
$TP2_{tfh}^{s}$ [€/kWh]	Unit variable selling tariff for all the prosumers for TOU block f of time period t under scenario s
Auxiliary Variables	
σ^{s} [€]	Auxiliary variables for CVaR linearization
Derived Quantities	
Y^{s} [€]	Overall profit for the coalition under scenario s
V_{TC}^{s}[€]	Revenue from energy sold by the aggregator to the prosumers under scenario s

V_{MKT}^{s}[€]	Revenue from energy sold on the DAEM under scenario s
C_{TP}^{s}[€]	Cost for the energy bought by the aggregator from the prosumers under scenario s
C_{BC}[€]	Cost for energy procurement from bilateral contracts
C_{Prod}[€]	Cost for energy production from traditional systems
C_{MKT}^{s}[€]	Cost for energy bought on the DAEM
C_{Err}^{s}[€]	Cost (revenue) for energy bought and sold on the balance market
VaR_{β}[€]	Value at Risk at a confidence level β for profit distribution
$CVaR_{\beta}$[€]	Conditional Value at Risk at a confidence level β for profit distribution

Appendix B. Coalition data

The following Table B.8 reports the expected values of the overall demand for each group, while Table B.9 contains the expected values of renewable production.

Table B.8: Expected value of energy demand for each prosumer group (MWh)

		TOU block		
Prosumer group	Month	F1	F2	F3
Residential	Jan	26.28	28.96	28.83
	Feb	28.34	30.14	22.96
	Mar	29.94	25.87	19.03
	Apr	16.31	17.47	20.52
	May	12.43	22.68	24.02
	Jun	16.77	22.81	26.05
	Jul	30.49	36.32	25.12
	Aug	14.70	15.11	20.82
	Sep	12.03	19.33	22.06
	Oct	16.37	26.28	22.54
	Nov	14.48	25.70	21.88
	Dec	20.46	25.14	27.44
Commercial	Jan	78.84	86.87	86.48
	Feb	85.03	90.41	68.89
	Mar	89.81	77.61	57.08
	Apr	48.92	52.41	61.55
	May	37.30	68.03	72.05
	Jun	50.31	68.44	78.15
	Jul	91.47	78.97	75.35
	Aug	44.11	45.32	62.45
	Sep	36.08	58.00	66.18
	Oct	49.11	78.84	67.63
	Nov	43.44	77.11	65.63

	Dec	61.38	75.43	82.32
Public utility	Jan	52.56	57.91	57.65
	Feb	56.68	60.27	45.92
	Mar	59.88	51.74	38.05
	Apr	32.61	34.94	41.03
	May	24.87	45.35	48.04
	Jun	33.54	45.62	52.10
	Jul	60.98	52.65	50.24
	Aug	29.40	30.21	41.63
	Sep	24.05	38.66	44.12
	Oct	32.74	52.56	45.08
	Nov	28.96	51.41	43.75
	Dec	40.92	50.28	54.88

Table B.9: Expected value of energy production from renewable systems (MWh)

	TOU block				TOU block		
Month	F1	F2	F3	Month	F1	F2	F3
Jan	78.13	16.36	21.22	Jul	171.99	40.44	29.11
Feb	77.22	22.21	21.55	Aug	184.24	37.77	24.12
Mar	139.02	16.34	14.53	Sep	77.40	23.84	22.13
Apr	167.28	13.44	19.22	Oct	87.47	20.10	11.88
May	121.74	45.93	17.27	Nov	64.43	5.51	5.09
Jun	116.65	37.52	40.64	Dec	68.55	11.33	5.38

Appendix C. Bilateral contracts

Since the price components for each contract can differ for each month, in Table C.10 we report just the average unit price for each time-of-use block and the fixed cost. A more detailed description of the considered bilateral contracts is reported in Appendix B of [12].

Table C.10: Bilateral contracts price components

		BC1	BC2	BC3	BC4	BC5	BC6	BC7	BC8	BC9	BC10
Aver. price [e/MWh]	F1	42.3	42.3	42.3	41.6	41.4	42.0	40.9	40.8	40.9	40.8
	F2	43.3	43.3	43.3	43.7	42.4	41.3	41.3	40.9	40.9	41.4
	F3	35.8	35.7	35.6	35.6	35.1	34.6	34.5	34.5	34.4	34.7
Fixed cost [e]		650	675	575	375	725	650	625	500	550	850

CHAPTER 6

A Constrained Portfolio Optimization Framework with a Tri-Objective Design using Multi-Objective Evolutionary Algorithms and Decision Parameters

6.1 Introduction

Modern portfolio theory is inspired by the Markowitz [1] forerunner work that introduced the classical mean variance framework. Nevertheless, this framework is widely applied in portfolio application, still computing precise values of expected returns and covariance of assets is a complicated task considering dynamic markets trends. Amidst this dynamic financial domain, considering the continually altering stock markets, evaluating the values of covariance parameters appears to be complicated, and this imposes enormous complexities in the model. The task of assessing the magnitudes of expected returns using a model based on several predicted values is a challenging task since these values are calculated using forecasting methods. Due to a higher degree of inclination of the solution towards the dynamic state of the evaluation scheme, which is used for calculating the values of the stock returns, the values obtained for an optimization solution appear to be ambiguous. Therefore, it is customary to treat parameters affecting expected returns with a degree of inexactness [3].

A portfolio is described in terms of expected returns and a risk value, where the values of the expected returns are selected based on a specific amount of the risk-term [1, 2, 3]. The essence of portfolio optimization is formulating a trade-off between computed magnitudes of returns and risk-term. Numerous internal and exterior elements influence these two parameters; thus, extracting an appropriate trade-off amidst these two magnitudes is an extensive direction for performing research activities and investigations. Typically, portfolio optimization models are based on complicated functions representing a set of decision variables. While formulating a portfolio optimization model comprised of these functions along with realistic constraints, e.g., floor and ceiling constraint, cardinality constraint, pre-assignment constraint, etc., the design can lead to a complicated assorted problem. The classical methods are inapt to generate optimal solutions considering these problems. However, multi-objective evolutionary algorithms (MOEAs) are gaining acceptance for finding solutions to a portfolio optimization problem based on mean-variance theory with realistic constraints [13, 14, 20]. The classical

techniques used for solving portfolio optimization problem have significant drawbacks. These issues must be considered in the design of a better model to obtain a realistic and optimal solution of the problem. Finding improved models to obtain optimal solutions for capital allocation in a situation based on diverse portfolios is a complicated issue, and it is exceedingly required due to numerous factors. The primary factor is the inclusion of risk in the situations of possible losses in different and interconnected portfolios [2]. The investors' preferences regarding the choice of cost and possible risk related to numerous losses are not defined precisely in the situations of continuously varying values of the return. Finding optimal solutions of the problem comprising of a variety of inexactness and risk values is a complicated task. The issue related to a decision parameter for deciding a trade-off between objectives of expected return and variance is discussed here. The investor might be interested in setting up a trade-off between two or more objectives. Generally, one of the objective is expected return. The other objectives could be reliability [2] or the magnitude of variance. A decision parameter a_i is used for a trade-off between expected return and reliability is described in [2]. A new decision parameter is used in this study for this purpose. An investor can set a trade-off between expected return and variance. An algorithm is also presented in this study for obtaining deviations in the value of new decision parameter based on a drift value. This algorithm also uses a pre-selected maximum value of new decision parameter. The issue pertaining to allocation of capital in a portfolio is discussed next. The expected return in a portfolio optimization problem is affected by the amount of allocated capital to a set of sub-portfolios. The ever-changing market trends greatly influence the expected returns. The inappropriate allocation may lead to a sub-optimal solution [12]. Further, this solution may be undesirable because of the investors' preferences about the parameters of the model. Therefore, there is a need to bring out a convenient model of a portfolio optimization problem, which incorporates these features in the solution. The proposed model has a description of a tri-objective framework, which is based on three decision parameters. These decision parameters are related to the proper allocation of the capital of a sub-portfolio category and a desirable level of downside risk.

The second decision parameter q_i is used to represent an appropriate level of the proportion of capital allocated to a category of sub-portfolios [2]. This decision parameter is significant for generating optimal solutions for the portfolio optimization problem.

The issue related to risk measures and downside risk is discussed next. There are plenty of ways to include risk measures in the portfolio optimization model. However, traditional Markowitz's postulate is using variance as a risk measure. This concept is based on an assumption of normalization of the returns [8, 9]. This assumption does not hold in a real market situation [7]. Therefore, research activities focus on novel techniques for computing risk, and one such prominent method is based on including downside risk.

The downside risk can focus on adverse situations occurring in the portfolio optimization problem. While semi- variance lays focus on negative deviations from its mean value [8], the downside risk concentrates on finding a maximum probability that the return is more significant than a desirable value. The downside risk faces the errors that are occurring due to incorrect estimation based on using only a portion of the information. If the distribution of the returns is asymmetric, then the downside risk is more reliable than variance [8]. Currently, the possible downside risk is represented in the form of Value at Risk and Conditional Value at Risk. In the proposed portfolio optimization model, the third decision parameter is d_i, which signifies a level of downside risk for a sub-portfolio, and its value is calculated using CVaR.

The issue related to formulation of a composite index and the proposed tri-objective model is dealt next. In literature [75], it is pointed out that a formulation of the desired objective for the solution of the portfolio optimization problem is rarely based on one measure, preferably a group of significant variables is combined to provide a more realistic and precise definition of the objective. The techniques that are commonly used for formulating a composite index are discussed here. Few prominent strategies, which are used for formulating an index in optimization problems, are discussed here. A monetary index is framed using a weighted sum of parameters, e.g., real interest rate, difference of exchange rates, etc. [77]. This index uses logarithmic calculations for flattening skewness, and it is responsible for an increase in correlation [77]. The use of a single index based on mean values of present outputs and related output level for a variable is described in [76, 77]. Therefore, the proposed model has the third objective represented by a single index. The proposed model presented in this script introduces a tri-objective framework. Three significant decision parameters are used for the formulation of a composite index. An upper bound value of the composite index is also introduced in the model based on a desirable highest level of probability.

The proposed tri-objective model is suitable to generate efficient and optimal solutions of the problem in the context of appropriate allocation of capital to sub-portfolios, and it considers the investors preferences regarding downside risk and a trade-off between expected return and variance. The first two objectives in the model are expected return and risk, which are commonly used in a modern portfolio optimization problem. The third objective in the model is introduced in the form of a single composite index, which is based on three significant decision parameters viz. a_i, q_i and d_i, [2]. The decision parameters in the proposed model are defined using modified algorithms in this study. These decision parameters represent the desirable values for the trade-off between expected return and variance, proportion of capital allocated to a category of sub-portfolios, and a downside risk level that is used in the optimization problem. The significance of using a composite index is to generate a means of grouping all essential parameters and indicators into an index that is capable of representing the objective and its

status, which can be formulated from a set of numerical targets [77].

The third objective is based on a single index value comprised of an aggregation of these three decision parameters. The aggregation value of these decision parameters generates a single index value, which can be used by an investor for choosing an appropriate bound for the third objective more easily. The availability of a unique index is beneficial to an investor for selecting an appropriate level for each of these essential decision parameters. This objective is introduced in the strategy to represent an aggregation of decision parameters in the form of a single index. An upper bound of the composite index is formulated, and the upper bound considers the variations in three decision parameters. Roy *et al.* [6] describe a process to minimize the value of an upper bound for a chance of a disastrous situation. The upper bound is represented as a highest level of the expected return for a portfolio optimization problem [5]. The probability level is related to the mean value of the return and its variance [5, 7]. The upper bound value used in the proposed tri-objective model can help an investor choose the desired value for it. The proposed model's primary goal is to bring out a novel design of a tri-objective portfolio optimization model in the form of a composite index based on decision parameters. Moreover, an upper bound is formulated for the composite index that allows the investor to select a desirable aggregation value of decision parameters. This objective is significant for representing a trade-off between objectives of expected return and variance. We then consider the importance of downside risk level and a fraction of capital amount invested in a category of sub-portfolios. These are included in the third objective, and these parameters are represented as decision parameters. The third objective serves as a means to study the impact of these parameters on the portfolios.

The proposed model is tested on existing MOEAs viz. NSGA-II, SPEA2, MOEA/D, and MOPSO, and their performances are compared. The third objective is used to study the impact of decision parameters on the portfolios, validate their inclusion in the objective, and analyze the optimal solution based on these parameters to compare the outputs.

The chapter is divided into six main sections. Section 6.2 provides a review of the existing concepts in portfolio optimization and recent algorithms that are used in its solution. The description of the tri-objective portfolio optimization, along with a description of the modified algorithms used in the proposed model and based on decision parameters, is provided in Section 6.3**.** Section 6.4 contains a description of MOEAs along with a discussion on constraints used in a portfolio optimization problem. Section 6.5 is comprised of numerical illustration, outputs, and a comparative analysis of the algorithms. Finally, the conclusion with a summarization note and scope for future work is provided in Section 6.6.

6.2 Literature Review

The next section describes the role of risk representations in portfolio optimization models and the significance of formulating a composite index in a model. This section also has a review of robust optimization models of portfolio optimization, different types of uncertainties used in a portfolio optimization model, using different kinds of risk measures in portfolio optimization, and handling the practical constraints in a portfolio optimization model. Next, a review of multi-objective methods that are used for the portfolio optimization model is given.

A method for solving a portfolio optimization problem based on investors' preferences is described with a selection scheme for choosing the varied magnitudes of risk level [7]. This selection of coveted risk levels in the formulation has a significant role. In the literature, there are plenty of techniques to model risk in the solution. Konno *et al.* [86] [88] employed a mean value of deviation for choosing risk levels. Konno *et al.* [87] used variance and the value of skewness for selecting the appropriate risk level. The solutions are obtained using functions based on the mean value of deviation and degree of variation. Since variance is not a good indicator of risk level [4, 11] due to its inappropriate effect on the expected return around the average value, there is a need for exploring modern ways of finding the magnitude of the risk level. One such measure is VaR that also has impediments in terms of subadditivity, convexity and anomaly [8]. It neglects loss above VaR, and the computation of the risk level is arduous. Another significant index employed is CVaR, which represents the mean values of the loss for worst-case scenarios [21].

Composite indexes are vital tools for statistical analysis related to access and finding ranks of countries as well as universities [73]. The significance of using a composite index is to generate a means of grouping all essential variables and indicators into an index that is capable of representing the objective and its status, which can be formulated from a set of numerical targets [75]. Recently, a scheme of portfolio optimization theory for choosing weights and variables of the composite index is discussed in [75].

A scheme based on entropy measures for representing inexactness in risk premiums and the covariance matrix is given in [83]. A model for the mean-variance framework is designed, and outputs are obtained under the worst- case situation. The comparative results indicate that parametric inexactness has a vital role in the portfolio decision-making process, and this uncertainty is inversely proportional to outputs of the portfolio optimization problem. Recently, Garlappi *et al.* [84], and Asadujjaman *et al.* [85] proposed some prominent parameter estimation methods to deal with the inexactness of parameters in the portfolio optimization problem.

Robust optimization models have a vital role in the portfolio optimization (PO). It searches the optimum decision inside the uncertainty set from worst-case framework perspective. Goldfarb *et al.* [22] proposed the statistical techniques for

building the uncertainty sets corresponding to the component frameworks of asset returns. They reported and suggested that their proposed robust portfolio optimization problem can have alternative representation in the form of a second-order cone program through demonstration. Koening *et al.* [23] assumed the expected return of the assets to be stochastic and proposed the more robust representation for mean variance model. Moreover, they reported the importance and the impact of the parameters defining the variance-covariance, mean matrix with distinct kinds of uncertainty such as box, ellipsoidal, polytopic, and constructed the second- order cone programs. Fabozzi *et al.* [24], [25] studied recent contributions to robust portfolio optimization with downside risks (VaR, CVaR). Recently, Gülpınar *et al.* [26] proposed a robust portfolio framework with temperature uncertainty. Fernandes *et al.* [27] introduced a robust portfolio optimization framework with a data- driven polyhedral-based loss constraint for the optimum portfolio. The impact of risk is the most challenging task in the portfolio optimization problem.

The different versions of uncertainty presented in the literature, which is considered by researchers. Li *et al.* [28] provided a reasonably robust policy with moment uncertainty for a portfolio optimization problem to seek optimal solutions. Cont *et al.* [29] introduced a quantitative framework for calculating the uncertainty model in option price models. Consigli *et al.* [30] introduced financial decision making under uncertainty with multi-stage robust portfolio optimization. Qin *et al.* [31] proposed a model for POP called mean semi-absolute deviation model. They have converted the proposed model into deterministic representation through numerous uncertainty distribution. A variety of extended mean-variance model is presented in the literature, so in this direction, Ning *et al.* [32] proposed a variation of the extended mean variance (VaR) model called Mean-VaR model. In the proposed model, the uncertainty variables are manifestation of return of the security.

The returns of risky assets are recognized as arbitrary elements in the probabilistic portfolio model. According to the fundamental inference of probabilistic portfolio model, a sufficient amount of past data may be instrumental in reflecting the future state of the risky assets that can be reflected by past data. Several non-probabilistic factors affect portfolio decision-making, such that psychological, economic, political factors, etc. The influence of these factors leads to a fuzzy representation of the future state of risky assets. Liu *et al.* [34] proposed the two possibilistic mean-semivariance models instead of variance for the risk measurement with realistic constraints in a fuzzy environment, and reported the drawbacks of use variance for risk measurement. In the same direction, Zhang *et al.* [35] presented a new multi-period mean absolute deviation fuzzy portfolio optimization model. Ferreira *et al.* [36] introduced a fuzzy hybrid integrated model for the portfolio optimization problem. Gupta *et al.* [37] reported an IFPS (intuitionistic fuzzy portfolio selection) model exploiting entropy, variance, skewness, and mean as objectives in with realistic constraints.

The convex programme that is the essence of the Markowitz portfolio selection becomes a complex nonlinear problem with the inclusion of realistic constraints, such as cardinality constraint, and pre-assignment constraints. In the presence of these practical constraints, classical optimization algorithms are unable to handle this optimization problem [38], resulting in the employment of Evolutionary algorithms (EA) to solve portfolio optimization challenges with practical constraints. Several authors have proposed the variations of EAs to address this problem. Heuristic approaches focused on simulated annealing, Tabu search, and the genetic algorithm was suggested by Chang *et al.* [39]. M., Lucas, C. *et al.* [40] achieved an upgradation over these heuristics by incorporating a subset optimization step. A PSO based heuristic approach was proposed by Cura *et al.* [41] for cardinality constrained portfolio optimization. Ni *et al.* [40] observed an improvement in the performance of portfolio optimization by utilizing the concept of population topology used in PSO.

A tri-objective portfolio optimization framework is presented in [13, 14, 15, 16, 17] with different objective apart from return and risk. The consideration of multiple conflicting objectives in portfolio optimization seems to make it a multi-objective problem. Therefore, multi-objective EA variants are more favoured since it enables to use of a single run of the algorithm for estimating the entire Pareto front. Anagnostopoulos *et al.* [43] showed an improvement of their multi-objective (MO)EAs over single objective EA. They assessed its performance using CPU time as the performance evaluation criteria for a constrained portfolio selection problem. In recent years, several implementations of MOEAs have been widely used to solve the portfolio optimization problem with cardinality constraint.

Further, a differential evolution based MOEA was suggested by Krink *et al.* [44]. A combined approach based on SPEA2 and NSGA2 was utilized by Macedo *et al.* [45] to solve the mean semi-variance portfolio optimization problem. Lwin *et al.* [46] suggested a learning heuristic-based multi-objective EA for portfolio optimization. This algorithm recognizes the most promising solutions in the Pareto front. Soares *et al.* [47] modified the decomposition-based multi-objective algorithm (MOEA/D) using interval analysis to solve the portfolio optimization problem to get more diversity in the solutions. The selection of an encoding strategy has a significant contribution for improving the performance of MOEA, specifically for the portfolio optimization problem. In this direction, recently, Liagkouras *et al.* [48] suggested an efficient encoding strategy for handling the difficulties of cardinality constrained portfolio selection problem. Lwin *et al.* [49] proposed the mean-Value at Risk portfolio selection model with practical constraints. They have taken the value at risk instead of variance as representation of risk measure to get a better market risk exposure. Chen *et al.* [50] proposed a heuristic method called e-NSLS that is extended version of multi-objective evolutionary paradigm, it is focussed on Non-dominated Sorting and Local Search (NSLS) to solve the problem of portfolio optimization restricted by cardinality constraint. Silva *et al.* [51] presented a modified variant of multi-objective particle

swarm optimization (MOPSO), which is based on a ranking procedure named as adaptive ranking MOPSO (ARMOPSO). Fernandez *et al.* [52] modeled the investor's preferences based on the interval principle and the outranking approach to search the optimal portfolio from the point of view investor, where return and risk are described by probabilistic confidence intervals. This model is resolved by the decomposition-based multi-objective evolutionary algorithm (MOEA/D). Zhou *et al.* [54] provided a survey of widely accepted MO algorithms, which are used to several real MO problems. Doering *et al.* [55] analysed the recent advancement in the employment of metaheuristics methods within portfolio optimization problems and risk management issues. Recently, Kalayci *et al.* [56] reported the advancement in use of deterministic models for the mean-variance portfolio optimization problem (MVPO).

6.2.1 Motivation and contributions of the proposed strategy

The motivation and contributions of the proposed strategy are outlined below:

Modelling. The commonly used models of portfolio optimization issue lack a suitable representation of the problem considering the latest market trends. These models need to find relevant factors in the design that can affect the investors' preferences. One such factor could be to decide among multiples objectives comprising of return, reliability, or variance. Therefore, a choice has to be made regarding a trade-off between these objectives. Investors are also concerned about the optimal allocation of capital among categories of sub-portfolios [2]. The proportion of the capital invested in a category influence the expected return under a dynamic market situation. Therefore, the investor needs to choose a more straightforward way to tackle this issue. Generally, the allocation of capital is based on commitments [12] in a complicated market situation. The choice of allocation in a portfolio optimization may yield a sub-optimal solution due to a lack of a convenient utility. If the allocation of the capital is inappropriate, then the optimal solution obtained is not desirable. Besides, with an appropriate allocation, the solution could be undesirable if the investors' preference regarding the trade-off between return and variance is not considered.
The formulation in this script puts forth a tri-objective model to get efficient portfolios considering the allocation to sub-portfolios and having portfolios closer to the investor's preferences. The tri-objective model is based on the selection of a set of decision parameters. Besides, this is beneficial to allow a limit for downside risk [2] for a sub-portfolio. The advantage of this model is that it quantifies the impact of the decision parameters as a single index in a portfolio optimization problem. The investor can make an appropriate choice regarding the desirable value for the index. This model aims to provide investors with an alternative for quantifying the performance for optimal solutions and obtain useful outputs.

Composite indexes are vital tools for statistical analysis related to access and finding ranks of countries and universities [73]. These composite indexes can be utilized for finding the performance metrics index for sustainability and formulating indexes for perplex situations, which cannot be measured directly. This study presents a formulation of a composite index based on an aggregation of the values of three decision parameters. This composite index is used as the third objective in the proposed portfolio optimization model. Moreover, realistic constraints such as cardinality, equality, and quantity constraints are also included in the scheme.

Algorithmic. Traditional optimization methods are incapable of maintaining the portfolio selection model in the existence of realistic constraints like cardinality, pre-assignment [20]. Consequently, the present study aims to seek the use of prevailing MOEAs (multi-objective algorithms) for the introduced model. The representation of a composite index as the third objective, along with its upper bound in the model imposes multi-folded restrictions in the model. Currently, these restrictions are not represented appropriately in a model based on conventional multi-objective algorithms. The upper bound can be included in the framework with fairly less difficulty. A scheme to include the upper bound in the model is described in the current perusal. To check the usefulness of the introduced scheme and to find an optimal solution to the portfolio optimization problem, this framework is tested with four prevailing multi-objective evolutionary algorithms, viz. NSGA-II [57], SPEA2 [58], MOPSO [68] and MOEA/D [59]. MOEAs are adapted so that the solution of the portfolio optimization problem is computed based on the fulfilment of all the restrictions in the introduced model. The crossover function that is used in the MOPSO algorithm is modified in the proposed strategy.

Experimental Study. This study evaluates the adapted MOEAs for the tri-objective portfolio optimization model using a dataset for five markets (HS33, DAX100, FTSE100, S&P100, and Nikkei225). The efficient frontiers, along with the outputs based on the hypervolume metric, are used to compare the performances of adapted algorithms. Further, the values of the third objective and the decision parameters are also used for this comparison. Based on hyper-volume statistics, NSGA2 and SPEA2 have better performance as compared to other adapted algorithms. Based on statistics of the decision parameter (a_i), NSGA2 has better performance over other adapted algorithms for market 1, market 2, and market 4. At the same time SPEA2 exceeds in performance over other adapted algorithms for market 3 and market 5. In the case of the decision parameter (q_i), MOPSO exceeds in performance over other adapted algorithms for all markets data set.
Similarly, in the case of the decision parameter (d_i), MOPSO has better performance over other adapted algorithms for market1, market 3, and market 5. At the same time, SPEA2 outperforms other algorithms for market 2, and NSGA2

outperforms other algorithms for market 4. Experimental results indicate that choosing a minimal value of the third objective is preferable for obtaining a higher expected return for a specified range of risk. Moreover, the third objective's value is highly correlated to the value of the decision parameter (a_i). A practical way of calculating the upper bound values of the third objective in the proposed tri-objective model is also illustrated.

6.3 Problem Definition

The next sub-sections confer the details of the framing of the proposed tri-objective model. The description of the realistic constraints used in the model is given in sub-section 6.3.1. The details of the objective functions are given in sub-section 6.3.2. The proposed third objective is discussed in sub-section 6.3.3.

6.3.1 Constraints

Assuming a domain of N assets, a vector space $x = (x_1, x_2, \dots, x_N)$, which specifies the ratio of the amount invested in the i^{th} asset is used to describe a portfolio. In a situation when an investor is inclined to investing the entire amount amidst N assets, the following condition holds true

$$\sum_{i=1}^{N} x_i = 1 \tag{1}$$

Practically, the number of assets N could be enormous. Distributing amount amidst accessible assets needs high management and transaction cost of every asset [20]. Thus, an investor is inclined towards putting investing amount for subsets of N assets by considering a constraint on the size of assets having non-zero weights in the portfolio. This constraint that binds these non-zero weights assets using an integer value is given below:

$$k_1 \leq \sum_{i=1}^{N} z_i \leq k_2 \tag{2}$$

The z_i is a binary-valued parameter, which is used for either including i^{th} asset or excluding it in the portfolio. Therefore, $\forall_i \in \{1,2, \dots, N\}$, z_i is given by the following criteria:

$$z_i = \begin{cases} 0, & if\ i^{th}\ asset\ is\ not\ incorporated\ in\ portfolio \\ 1, & if\ i^{th}\ asset\ is\ not\ incorporated\ in\ portfolio \end{cases} \quad (3)$$

These binary-valued z_is' are utilized in another way for modelling constraints on floor and ceiling, which are named as quantity constraints. These constraints restrict asset distributions inside fixed pre-decided bounds. The first one is used for limiting the cost associated with management, whereas the second one is used to avoid portfolio from exceeding focusing on a specific asset. When the cardinality constraints are used equation 2, the quantity constraint is defined as follows

$$z_i lower_i \leq x_i \leq z_i upper_i, \qquad \forall_i \in \{1,2,\dots,N\} \quad (4)$$

Where $lower_i$ and $upper_i$ are the minimum and maximum bounding value of the amount used for i^{th} asset respectively. The significance of including z_i in equation 4 is that when $z_i = 1$, the i^{th} asset is included in the portfolio and amount for i^{th} asset is bounded by $lower_i$ and $upper_i$. As we have considered in this perusal $0 \leq lower_i \leq upper_i \leq 1$ which along with quantity constraints impose long-only constraints ($x_i \geq 0\ \ \forall_i \in \{1,2,\dots,N\}$).

The pre-assignment constraint is another way for including investor's choices in the framework, which is described as the ever inclusion of previously selected assets in the portfolio. Assume I_p indicates a set of indexes for the assets that are restricted to be ever included in the portfolio then

$$z_j = 1, \qquad \forall_j \in I_p \quad (5)$$

Moreover binary-valued parameters imposes the following constraints

$$z_i = \{0,1\} \qquad \forall_j \in \{1,2,\dots,N\}/I_p \quad (6)$$

Quantity and cardinality constraints are interconnected. In the realistic portfolios, the values of $lower_i$ s' and $upper_i$s' is selected in a manner so that the addition of lower binding values of k_2 assets remain less than 1 and the addition of upper binding values of k_1 assets remain more than 1. The following conditions are available

$$\sum_{i \in S} lower_i < 1 \qquad \forall S \in G_{k2} \quad (7)$$

$$\sum_{i \in S} upper_i > 1 \qquad \forall S \in G_{k1} \quad (8)$$

6.3.2 Objective functions

Portfolio optimization theory postulates that objective of risk and objective of return should be simultaneously used for allocating assets in a given period. Whereas, in the case of multi-period portfolio optimal solutions, the factors that depend upon the current market situations, require an enormous amount of formulation for handling uncertainties while allocating assets. Keeping in view these uncertainties, which are inherently present in a portfolio model, there is a necessity to include the risk analysis in the form of decision parameters within a portfolio model, which can accurately represent such uncertainties and the values of decision parameters. These decision parameters and uncertainties can prominently affect the portfolio optimization process; therefore, it is mandatory to include the effects of these parameters and uncertainties using another objective in the formulation of the portfolio optimization model. In this work, we have included the third objective, which minimizes the value of a composite index. Simultaneously, the existing objectives related to risk and return are also present in the proposed model. Further, the formulation of an upper bound value representing the composite index as the third objective is also discussed in the proposed optimization model. The proposed tri-objective portfolio optimization model is described below:

$$Minimize\ \sigma_p = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \tag{9}$$

$$Maximize\ \mu_p = \sum_{i=1}^{N} x_i \mu_i \tag{10}$$

$$Minimize\ V_p = \sum_{i=1}^{3} d_{p_i} \tag{11}$$

Where d_{p_i} can have one of the values from any of the three decision parameters viz. a_i q_i and d_i.

The value of V_p is determined using aggregation of these three decision parameters viz. a_i q_i and d_i. The user preferences about these three decision parameters allow for selecting an appropriate weight for a decision parameter in the aggregation operation.

Risk in a portfolio optimization model serves as a first objective, and it represents a deviation of the rate of the return. It is described as a $N \times N$ matrix that contains

co-variance values represented by a parameter σ_{ij}. The value of this parameter is calculated using the correlation value between i^{th} and j^{th} asset. Return in the form of its expected value for a portfolio is used as the second objective. The second objective is calculated by summing up the weights of returns μ_i for all the assets in the portfolio. The model also comprises limitations and bound values that need to be satisfied with the objectives described above. These limitations and bound values vary with each kind of problem. The third objective in the proposed portfolio optimization model is based on the values of three important decision parameters. Moreover, an upper bound value, which is representing a composite index as the third objective, is formulated in the proposed model. This objective minimizes an upper bound value of decision parameters.

The outputs obtained for the solution of a portfolio optimization problem; do not change drastically with a corresponding change in the value of the decision parameter a_i[2]. Therefore, the portfolio's composition is relatively stable corresponding to changing values of the decision parameter a_i. The solution of the portfolio optimization problem does not change significantly with an increment of 20% in the value of decision parameter

a_i [2]. The value of the decision parameter q_i affect the composition of the portfolio. If the value of q_i, for a specific sub-portfolio, is highest then the expected return would be highest for that sub-portfolio [2]. Choosing a higher value of d_i could lead to lower values of the expected return, which is occurring because of a trade-off between return and risk. Thus, minimizing the value of the decision parameter d_i may prove beneficial for obtaining the desired value of the expected return.

The algorithm used in the proposed portfolio optimization model is made to run in the starting with an equal weight assigned to three decision parameters viz. a_i q_i and d_i. Next, the mean and standard deviation values of these three parameters, which are obtained by running the algorithm, are used to formulate an upper bound for the value of the composite index, and this composite index is representing the third objective in the proposed portfolio optimization model. The standard deviation values for these parameters play a significant role in modelling the variation of the parametric values from their mean values. The solution obtained using these deviations represents an optimal value based on an aggregation scheme using selective weights for these decision parameters.

The selection of weights, which are assigned during the aggregation process, is a crucial phase in the formulation of a composite index. The weights used for the decision parameters in the proposed model need to be chosen appropriately. A variety of schemes may be adopted for assigning weights to the decision parameters. The discussion on weighting schemes that are used in a practical situation is given in [89]. It is observed that the available schemes are not sufficient for a practical application. A scheme used in data envelopment analysis (DEA) method changes weights based on the country of data [90]. This scheme has a drawback since it does not produce optimal solutions in case of comparative

analysis. Next, the usage of equal weights was adopted to take into account the changes in the weighting schemes. The scheme based on equal weights was observed to be the most prominent choice because it uses a more straightforward comparative analysis and allows for suitable changes in the weights [89, 90]. Whereas the other schemes offer a careful comparative analysis, still these do not produce optimal solutions [89, 90]. The scheme used in the proposed model for assigning weights to the decision parameters is based on the deviation of the weights from their equal values.

6.3.3 Description of the Third Objective

This objective is based on the concept of extending a basic portfolio optimization scheme into a domain of investors' preferences regarding decision parameters. The real-world decision-making process of portfolio optimization is best described with the modelling of the third objective (V_p), which is based on the three parametric values viz. (a_i q_i and d_i).

Since all the relevant information for portfolio optimization cannot be captured only in terms of return and risk only, the formulation of the third objective is done for protecting from the risk of potential losses occurring due to numerous and interconnected portfolios. This objective serves as a composite index that manages decision-makers preferences about three key decision parameters such as sustainable loss value (d_i), tabular variable (a_i) which signifies suitable equilibrium among return and variance level of the objective in the proposed model and ($a_i$that accounts for proportional weights assigned to categories in portfolio or sub-portfolios. This criterion might be of equal, if not higher, importance as return and risk objectives, in the portfolio optimization problem. The equation used for representing the third objective is given below:

$$V_p = a_i + q_i + d_i \tag{12}$$

The decision parameter (a_i) signifies the relative importance of two objectives viz. return and variance. The decision parameter (d_i) indicates sustainable loss value, and it is related to CVaR value. The decision parameter (q_i) defines the proportion of the capital allocated to sub-portfolios.

6.3.3.1 Description of an existing model for portfolio optimization based on decision parameters

A discussion on designing novel schemes for finding the solutions of multi-

objective problems based on stochastic methods is given in [2]. The scheme intended involves a set of inequalities with a specified probability level (p_q). This probability level represents a term used for specifying reliability or service level in the algorithm, and its value lies in a range between 0 to 1. The probability level varies between its lowest desirable level (p_{q0}) to the highest value of 1. The scheme considers approaches based on centralized and decentralized investment policies; the models are formulated using one of these approaches.

One such formulation is based on the values of the probability level (p_q) and a decision parameter (a_q), which represents a trade-off between the expected return and reliability. The decision parameter (a_q) is calculated using maximum and minimum values of expected return and reliability.

The value of the decision parameter (a_q) is computed using the following equation [2]:

$$a_q = (er_{max} - er_{min})/(\rho_{max} - \rho_{min}) \tag{13}$$

where er_{max} and er_{min} is the maximum and minimum value of the expected return, respectively. Similarly, ρ_{max} and, ρ_{min} is the maximum and minimum value of reliability, respectively. The objective in this formulation is represented as shown below [2]:

$$\max \sum_{j \in J} (\mu_j \sum_{i=1}^{r^*} x_{ij}) + a_q p_q \tag{14}$$

The variable J represents financial securities, and each security $j \in J$ has a stochastic return ξ_j with the mean return μ_j and belongs to one or several of the r^*assets classes i.

The probability p_q is significant for specifying the tolerance level of risk and the decision parameter a_q is significant for a trade-off between expected return and reliability. The parameter a_q is assigned a value chosen from three possible values: an initial value (a_{q1}) which is computed using equation 13, another value $a_{q2} = 0{,}80\ a_{q1}$, and $a_{q3} = 1.2\ a_{q1}$.

Second decision parameter d_{ai} is used to formulate a constraint in the model, it represents loss associated with a sub-portfolio, and this loss is bounded by the value of decision parameter d_{ai}. The loss should not exceed the value of the decision parameter d_{ai}. Each sub-portfolio may be assigned a distinct value of the decision parameter d_{ai}.

The constraint can be represented in a simpler way [2]:

$$\textit{subject to} \quad \mathbb{P}(-\sum_{j\in R_i}\xi_j x_{ij} \le d_{ai}, \quad i = 1,\dots,r^*) \ge p_q \tag{15}$$

$$\xi_j x_{ij} \le d'_{aj}, \qquad j \in J \tag{16}$$

The constraint specified in equation (15) signifies that the loss for a sub-portfolio must not exceed d_{ai} with a probability level p_q.
The constraint specified in equation (16) signifies an upper bound for the loss of an asset. Further, the scheme considers another decision parameter d_{ai}, which represents the portion of the capital that is allocated to a sub- portfolio category. This decision parameter signifies that q_{ai} a portion of the capital is allocated to i^{th} sub-portfolio category.
The following equation signifies that the entire capital amount is invested.

$$\sum_{i=1}^{r^*} q_{ai} = 1 \tag{17}$$

The scheme also includes other constraints like concentration constraints, short-selling constraint, etc. in the model [2]. The second model of the portfolio optimization problem assigns the value of decision parameter d_{ai} for each sub-portfolio, and it is considered as a variable rather than having a specific parametric value [2].
In this formulation, the objective is comprised of an aggregated value for the loss rather than including a term for the expected return. This loss level must not exceed the probability level p_q. Therefore, the objective in this formulation is composed of the following weighted loss equation:

$$\max -\sum_{i=1}^{r^*} q_{ai} d_{ai} + a_q p_q \tag{18}$$

The constraints used in this model are similar to the limitations, which are used in the first model. The weighted loss equation (18), is used for representing the parameter d_{ai} as a decision variable in the scheme. Previously, d_{ai} the parameter was assigned a predetermined and specific value in the formulation. Now, the variations of the decision parameter (d_{ai}) are inculcated in the formulation. The significance of treating (d_{ai}) as a decision parameter lies in allowing variations of d_{ai} in the model.

6.3.3.2 Role of the decision parameters in the formulation of the third objective in the proposed tri-objective model for portfolio optimization

This section describes three significant decision parameters, which are used in the third objective and gives a formulation for these decision parameters. Further, a discussion on the concept of formulating composite index with a strategy to incorporate an upper bound for the composite index, which can facilitate an investor regarding the choice of a desirable level for this upper bound. The composite index formulated in the proposed tri-objective
model uses three decision parameters viz. a_i q_i and d_i, and a single index is formulated using aggregation of these parameters.
First parameter a_i signifies relative importance of the expected return objective and a level of the variance that is used in the model.
The value a_1, which is used to initialize a_i decision parameter, is calculated using boundary values of expected return and variance, and it is calculated using the following equation:

$$a_1 = \frac{er_{max} - er_{min}}{var_{max} - var_{min}} \tag{19}$$

The variables var_{max} and var_{min}represent the maximum and minimum value of variance, respectively.
The variations in the value of a_i are considered for assigning an appropriate value to this decision parameter. If a_i is assigned a value that exceeds the value of a_1, then variance becomes more significant in the portfolio optimization model. This observation is deduced from equation (19). Similarly, if the decision parameter (a_i) has a value less than a_1 then the expected return objective becomes more prominent and significant in the portfolio optimization model.
The second decision parameter used in the proposed tri-objective problem is a_i. This decision parameter indicates the proportion of the allocated capital to a sub-portfolio. If we allocate an equal amount of capital to each sub- portfolios, then the value of the decision parameter qi is 1/3. If three categories of sub-portfolios are present in the portfolio optimization problem then the assignment of equal proportion to each of the category, implies that the value of decision parameter q_i is 1/3 for all the three categories. Whereas, in other schemes, a problem instance can be formulated for portfolio optimization problem using the value of decision parameter qi as: the first category of sub-portfolio is assigned 40% of the capital amount (i.e. $q_1 = 0.4$), second and third categories of the sub- portfolios are assigned 30% of the capital amount for each category (i.e. $q_2 = 0.3, q_3 = 0.3$). In this scheme, the minimum value of the decision parameter q_i is 0.3 and maximum value of the decision parameter q_i is 0.4. Upper bound value of q_i needs to be minimized, and this minimization is required for investing a sufficient proportion

of the capital amount for each category of the sub-portfolios.

6.3.3.3 Description of new algorithms based on decision parameters that are used in the proposed tri-objective model of portfolio optimization

The value of the decision parameter a_i represents the relative importance of the calculated value of expected return and variance. First decision parameter (a_i) is calculated using the vectors of expected return values and variance levels. The initial value of a_i is a_1, which is calculated using equation (19).
The calculations for a_i are performed based on sorted values of the Sharp ratio, and the Sharp ratio is computed by dividing the expected return vector by variance levels. The deviations in the values of the decision parameter a_i is calculated using a drift parameter in the algorithm.
The value of the decision parameter a_i is allowed to change within a range consisting of a minimum value = a_1 and a maximum value = 1.3 * a_1. In literature [2], three values are used for the parameter (a_{qi}) in a model described in [2], and these are a_{q1} (an initial value), a_{q2}= 1.2 a_{q1} and a_{q3} = 0.8 a_{q1}. In the proposed strategy, the values of the decision parameter ai are allowed to deviate from the initial value a_1 in a selected range using a drift. The possible variations in the values of a_i are found by changing the drift values. The drift is comprised of an angular deviation which is evenly distributed in a range from 0 to (($\pi/2$)/ n_1), where n_1 is the sample size of variance vector.
The following equation is used to assign a value to drift in every iteration:

$$drift(i) = (a_{imax} - a_{imin}) * \cos(\delta) \quad (20)$$

The variables a_{imax} and a_{imin}represent maximum and minimum values of a_i respectively, δ is angular deviation. The value of the decision parameter a_i for an iteration is given by the following equation:

$$ai(i) = aimax - drift(i) \quad (21)$$

The decision parameter $d_i$represents the allowable loss level for an asset. The decision parameter d_i is calculated using CVaR values in the algorithm.
The decision parameter q_i represents the proportion of capital allocated to a category of sub-portfolio. The proposed algorithm considers three categories of sub-portfolios [2]. The variations in the values of q_i are calculated using minimum Sharpe Ratio value and equally dividing the weights among three categories. The variations are evenly distributed using a step value of 0.01.
For each sub-portfolio category, the choice of selective index is made using a

passing condition, and these are selected from the Sharpe Ratio values vector. Let the Sharpe ratio value for the current index is s1, and the Sharpe ratio for the next index in the vector is s2. The passing condition ensures that the value of the chosen sharp ratio lies in a range between s1 and s2. The index values, which satisfies the passing condition, are selected. A vector is initialized with the initial weights for three categories. The weight for a category is replicated for each selected index. The value of the decision parameter q_i for a category is computed based on the weights assigned for the selected index.

The third objective in the proposed model is based on aggregation values of the decision parameters a_i, q_i and d_i . The third objective is a weighted sum of these parameters in the form of a composite index. The following equation represents the third objective (V_p):

$$V_p = \sum_{i=1}^{3} dp_i \tag{21}$$

The decision parameters a_i q_i and d_i represent the possible values of dp_i in equation (22).

The solution explained in the script has accomplished the design of a new strategy for the portfolio optimization problem using a tri-objective structure. The first and second objectives represent the maximization of the return and minimization of the risk respectively. The third objective introduced in the strategy represents a composite index based on three decision parameters. An upper bound of the composite index is formulated, which includes
the variations in the decision parameters viz. a_i q_i and d_i. The upper bound value can be beneficial for an investor in choosing a desirable value for it.

In the proposed tri-objective portfolio optimization scheme, deviations from the mean values of the decision parameters represent variations in the weights of the decision parameters. The different outputs obtained by including these deviations in the proposed tri-objective model are used for comparative analysis. The third objective, introduced in the study, provides an option for an investor to select a desired level of the composite index more quickly and simply. The weights assigned to decision parameters have equal magnitudes at the starting of the algorithm. Whereas, the solutions computed for the portfolio optimization problem are comprised of variations in the values of the decision parameters.

6.3.3.4 Modelling of an upper bound value in the proposed third objective

This study encapsulates a new multi-objective model for the portfolio optimization

problem, to include the efficacy of decision parameters on a portfolio. The third objective serves as a single index, which manages decision-maker precedence about three key decision variables viz. a_i q_i and d_i. These decision parameters are significant for framing a multi-objective portfolio optimization model that can find optimal solutions in terms of investors' preferences regarding the decision parameters. The values of these decision parameters are selected based on deviations from their mean values, which occur due to the existence of several interlinked portfolios. Since the predictions about expected returns and chances of losses in the portfolio are generally vague due to messy investment returns, there is a need for formulating a single composite index using appropriate weights for decision parameters.

The values of variance in decision parameters need to be chosen by an investor based on the expected return's desirable value. Therefore, performing a multi-dimensional Pareto analysis becomes complicated, and under this situation, the task of describing the shape of an efficient frontier is challenging. The proposed tri-objective model can help decision-makers who need to obtain optimal solutions with a desirable risk level and a preferable level for the composite index value. The trade-off that is existing among several random outcomes of multi-objective portfolio optimization problems required to be formulated more readily for the investors, which is the principal motivation for this study.

The proposed strategy focuses on minimizing an upper bound for the desired value of the composite index and a probability of an event that third objective level V_p is less than the highest desirable level h_d. Let r_n be a random variable representing an expected value of V_p, distributed with mean μ and variance σ^2. Assume h_d is highest desirable level. Then, by Tchebycheff's inequality [5]:

$$P_r\{|r_n - \mu| \geq \mu - h_d\} \leq \frac{\sigma^2}{(\mu - h_d)^2} \tag{23}$$

Therefore, one can certainly state:

$$P_r\{\mu - r_n \geq \mu - h_d\} \leq \frac{\sigma^2}{(\mu - h_d)^2} \tag{24}$$

i.e.

$$P_r\{r_n \leq h_d\} \leq \frac{\sigma^2}{(\mu - h_d)^2} \tag{25}$$

So that minimizing $P_r\{(r_n \leq h_d)\}$ is equivalent to maximizing $(\mu - h_d)/\sigma$ [5]. This inequality may be represented by the following equation [5]:

$$P_r\{r_n \geq C_1\} = 1 - F_n[\left(C_1 - \sum_{i=1}^{N} w_i a_i + w_2 q_i + w_a d_i)/\sum_{i=1}^{N} w_i \sigma^2)^{\frac{1}{2}}\right] \quad (26)$$

$$Max\ P_r\{r_n \geq C_1\} \quad (27)$$

$$Max\ F = \left(\sum_{i=1}^{N}(w_1 a_i + w_2 q_i + w_3 d_i) - \frac{C_1}{\sum_{i=1}^{N} w_i \sigma^2)^{\frac{1}{2}}}\right] \quad (28)$$

Here, w_i is the proportional weight assigned to a decision variable and C_l is a bounding value for the upper bound of the composite index. F_n is a function using these parameters.

Definition 1. [7] A solution $x^* = [x_1^*, \dots, x_n^*]^T \in X$ is believed to be a Pareto-minimal solution for the triobjective optimization problem if there does not exist a solution $\tilde{x} = [\tilde{x}_1, \dots, \tilde{x}_n]^T \in X$ such that

$$\overbrace{1 \leq j \leq N}^{min}\ f(x_j^*) \underbrace{_{1 \leq j \leq N}}_{min} f(\tilde{x}_j)\ and\ \sum_{j=1}^{N} r_j x_j^* \leq \sum_{j=1}^{N} r_j \tilde{x}_j \quad (29)$$

The inequality based on the upper bound of the composite index (minimum upper bound of probability for the desired value of V_p (third objective)) is given below:

$$P_r\{r_n \leq h_d\} \leq \frac{\sigma^2}{(\mu - h_d)^2} \quad (30)$$

At least one of the inequalities given above must hold strictly to make it a Pareto-minimal solution.
We can transform the above problem into a tri-objective optimization problem by adding another variable p_{uj} (user preference) and N constraints.

$$\max(p_{uj}, \sum_{j=1}^{N} r_j x_j \quad (31)$$

$$s.t.\ p_{uj} \leq f(x_j) \quad j = 1, \dots, N \quad (32)$$

$$x \in X \quad (33)$$

where $p_{uj} \leq f(x_j)$ is the j^{it} constraint for composite index. Since the optimization process will push the value of p_{uj} to be equal to $\overbrace{1 \leq j \leq N}^{min} f(x_j)$ where $f(x_j) = P_r\{r_n \leq h_d\}$, minimizing $f(x_j)$ value is equ8ivalent to maximizing $(\mu - h_d)/\sigma$ from equation (25) [5].

The definition given in definition 3.1 describes the solution of the problem for the third objective considering the composite index for a portfolio optimization problem. This definition explains the existence of a solution under inequality and the upper bound of the composite index.

The formulation is based on the concept that is minimizing $P_r\{r_n \leq h_d\}$ is equivalent to maximizing $p_{uj}(\mu - h_d)/\sigma$. Besides, the user preferences are dependent on the choice of preferences to reject portfolios with a higher probability of lower returns.

Theorem 1. [7] *Assume assets are sorted in a manner such that* $v_1 \geq v_2, \dots, \geq v_n$. *Here,* v_i *indicate the value of the third objective in an iteration i. Then, an integer* $v_k \leq N$ *exist so that*

$$\sum_{j=1}^{v_{k-1}} u_j < 1, \quad and \quad \sum_{j=1}^{v_{k-1}} u_j \geq 1 \tag{34}$$

Here, $u_j uj$ is upper bound for x_j.

and

$$x_j^* = \begin{cases} u_j & j = 1, \dots, v_{k-1} \\ 1 - \sum_{j=1}^{v_{k-1}} u_j & j = v_k \\ 0 & j > v_k \end{cases} \tag{35}$$

$$x = [x_1, x_2, \dots., x_n]$$

$$v_i = [v_1, v_2, \dots., v_n]$$

$$p_{uj} = [p_{u1}, p_{u2}, \dots., p_{un}]$$

Let the assets be sorted in such an order that

$$p_{u1} \geq p_{u2} \geq, \dots, p_{un}$$

Thus , there exists an integer $k_{pu} \leq N$ such that $\sum_{j=1}^{k_{pu}-1} u_j < 1,\ and\ \sum_{j=1}^{k_{pu}} u_j \geq 1$

and

$$x_j^* = \begin{cases} u_j & j = 1, \dots, k_{pu-1} \\ 1 - \sum_{j=1}^{v_{k-1}} u_j & j = k_{pu} \\ 0 & j > k_{pu} \end{cases} \quad (36)$$

and this is an optimal solution to the problem. Here, p_{uj} is vector instead of a variable and the value of p_{uj} can be computed using definition 3.1.
The pseudocode for calculating the values of the third objective (V_p) is given in algorithm 1. The pseudocode for computing the values of the index a_i and index d_i is outlined in algorithm 2 and algorithm 3.

Algorithm 1: Procedure to compute the value of the third objective (V_p)

Input: Population, CVaR value
Output: Value of the third objective (V_pVp) for the given population
1. Store the data of given population _ pop in a variable b_1
2. $b_1 = [pop\ .out]$
3. Let no _ population = length(pop)
4. Initialize expected _ return and variance
5. $expected\_return[i] = b1 * risk[i]$
$// expected\_return[i]\ represents\ ith\ element\ of\ the\ array\ expected\ \_return$
// Same convention is used for other variables in the algorithm
6. $variance[i] = b_1 * return[i]$
7. Calculate the value of index – q_i
8. n _ category = 3
9. Calculate Sharpe Ratio (sratio)
10. $stratio[i] = \frac{Expected\_return[i]}{Variance[i]}$
11. Find maximum and minimum values of sratio
12. $max\ \_\ sratio = max[sratio], min\ \_\ sratio = min[sratio]$
13. Find the difference between max and min values and store it in variable diff. and array
14. $S = min\ \_\ sratio$
15. Find the value of S parameter
16. **for** $each\ category$

17. $S[i] = S[i](\min\_ sratio + (1/category(i)) * diff), \quad i = 1,2\ and\ 3$
18. **end**
19. Initialize $count[\ ]\ and\ categories = [\ ]$
20. $S[4] = S[4] + (S[4] * 0.01)$
21. **for** $each\ category(i), call\ the\ function\ find\ idx$
22. $idx = (sratio \geq S(i)\ and\ sratio < S(i+1))$
23. $count = [count\ sum(idx)]$
24. Add indexes satisfying the criterion
25. $category(i) = find\ idx$
26. **end**
27. Initialize weights (wts) [0.2, 0.3, 0.5]
28. $acategory(i) = category(i)$
29. $t = wts(i) * acategory(i)$
30. $qi = t$
31. Compute the index (ai) // see Algorithm 2
32. **for** $i\ in\ 1\ to\ no\_ population$
33. $Compute\ the\ index\ (di)\ using\ CVaR$ // see Algorithm 3
34. $Vp = qi + ai + di$
35. $pop(i).out.Vp = Vp$
36. **end**

Algorithm 2: Computing values of ai index for the relative importance of expected return and variance

---

Input: Expected return values (objective), Variance values (objective)
Output: Computed values of index (ai)

1. Initialize the parameter: no_ population = length (Variance)
2. Compute the values of Sharpe Ratio (sratio) for each index i in 1 to no _ population
3. $stratio[i] = \frac{Expected\_return[i]}{Variance\ [i]}$

4. Sort the values of sratio (i) for i in no _ population
$sratio(sorted\_ index) = sort(sratio(i))$
5. Initialize the deviation factor [dw ($dw(i) = 0$))] for $i\ in\ 0\ to\ no\_ population$
6. $drift(i) = dw(i)$
7. Initialize ai
8. $ai(i) = dw(i) \quad 1 \leq i \leq no\_ population$
9. Find the values of $emax(i) = max(expected\_ return\ (i))$, $emin(i) = min(expected\_ return\ (i))$,
$vmax(i) = max(variance\ (i))$ and $vmin(i) = min(variance\ (i))$
10. Initialize $emax = emax(i)$, $vmin = (i)$
11. $a = (emax - emin)/(vmax - vmin)$
12. Select the minimum and maximum values of ai
13. $aimax = a$, $aimax = 1.3 * a$
14. Select angular deviation values
15. $angular\_ deviation = 0$, $dw(1) = angular\_ deviation$, $drift\ (1) = (aimax -$

$imin$)
16. Select initial $ai(i)$ values
17. $ai(i) = aimax - drfit(1)$
18. Calculate values of $ai(i)$
19. **for** $i = 2$ to $no\_population$
20. a. Change angular deviation
21. $ang = ang + (pi/2/npop)$
22. $dw(i) = ang$
23. b. Find new drift value
24. $drift(i) = (aimax - aimin) * cos(dw(i))$
25. c. $ai(i) = aimax - drfit(i)$
26. **end**
27. Sort the values of ai

Algorithm 3: To calculate the value of the parameter di ($\approx CVaR$)

Input: Expected returns for each asset
Output: Value of di

1. Compute the return series for each asset present in the portfolio
2. Compute $ER(w(t))$, the expected returns at different historical time instances $t \in H_T$
3. Arrange the values of $\{ER(w(t)\}_{t=1}^{N_T}$ in their ascending order. The ascending order of time stamps are $\{t_a, t_b, \ldots T_{N_T}\}$
4. Calculate $\beta\_Value\_at\_Risk$ (βVaR)
5. $\beta_{VaR}(w(t)) = \inf[ER\left(w(t_q)\right) | \sum_{t=t_i}^{t_q} p_t \geq 1 - \beta],\ where\ p_t = \frac{1}{N_T}(uniform)$
6. The parameter $d_i(\approx CVaR)$ is calculated using β_{VaR} using the following equation:
7. $d_i(w(t_i) = \frac{\sum_{i=1}^{N_T} ER(w(t))(-ER(w(t)) - \beta VaR(w(t)))^+}{\sum_{i=1}^{N_T}(-ER(w(t)) - \beta VaR(w(t)))^+}$

The value of the third objective in the proposed model is computed using equation (3). This objective signifies the investors' preferences regarding decision parameters. Its value is dependent upon three decision parameters used in equation (3). The description provided in the section above gives an insight into a novel scheme to fix an upper bound on the value of the third objective. This upper bound has been formulated in the form of a composite index successfully. This formulation is beneficial to put an upper bound on the level of deviations that may be appropriately chosen by an investor.

6.4 Modifications of MOEAS (multi-objective evolutionary algorithms)

Taking up what we saw in chapter 2 in section 2.8, a general framework of a multi-objective optimization problem with conflicting objectives is described as follows:

$$Minimize \quad F(x) = (f_1(x), \dots, f_m(x))$$

$$s.t. \qquad x = (x_1, x_2, \dots., x_N) \in X$$

where, $x = (x_1, x_2, \dots., x_N)$ presents a vector of decision variables, and X is the number of possible solutions and $F(x)$ is an objective function vector that contains values of m objectives maps the possible set X into the set F which represents all possible values of the objective functions. All objective functions may be maximized, minimized, and be in a mixed manner. The general procedure in multi-objective is to search the entire Pareto optimal and non-dominated solutions, i.e., any solution we cannot improve on one objective function without worsening on another.

A decision variable x is mentioned as the dominating variable over decision variable y if the following conditions are satisfied,

a) $f_i(x) \leq f_i(y)$ it holds $\quad \forall i \in \{1, \dots, m\}$

b) $f_i(x) < f_i(y)$ is valid for a single value of $i \; \forall i \; \in \; \{1, \dots, m\}$

It is written as $F(y) \leq F(y)$. Now, a set of Pareto optimal solutions is described as

$$P = \{x \in X | \nexists y \; \in X, \qquad for \; which \; F(x) \leq F(y)\}$$

The multi-objective optimization problem can be converted into a single objective optimization problem using the parametric approaches, of which the weighted sum approach is one of them. A single run of the algorithm does not generate the set of optimum trade-offs amidst objectives. Moreover, in the case of the non-convex problem, the best approximate solutions of objectives are considered as Pareto optimal set. The searching techniques are employed in a MOEAs are generally using the concept of the population which is framed for obtaining a set of best approximate optimum solutions of trade-offs. Since no assumptions are made regarding the framework of the problem's objectives and constraints, it allows them to solve complex optimization problems to estimate a set of solutions near the Pareto front. Successfully, MOEAs have been used in different types of real-world problems such as biomedical problems [61], scheduling problems [62], and

wireless sensor networks [63].
We handled the CCPO with a composite index along with its upper bound (tri-objective portfolio optimization problem) using MOEAS in this study. The four state of the art MOEAs is used by comprising repair mechanism and candidate generation method, which are proposed by [20]. A general framework used to modify MOEAs is described in the following subsections.

6.4.1 Encoding

The candidate solution of the portfolio is represented as a single real-valued vector of size N. The method is based on candidate generation is used to handle the decision variables ($z_i s'$). This kind of representation has advantages in terms of higher space complexity, and the recombination procedure can be used to selective assets for efficient subspace exploration.

6.4.2 Population initialization

First, the initial population is initialized randomly for generating a candidate solution below steps are followed.
1. Firstly, the value of cardinality k is selected from a set i.e. $k \in \{k_1, k_1 + 1, \dots, k_2\}$
2. In the index set $\{1, 2, \dots, N\}$ I_p, select k_p position randomly out of N_p position for allocating the weights.
3. For k_p position generated in step 2, generate uniformly distributed random variable between their corresponding boundaries.
From the steps mentioned above, it becomes clear that the initial population is infeasible. Notably, the initial population's solutions adhere to realistic constraints such as cardinality, pre-assignment, floor, and ceiling but do not ensure that these solutions fulfill the budget constraint, so repair mechanism is required to repair the solution of the initial population.

6.4.3 Candidate generation

The critical part of any evolutionary algorithm forms by effective candidate generation. This move directly affects the algorithm's capability of exploration and exploitation. In the past studies, researchers have focused on developing novel approaches for candidate generation in evolutionary computation. Recently, S. Meghwani *et al.* [20] proposed a cardinality constrained bounded exponential crossover (CCBEX), which not only generated offsprings within variable

boundaries as well as controlled the pre-assignment and cardinality constraints. In this study, we have used the CCBEX crossover to generate offsprings, and a full description of CCBEX is presented in [20].
To enhance the ability of the algorithm in terms of diversity and searching ability, we have used a swapping operator for mutation for every portfolio $x_s(t)$ such that the searching ability could reach unallocated space. This operator exchanges the values of the weights for two-element in each candidate generation with a probabilistic value. Assuming $x \in X$ is portfolio used for mutation operator and p and q are arbitrarily selected location, which follows the given conditions:

1. $x_p = 0$
2. $l_q \leq x_q \leq u_q$

The value of a candidate solution x after applying the mutation operator is given below:

$$\bar{x} = \begin{cases} l_p + \dfrac{x_q - l_q}{u_q - l_q}\ (u_p - l_p) & if\ \ i = p \\ 0 & if\ \ i = q \\ x_i & if\ i \neq p\ and\ i \neq q \end{cases} \tag{37}$$

The newly updated value of a candidate solution $\bar{x}$ preserves K active locations, and the quantity constraint for p is satisfied, whereas asset q is removed in the portfolio.

6.4.4 Constrained handling

The candidate solutions created by CCBEX do not ensure that the budget limitation is met in the solution generated. Therefore, techniques available for handling the budget constraint in the optimal solution may be adopted, and these techniques are known as constraint handling techniques for an optimization problem. A category of constraint handling techniques based on evolutionary algorithms is presented in [66]. In deciding criteria for keeping the position of assets, the candidate solutions are initialized randomly in the algorithm's starting phase. These solutions are spread evenly within their lower (l_i) and upper (u_i) bounds. A set of candidate solutions satisfying cardinality, floor, and ceiling constraints can be obtained by utilizing the selection method, which is described in [20]. A repair mechanism that takes care of complete investment criteria is described below.

Proposition 1. ([20]). Let us take into consideration, a set of candidate solutions of a portfolio, $x = (x_1, x_2, \ldots, x_N)$ such that $\sum_{i \in I} x_i \neq 1$ and $I = \{x_i > 0 | i = 1,2, \ldots, N\}$, and assume

1. $for\ all\ i \in I,\ l_i \leq x_i \leq u_i$
2. $\sum_{i \in I}\ l_i < 1$
3. $\sum_{i \in I}\ u_i > 1$

$if\ for\ all\ i \in I, x_i'\ is\ evaluated\ using\ the\ following\ equation$:

$$\begin{cases} l_i + \dfrac{(x_i - l_i)}{\sum_{i \in I}(x_i - l_i)} & if \sum_{i \in I} x_i > 1 \\ x_i & if \sum_{i \in I} x_i = 1 \\ u_i - \dfrac{(x_i - l_i)}{\sum_{i \in I}(x_i - l_i)} & if \sum_{i \in I} x_i < 1 \end{cases} \tag{38}$$

Now, the candidate solutions of x' satisfy cardinality, quantity, and budget constraints.

6.4.5 Solution Approaches

This section describes the four multi-objective evolutionary algorithms viz. NSGA-II, SPEA2, MOEA/D, and MOPSO, are used for the proposed tri-objective model. These algorithms are selected because of their unique methodology of selection mechanisms.

6.4.5.1 NSGA-II

In the domain of MOEAs, NSGA-II is considered to be one of the elite and fast non-dominated sorting approaches [57]. The reason for the latter being faster is the adoption of a fast non-dominated sorting algorithm. Moreover, to achieve diverse solutions, a crowding based selection operator was presented. The concept of dominance relation is used to assign ranks to individuals, which is used as a criterion for selection in NSGA-II. Additionally, when deciding among non-dominated solutions, individuals with less crowing distance are given preference. The conventional NSGA-II employs simulated binary crossover as a recombination operator and polynomial mutation as the mutation operator.

6.4.5.2 SPEA2

An advancement over SPEA is SPEA2 that incorporates the concept of k-nearest neighbour's density estimation technique [58]. The reciprocal of the distance to the k-nn is used to compute the density. The aggregate of raw fitness and estimated density is used to estimate fitness. The estimated density uses the truncation method to maintain the consistency of the archive size throughout the algorithm. An improved distribution characteristic of solutions is obtained by SPEA2, in comparison with NSGA-II.

6.4.5.3 MOEA /D

The decomposition-based MOEA algorithm provides for the disintegration of the multi-objective optimization problem into several scalar objective optimization problems. Meaning thereby, the MOEA/D, which decomposes the multi-objective optimization problem into N scalar optimization subproblems and evolves a population of solutions to solve these subproblems simultaneously [59, 62]. The population is composed of the best solution so far found for each subproblem at each generation. The neighborhood relations amidst the subproblems are described based on distances between their aggregation weight vectors. If a subproblem has a weight, which is approximately near to the weight of another subproblem, then these two subproblems may be termed as neighbours. In MOEA/D, the task of finding an optimal solution is achieved by capturing information from these subproblems, which are existing in the neighbourhood. Here, each of the subproblems maintains a solution and puts that solution in the memory. That solution can be regarded as the most optimal solution of the subproblem. The MOEA/D is flexible in its technique for decomposing and finding the solutions of the multi- objective problem. The MOEA/D solves N scalar problems comprising of optimal solutions, instead of creating a single image of the solution of the multi-objective problem. Thus, it utilizes different scaling optimization techniques to find an optimal solution linked with the scalar optimization problem.

6.4.5.4 MOPSO

An optimal solution for an optimization problem can be conceptualized by following the pattern of birds in a flock, which are searching for their food and flying over a specific area. This conceptualization was observed by Kennedy and Eberhart [67]. A new optimization model is formulated, which is called PSO. Assuming $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})$, where the searching space has dimension d then a solution is modelled with the help of particle, and ith particle is represented

by X_i **.** The *i*th particle contains the value of the best position, which represents the best fitness value and it may be represented as $(p_{i1}, p_{i2}, p_{i3}, \dots, p_{id})$. Similarly, the best position in global space is represented as $P_g = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{id})$ and it indicates the location of the best particle, which has the highest fitness value among the population. Every particle is associated with its recent velocity, which is represented as $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id})$. All the particles exchange the information in a fully connected structure. The following equation represents the position and velocity of a particle for the next iteration.

$$V_{id}(t) = wv_{id}(t-1) + C_1 r_1 (p_{id} - x_{id})(t-1) + C_2 r_2 \left(p_{gd} - x_{id}\right)(t-1) \qquad (39)$$

$$x_{id}(t) = x_{id}(t-1) + V_{id}(t) \qquad (40)$$

The total number of particles in a population is N, where $i = 1,2, \dots, N$ and $d = 1,2, \dots, N$. w is the parameter of inertia weight used to manage search space exploration and exploitation. The value of this parameter may be a positive constant or a function, which can be linear and nonlinear function [67] based on timing values. Apart from this, two random values r_1 and r_2 are also used for speeding the search process, and their values are greater than or equal to zero and remain less than or equal to one.

Coello *et al.* [68] proposed an improvised version of PSO (MOPSO) to deal with the multi-objective problems. The MOPSO algorithm follows the Pareto dominance principle for finding solutions to multi-objective problems. It maintains another storage for the second set of the population that contains solutions to the problem, which are non-dominated, and it helps in the searching process for the next generation. MOPSO is based on two parameters of external storage, and these are the storage controller and the grid parameters. Thus, the storage controlling parameter works as a deciding index for adding and deleting a solution in the storage**.** The comparisons of the solutions computed from the main population for each iteration are stored in an external repository, which has a non-dominating nature. Whenever a new solution is added into the storage, the grid space will adapt to keep the solutions that are within the boundary of the recent grid. This grid has a searching space, which is represented by an objective function comprising of a number of regions. The grid space is composed of hypercube structures, and it has components that are equal to the number of objective functions. The usage of a particular operator that is based on the mutation operation is used in MOPSO that augments the searching process. Some dialects of MOPSO are based on the uses of crossover operators because this operator is useful for efficiently searching the output space and finding reasonable optimal solutions in the space of the objective function [69, 70, 71]. The proposed model of the tri-objective problem, also uses a kind of crossover operator (CCBEX).

6.5 Experimental results and discussion

6.5.1 Data description

This study uses five market data sets for performing tests of the tri-objective portfolio optimization model taken from Chang *et al.* [39]. The information contained in these data sets is used for framing UCEF (Unconstrained efficient frontier), values of expected returns, and for constructing the sample matrix of covariances values of assets returns. The time duration used in the data set is comprising of 291 weeks. These data sets are publicly available from Or-library [72]. The set of algorithms discussed in section 6.5 is run for analytical demonstration using the data available in these data sets.

Stock Index	N	k_1	k_2
HS33 (Hang Seng)	31	5	10
DAX100 (DAX 100)	85	5	10
FTSE100 (FTSE 100)	89	5	10
S&P100 (S & P 100)	98	5	10
Nikkei225 (Nikkei 225)	225	5	10

Table 1. Datasets Or-library [72]

6.5.2 Parameters

The parameters used in the algorithms are adjusted to get the best optimal values before using them in the experiment for the HS33 dataset. Besides, the parameters in the algorithms, apart from the size of the population, the size of the archive, and the maximum number of generations, are calculated for their optimal values before being used in the experiment. The best values of the parameter are obtained for the algorithms (NSGA-II, SPEA2, MOEA/D, and MOPSO) are cataloged in Table 2. The algorithms utilize the encoding scheme, candidate generation method, and repair mechanism algorithm described in sections 6.4.1, 6.4.3, and 6.4.4, respectively. The statistics for the experiments are collected by running twenty separate runs for every algorithm and making use of all the five datasets.
The implementation of these algorithms is done using MATLAB (2017a), and these are executed on a Windows operating system with i5 CPU operating at 3.60 GHz and having 8GB RAM.

Name of Parameters	MOEA / D	MOPSO	SPEA2	NSGA-II
Size of Population	100	100	100	100
Size of Archive	-	100	100	-
Number of generation	1000	100	1000	1000
BEX scale parameter	5	5	5	5
Crossover probability	0.9	0.9	0.9	0.9
Mutation probability	0.7	0.7	0.7	0.7
Total number of runs	20	20	20	20

Table 2. Parameters and their values that are employed in algorithms

6.5.3 Performance metric

Due to the unavailability of the true Pareto fronts for the proposed model, the performance metric used for the comparison of adapted algorithms is Hypervolume (HV) [79]. The reasons for considering HV as a performance metric, in comparison to other metrics (generational distance [80], inverted generational distance [81], diversity metric [57] and ϵ indicator [82]) are:

1. The convergence and diversity of the estimated front can be estimated by HV.
2. In the absence of true Pareto fronts, HV is a good option as a performance metric for the comparison of adapted algorithms.

Given approximation set $\bar{P}$, the HV calculates the hypervolume of the space dominated by solutions. A high value of HV indicates convergent and diverse solutions in set $\bar{P}$. For set $\bar{P}$, HV can be defined in the following way:

$$HV\ (\bar{P},\ z^*) = \lambda(U_{p\in\bar{P}}([f_1(p), z_1^*] \times [f_2(p), z_2^*] \times \dots [f_N(p), z_N^*]) \qquad (41)$$

where nadir point is indicated by z^* , Lebanese measure is denoted by λ, and $f_i(.)$ denotes the i^{th} objective among N objectives.

6.5.4 Comparison of adapted algorithms

The aims of the comparison are listed below:

1. To assess the efficacy of adapted algorithms in a constrained manner over five datasets of different lengths.

2. To analyze the impact of pre-assignment constraint in the tri-objective model.

3. The testing is completed using inequality cardinality limitation for the optimization model

4. To comparison the adapted algorithms using the hypervolume metric that is available in MOEA literature.

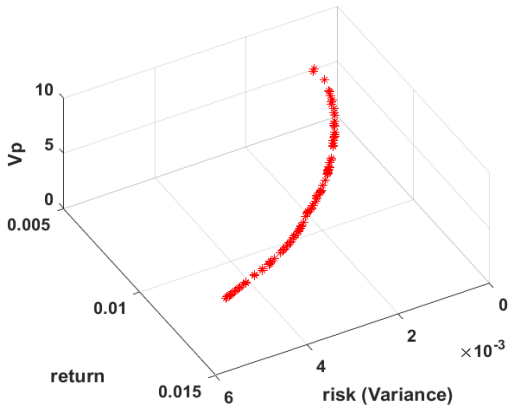
5. To analyze the values of the upper bound of the third objective and its correlation with the parameters used in the model.

The results obtained from the comparison of adapted algorithms are discussed next. A tri-objective model for portfolio optimization is used in the work. This research focuses on a cardinality constraint model, which is based on a standard dialect of the framework described by Lwin *et al.* [46].
The possible range of cardinalities is between k_1 and k_2, which indicates cardinalities values for a portfolio on its Pareto front. These values k_1 and k_2 are commonly quite less than the overall number of assets (N) in a portfolio [20].
If a portfolio is comprised of varying levels of cardinalities, then the resultant efficient frontier has a more significant degree of diversification [20].
The inclusion of cardinalities between a range of k_1 and k_2 is beneficial in a situation where the user has an ambiguous opinion regarding the overall number of assets. This inclusion is also useful in the selection of an optimal range for cardinalities considering realistic limitations.
The results obtained for the efficient frontiers for the proposed tri-objective portfolio optimization model using adapted algorithms viz. MOEA/D, SPEA-2, NSGA-2, and MOPSO based on inequality cardinality constrained are given in Figure 1-5.
It is observed that NSGA-2 and SPEA-2 have better diversification traits for a trade-off amidst the three objectives of the framework. These outputs are obtained for the proposed model under inequality cardinality constraints.
The decision parameters can have diverse values for their corresponding weights. A minimal value of the third objective (V_p) is advantageous in achieving a higher expected return, considering a selected amount of risk in a specified range.
A comparison of these algorithms' outputs, which is based on the hypervolume metric, is presented in Table 3.
The outputs of mean and median levels of hyper-volume metric amidst participating algorithms reflect higher values of diversification for the SPEA2 algorithm for Market 1 (HS33) when the output of this algorithm is compared with other adapted algorithms.
The comparisons carried out for the remaining markets (DAX100, FTSE100, S&P100, and Nikkei225) on the hypervolume metric show that the NSGA-II

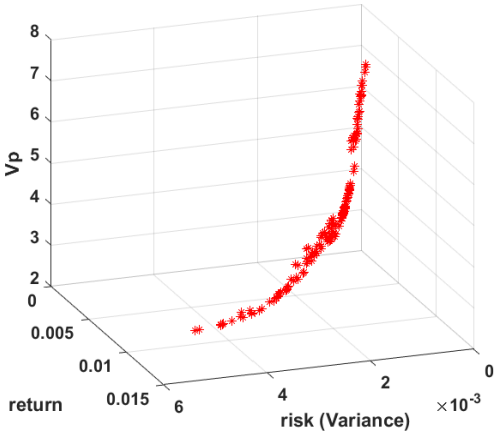
algorithm has a higher diversification level if compared with the results of other adapted algorithms.

The outputs of the hyper-volume metric for all the participating algorithms are shown in Figures 6.

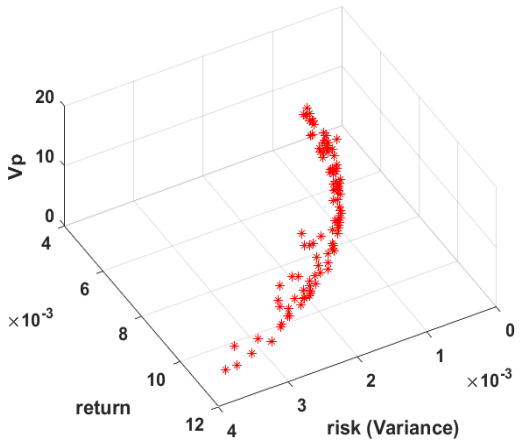
(a) NSGA-II



(b) SPEA2



(c) MOPSO



(d) MOEA /D

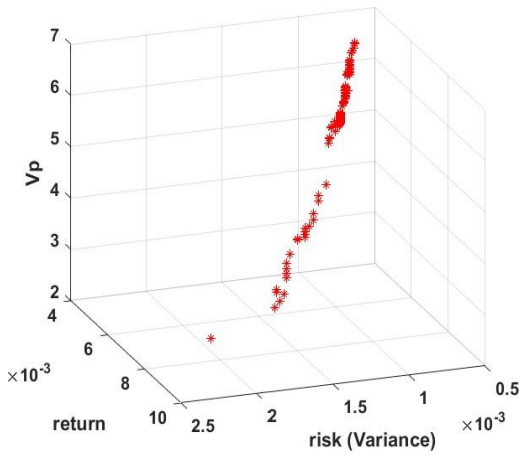
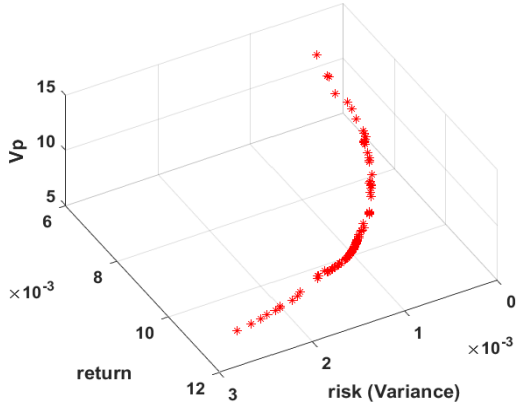
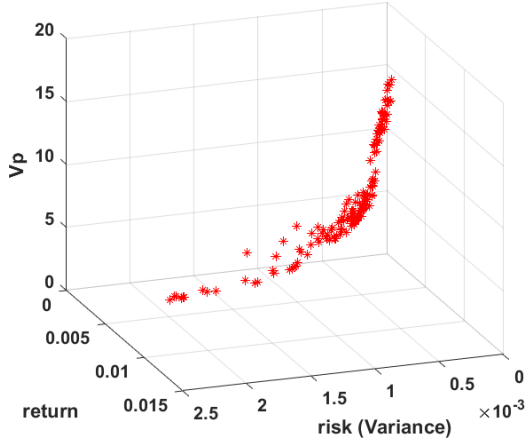


Figure 1: Estimated Pareto frontier of one specific run of four adapted algorithms (Dataset: HS33)

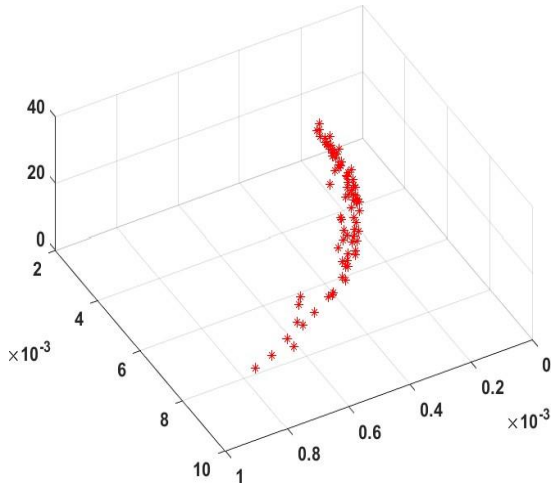
(a) NSGA-II



(b) SPEA2



(c) MOPSO



(d) MOEA / D

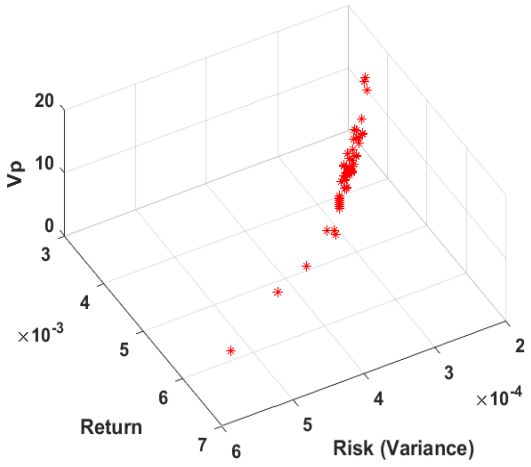
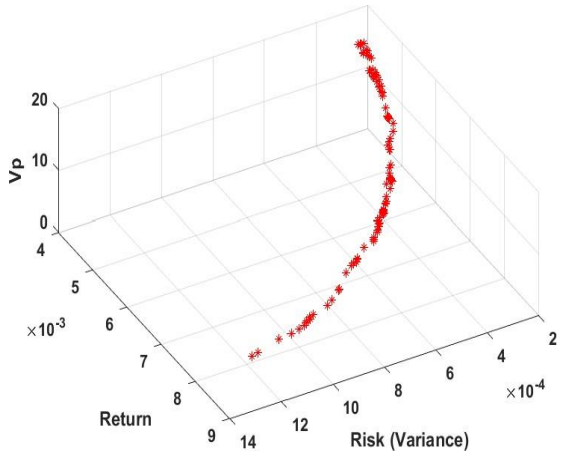
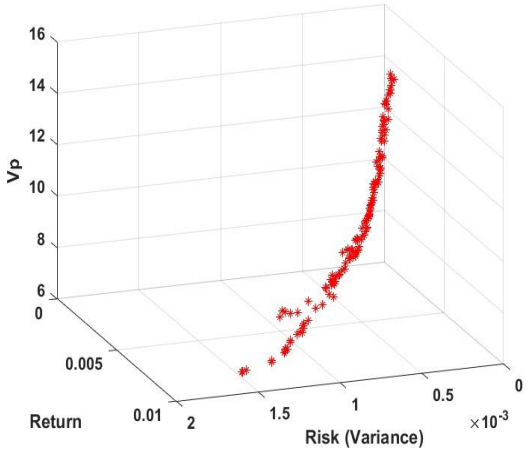


Figure 2: Estimated Pareto frontier of one specific run of four adapted algorithms (Dataset: DAX100)

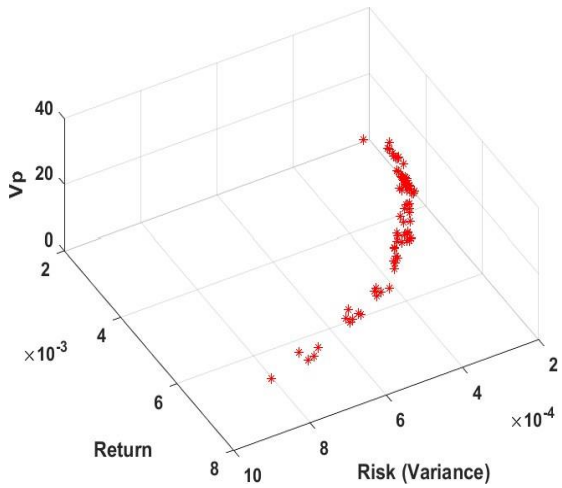
a) NSGA-II



(b) SPEA2



(c) MOPSO



(d) MOEA / D

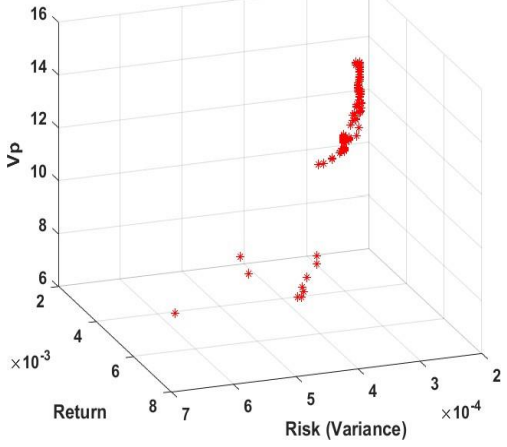
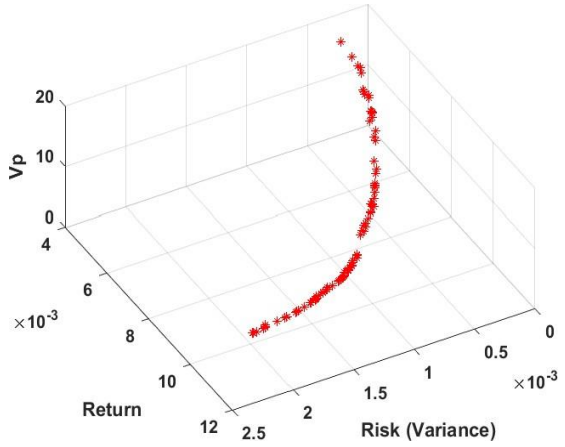
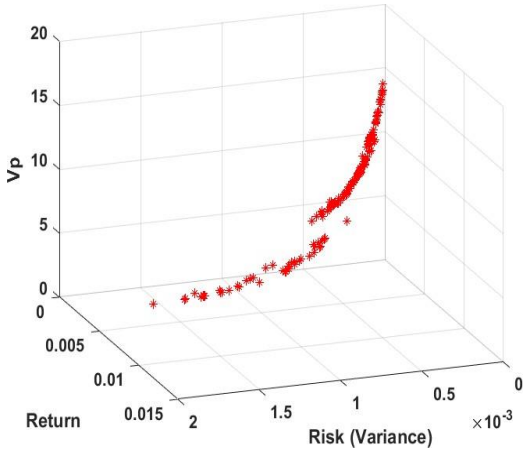


Figure 3: Estimated Pareto frontier of one specific run of four adapted algorithms (Dataset: FTSE100)

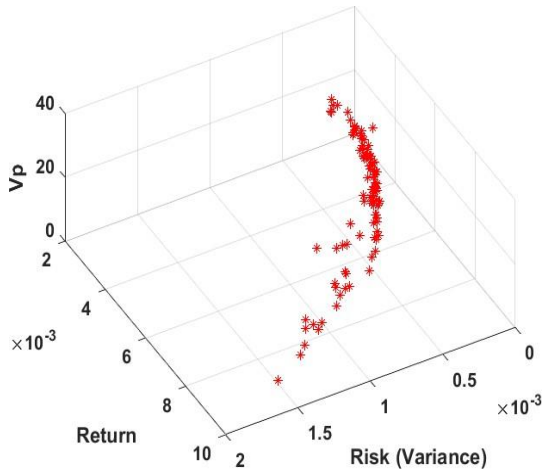
(a) NSGA-II



(b) SPEA2



(c) MOPSO



(d) MOEA / D

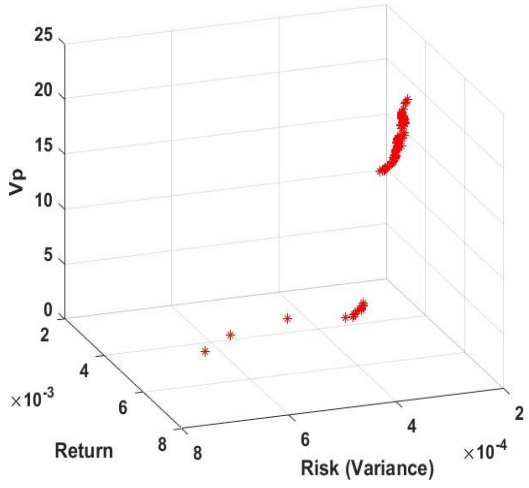
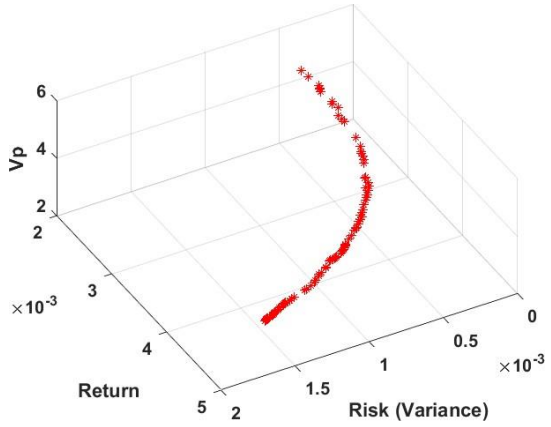
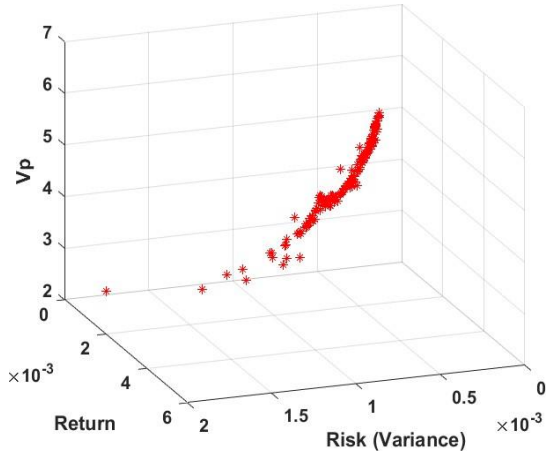


Figure 4: Estimated Pareto frontier of one specific run of four adapted algorithms (Dataset: S&P100)

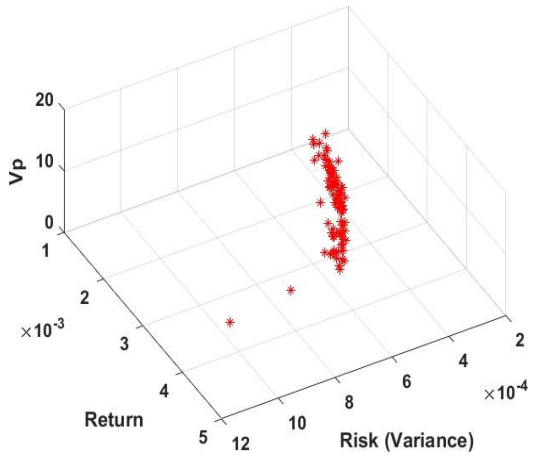
(a) NSGA-II



(b) SPEA2



(c) MOPSO



(d) MOEA /D

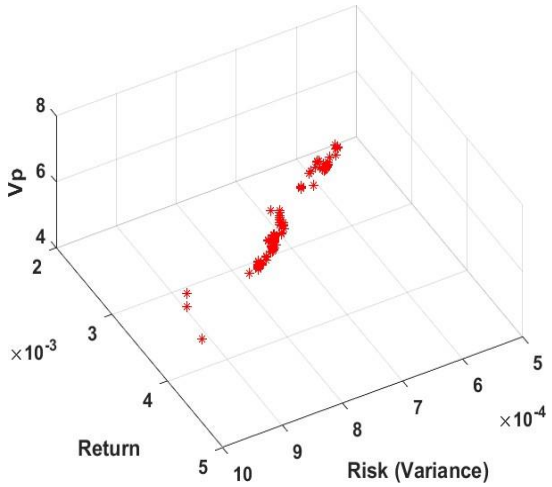
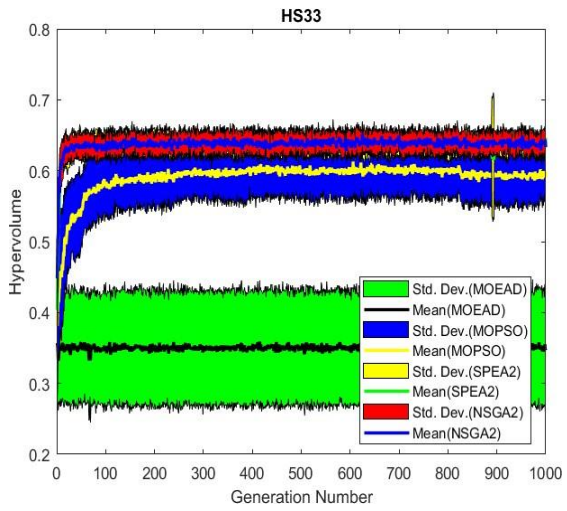


Figure 5: Estimated Pareto frontier of one specific run of four adapted algorithms (Dataset: Nikkei225)

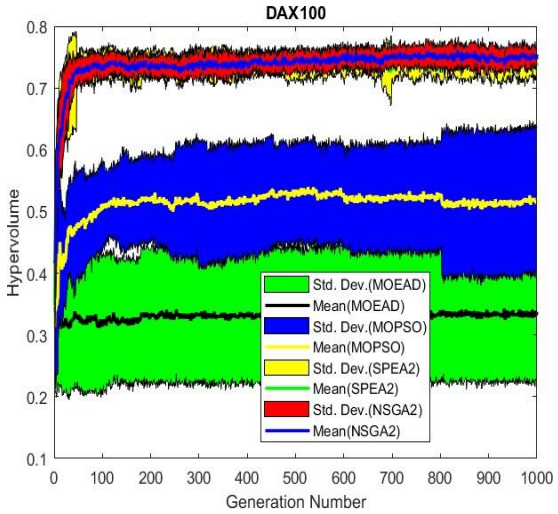
Dataset	Hypervolume Statistics	MOEA/D	MOPSO	SPEA2	NSGA-II
HS33	**Mean**	0.3466	0.5906	**0.6464**	0.6329
	Std. Dev.	0.0790	0.0365	0.0166	**0.0157**
	Max.	0.5030	0.6290	**0.6720**	0.6570
	Min.	0.2090	0.5060	**0.6100**	0.6050
	Median	0.3420	0.6045	**0.6480**	0.6345
DAX100	**Mean**	0.3370	0.5186	0.7352	**0.7545**
	Std. Dev.	0.1117	0.1166	0.0230	**0.0144**
	Max.	0.6360	0.7150	0.7830	**0.7870**
	Min.	0.1830	0.1750	0.6760	**0.7340**
	Median	0.3230	0.5335	0.7355	**0.7530**
	Mean	0.3779	0.5067	0.6185	**0.6644**
	Std. Dev.	0.0978	0.1015	0.1067	**0.0179**
FTSE100	**Max.**	0.6030	0.6260	**0.7010**	0.6920
	Min.	0.2470	0.1970	0.2170	**0.6210**
	Median	0.3810	0.5225	0.6475	**0.6650**
S&P100	**Mean**	0.4829	0.6493	0.6601	**0.6695**
	Std. Dev.	0.0955	0.0816	**0.0162**	0.0222
	Max.	0.6600	**0.7450**	0.6940	0.7170
	Min.	0.3030	0.4010	0.6220	**0.6310**
	Median	0.4950	0.6685	0.6610	**0.6705**
Nikkei225	**Mean**	0.1802	0.4161	0.6064	**0.6697**
	Std. Dev.	0.0810	0.0946	0.0743	**0.0189**
	Max.	0.3280	0.6180	0.6660	**0.7000**
	Min.	0.0540	0.2330	0.3130	**0.6370**
	Median	0.1750	0.4240	0.6280	**0.6690**

Table 3: Performance metric (Hypervolume) statistics of adapted algorithms.

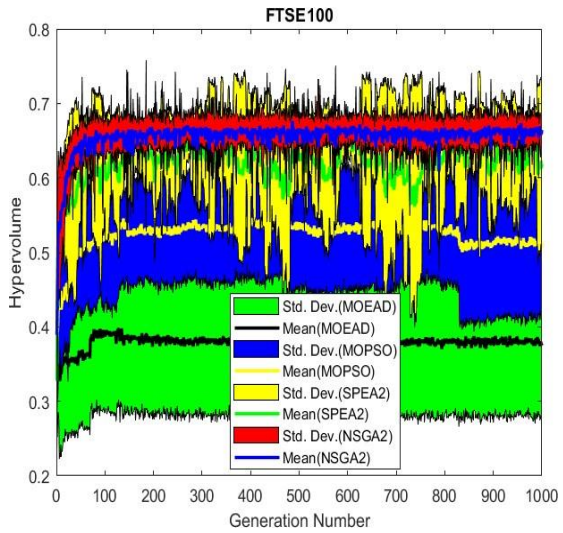
(a)


HS33
Hypervolume
Generation Number
Std. Dev.(MOEAD)
Mean(MOEAD)
Std. Dev.(MOPSO)
Mean(MOPSO)
Std. Dev.(SPEA2)
Mean(SPEA2)
Std. Dev.(NSGA2)
Mean(NSGA2)

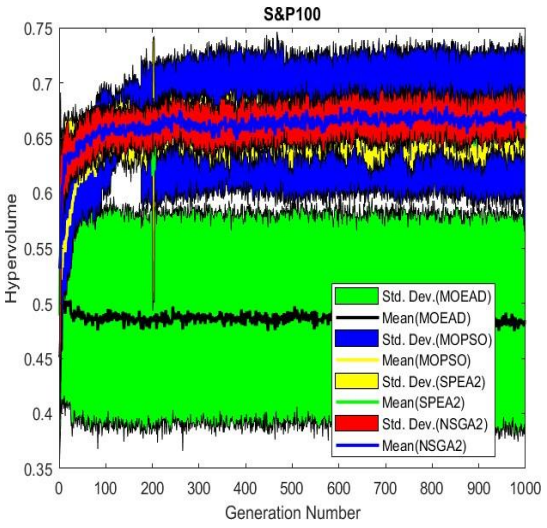
(b)


DAX100
Hypervolume
Generation Number
Std. Dev.(MOEAD)
Mean(MOEAD)
Std. Dev.(MOPSO)
Mean(MOPSO)
Std. Dev.(SPEA2)
Mean(SPEA2)
Std. Dev.(NSGA2)
Mean(NSGA2)

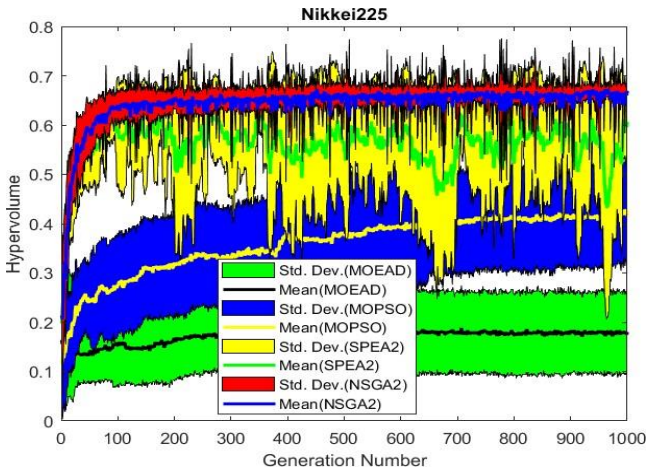
(c)


FTSE100
Hypervolume
Generation Number
Std. Dev.(MOEAD)
Mean(MOEAD)
Std. Dev.(MOPSO)
Mean(MOPSO)
Std. Dev.(SPEA2)
Mean(SPEA2)
Std. Dev.(NSGA2)
Mean(NSGA2)

(d)


S&P100
Hypervolume
Generation Number
Std. Dev.(MOEAD)
Mean(MOEAD)
Std. Dev.(MOPSO)
Mean(MOPSO)
Std. Dev.(SPEA2)
Mean(SPEA2)
Std. Dev.(NSGA2)
Mean(NSGA2)

(e)



.

Figure 6: Average wise evolution of hypervolume metric by generation.

As described in section 6.3, the values of the decision parameters viz. ai, qi, and di, which are used for the third objective in the proposed framework, need to be minimized. The outputs obtained for the values of decision parameters in terms of mean, standard deviation, maximum value, minimum value, and median using participating algorithms viz. MOEA/D, NSGA-II, SPEA2, and MOPSO are given in Table 4, Table 5, and Table 6.

These outputs are obtained by executing the algorithms for 20 runs. Considering the mean and median values of the decision parameter ai, the output results indicate that the NSGA-2 algorithm has higher optimal outputs for Market 1 (HS33), Market 2 (DAX100), and Market 4 (S&P100), while SPEA-2 outperforms for Market 3 (FTSE100) and Market 5 (Nikkei225).

Similarly, based on mean and median values for the decision parameters qi, the outputs show that the MOPSO algorithm outperforms other participating algorithms for Market 1 - Market 5 datasets. A similar analysis of the mean and median values of the decision parameter di indicates that MOPSO outperforms adapted algorithms for Market 1, Market 3, and Market 5.

In contrast, SPEA2 outperforms adapted algorithms for Market 2 and NSGA-II outperforms for Market 4 dataset.

Dataset	Decision Parameter	Decision Parameter Statistics	MOEA/D	MOPSO	NSGA-II	SPEA2
HS33	a_i	**Mean**	5.4487	4.7767	**4.1508**	4.2991
		Std. Dev.	0.5254	0.3110	0.2901	**0.2769**
		Max.	6.4177	5.2463	4.7754	**4.6293**
		Min.	4.6923	4.2194	3.7659	**3.4301**
		Median	5.3108	4.8499	**4.0965**	4.3467
DAX100	a_i	**Mean**	13.0574	13.1481	**8.8214**	9.3931
		Std. Dev.	2.2007	1.2857	1.5511	**0.8641**
		Max.	17.5115	14.9658	11.8288	**11.1267**
		Min.	7.6748	11.3606	**6.4769**	7.9325
		Median	12.9674	13.3246	**8.8830**	9.3996
FTSE100	a_i	**Mean**	11.5952	10.5289	9.4773	**8.8166**
		Std. Dev.	1.2766	0.8720	1.0409	**0.6390**
		Max.	14.0434	12.7575	11.5275	**9.7271**
		Min.	9.1579	9.1042	7.9996	**7.6757**
		Median	11.9037	10.4698	9.5374	**8.8647**
S&P100	a_i	**Mean**	12.1725	10.9533	**8.7972**	9.1867
		Std. Dev.	1.4213	1.0020	**0.8834**	1.0313
		Max.	15.0828	13.3273	**10.6391**	11.2632
		Min.	9.3037	8.86172	**6.8235**	7.5994
		Median	12.2625	10.9028	**8.6810**	8.9393
Nikkei225	a_i	**Mean**	5.6497	4.6237	4.2953	**4.2348**
		Std. Dev.	2.1358	0.4580	0.2968	**0.2182**
		Max.	11.6581	5.3147	4.9880	**4.6575**
		Min.	**3.1630**	3.7114	3.8301	3.9655
		Median	5.5576	4.7031	4.2416	**4.1975**

Table 4: Decision parameter (a_i) statistics for adapted algorithms

Highlighted value denotes the best value and the underscored value reflects the next best value together with every row of the table.

Dataset	Decision Parameter	Decision Parameter Statistics	MOEA/D	MOPSO	NSGA-II	SPEA2
HS33	q_i	**Mean**	0.3651	**0.3261**	0.3264	0.3486
		Std. Dev.	0.0554	0.0216	**0.0181**	0.0212
		Max.	0.4860	0.3770	**0.3580**	0.3840
		Min.	0.2810	**0.2870**	0.2871	0.3140
		Median	0.3570	**0.3260**	0.3270	0.3472

DAX100	q_i	**Mean**	0.3893	0.3382	0.3256	**0.3245**
		Std. Dev.	0.0574	0.0294	**0.0214**	0.0234
		Max.	0.4780	0.3850	**0.3660**	0.3800
		Min.	0.2870	0.2800	0.2920	**0.2760**
		Median	0.4035	0.3435	0.3275	**0.3270**
FTSE100	q_i	**Mean**	0.4168	**0.3445**	0.3651	0.3469
		Std. Dev.	0.0471	0.0234	0.0250	**0.0232**
		Max.	0.4770	**0.3710**	0.4060	0.4130
		Min.	0.3170	**0.2980**	0.3250	0.3190
		Median	0.4160	**0.3500**	0.3650	0.3525
S&P100	q_i	**Mean**	0.3788	0.3573	**0.3187**	0.3242
		Std. Dev.	0.0510	0.0227	**0.0183**	0.0229
		Max.	0.4670	0.3980	**0.3590**	0.3737
		Min.	0.3010	0.3040	**0.2790**	0.2869
		Median	0.3770	0.3625	**0.3190**	0.3247
Nikkei225	q_i	**Mean**	0.3854	**0.3582**	0.4332	0.4277
		Std. Dev.	0.0606	0.0171	0.0168	**0.0114**
		Max.	0.4680	**0.3860**	0.4560	0.4481
		Min.	**0.2560**	0.3200	0.4040	0.4069
		Median	0.3890	**0.3580**	0.4340	0.4303

Table 5: Decision parameter (q_i) statistics for adapted algorithms

Highlighted value denotes the best value and the underscored value reflects the next best value together with every row of the table.

Dataset	**Decision Parameter**	**Decision Parameter Statistics**	**MOEA/D**	**MOPSO**	**NSGA-II**	**SPEA2**
HS33	d_i	**Mean**	0.1798	**0.0685**	0.0789	0.0738
		Std. Dev.	0.0468	**0.0012**	0.0019	0.0021
		Max.	0.2968	**0.0716**	0.0810	0.0774
		Min.	0.1208	**0.0661**	0.0749	0.0691
		Median	0.1684	**0.0683**	0.0785	0.0742
DAX100	d_i	**Mean**	0.1256	**0.0344**	0.0560	0.0431
		Std. Dev.	0.0232	**0.0009**	0.0046	0.0030
		Max.	0.1638	**0.0361**	0.0616	0.0483
		Min.	0.0914	**0.0329**	0.0437	0.0368
		Median	0.1213	**0.0346**	0.0569	0.0440
FTSE100	d_i	**Mean**	0.1125	**0.0349**	0.0423	0.0407
		Std. Dev.	0.0237	**0.0007**	0.0030	0.0019
		Max.	0.1460	**0.0365**	0.0482	0.0442
		Min.	0.0683	0.0337	**0.0366**	0.0375
		Median	0.1168	**0.0347**	0.0426	0.0402
		Mean	0.1048	**0.0360**	0.0492	0.0476

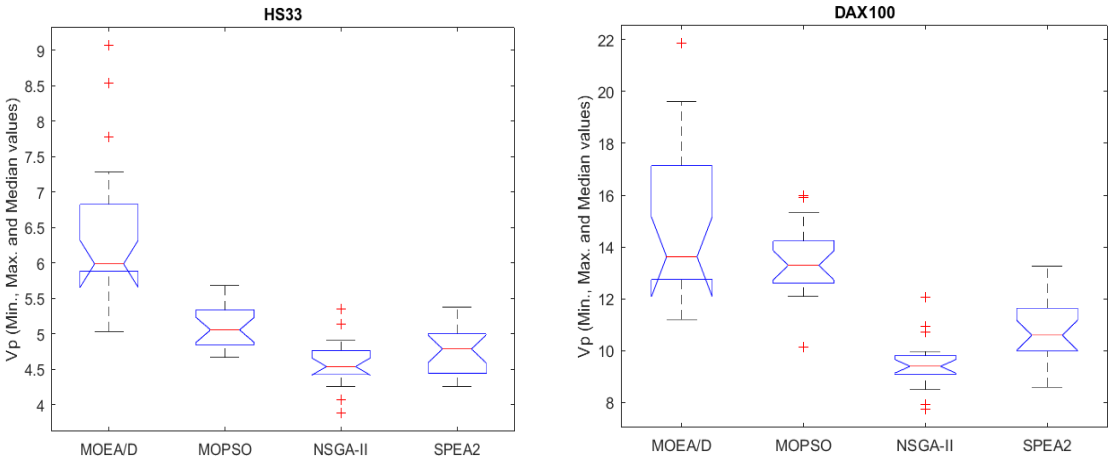
S&P100	d_i	**Std. Dev.**	0.0242	**0.0014**	0.0027	0.0026
		Max.	0.1518	**0.0385**	0.0540	0.0508
		Min.	0.0515	**0.0337**	0.0436	0.0413
		Median	0.1014	**0.0356**	0.0494	0.0480
Nikkei225	d_i	**Mean**	0.1160	**0.0477**	0.0537	0.0497
		Std. Dev.	0.0262	**0.0012**	0.0034	0.0024
		Max.	0.1668	**0.0506**	0.0585	0.0535
		Min.	0.0654	0.0459	0.0461	**0.0456**
		Median	0.1181	**0.0474**	0.0544	0.0498

Table 6: Decision parameter (d_i) statistics for adapted algorithms

Highlighted value denotes the best value and the underscored value reflects the next best value together with every row of the table.

The values of minimum levels, maximum levels, and median values of the third objective in the proposed model computed using adapted algorithms for five market data sets are shown in Figure 7. Based on the statistics of median values of the third objective (V_p), it is found that NSGA-II has better performance over other adapted algorithms for market 1, market 2, and market 5 dataset. At the same time, SPEA2 outperforms for market 3 and market 4 dataset.

(a) (b)



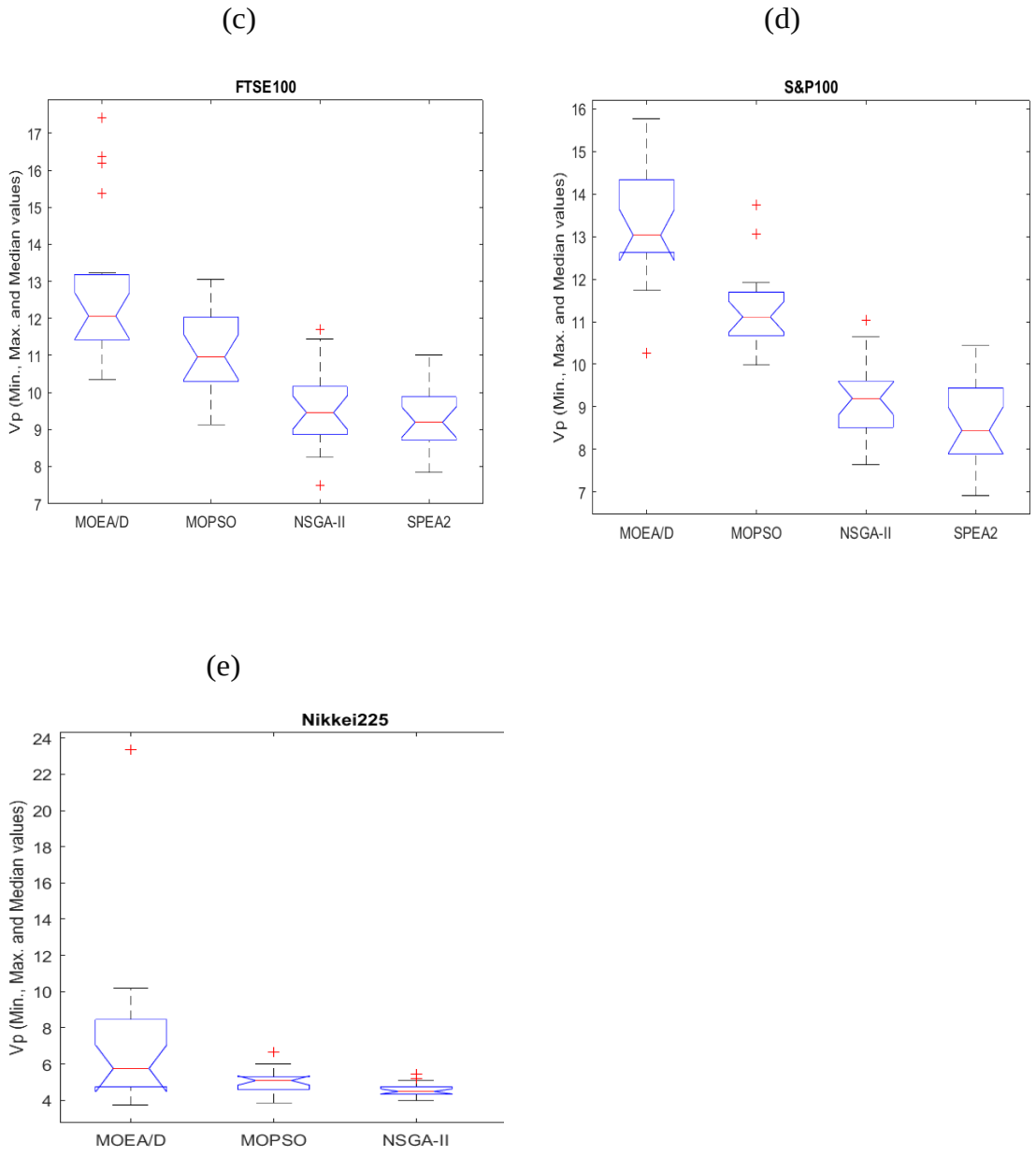
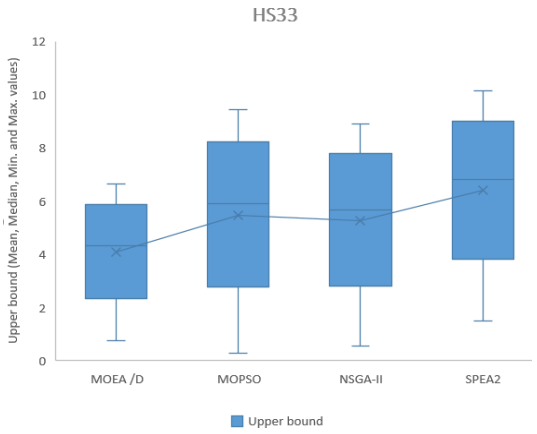


Figure 7: Statistics for the third objective (V_p) of the proposed tri-objective model for adapted algorithms

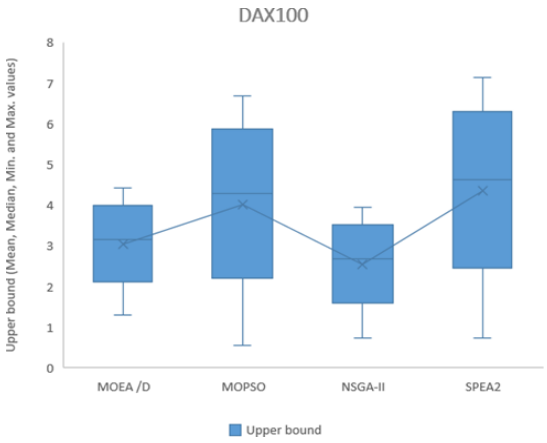
A discussion on upper bound values of the third objective in the proposed model is given next. The values of maximum levels, minimum levels, mean, and median of upper bound of the third objective are computed using equation (28) for the algorithms comprising MOEA/D, MOPSO, NSGA-2, and SPEA-2 for five markets, are shown in Figure 8. Based on the statistics of the upper bound of the third objective, MOEA/D has minimal level for market 1 and market 5, NSGA-II has a minimal level for market 2 and market 3, and SPEA2 has a minimum value for market 4. The upper bound values computed for the third objective are based

on a selection of weights (w_1, w_2 and w_3) in a range of (0-1). These upper bound values are calculated using a predetermined combination of the weights in this range. Investors may choose the different combinations of weights between the range and calculate the optimal values of the upper bound of the third objective.

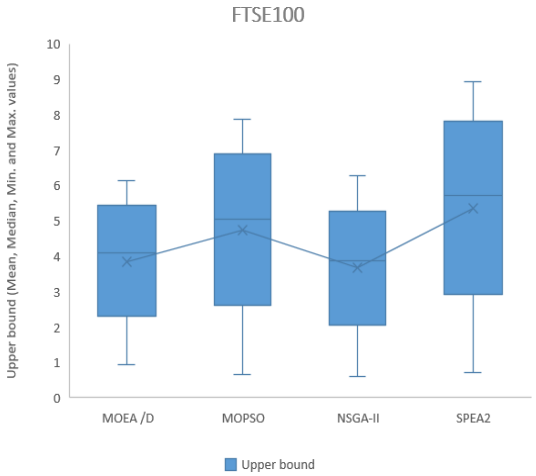
(a)



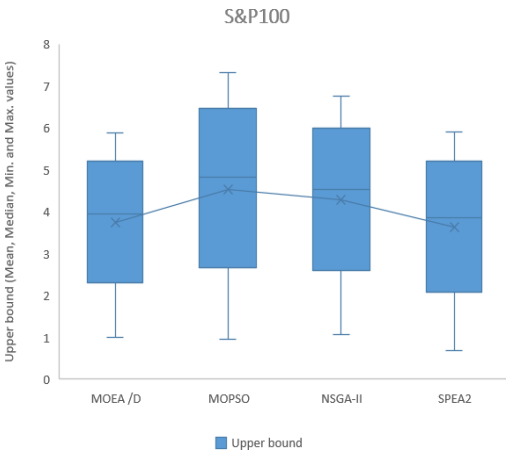
(b)

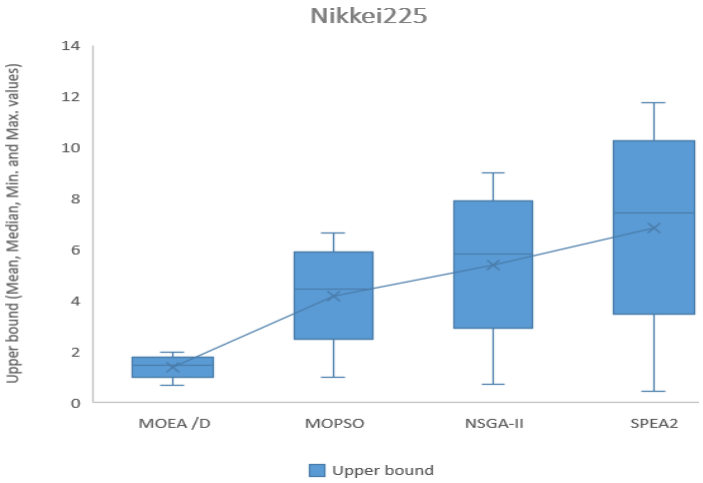


(c)



(d)





(e)

Figure 8: Upper bound values of the third objective of the proposed tri-objective model for adapted algorithm

The correlation amidst the third objective (V_p) values (minimum, maximum, mean value, standard deviation value) and its upper bound values are shown in Table 7 – Table 10 for Market 1 – Market 5 dataset, respectively.

It is observed that third objective's statistical values are positively correlated among themselves.

At the same time, upper bound values are related in a downward slope manner for Market 1 dataset. For market 2 - market 5 datasets, a similar positive correlation exists among the third objective's values.

At the same time, the upper bound values are positively correlated to the third objective minimum and maximum values for market 2 dataset. The upper bound maximum values are positively correlated to the third objective values in most cases for the market 3 dataset.

The market 4 dataset shows a downward slope relation among the upper bound values and the third objective value.

The market 5 dataset indicates that upper bound values are related to the third objective values in a downward slope manner in most cases.

Third objective variable	V_{pmax}	V_{pmin}	V_{pmean}	$V_{pStd.Dev}$	$Upper\ bound_{max}$	$Upper\ bound_{min}$
V_{pmax}	1.0000	0.8105	0.9809	0.9958	-0.9357	-0.0811
V_{pmin}	0.8105	1.0000	0.9087	0.7568	-0.6277	-0.1831
V_{pmean}	0.9809	0.9087	1.0000	0.9597	-0.8784	-0.1320
$V_{pStd.Dev}$	0.9958	0.7568	0.9597	1.0000	-0.9428	-0.3399
$Upper\ bound_{max}$	-0.9357	-0.6277	-0.8784	-0.9428	1.0000	0.2976
$Upper\ bound_{min}$	-0.0811	-0.1831	-0.1320	-0.3399	0.2976	1.0000

Table 7: Correlation data matching to third objective (V_p) values and its upper bound values (minimum, maximum, mean, and standard deviation (Std. dev.) for Market 1 (HS33)

Third objective variable	V_{pmax}	V_{pmin}	V_{pmean}	$V_{pStd.Dev}$	$Upper\ bound_{max}$	$Upper\ bound_{min}$
V_{pmax}	1.0000	0.9520	0.9178	0.9821	-0.2235	0.7897
V_{pmin}	0.9520	1.0000	0.9944	0.8785	0.0233	0.5683
V_{pmean}	0.9178	0.9944	1.0000	0.8270	0.0726	0.4825
$V_{pStd.Dev}$	0.9821	0.8785	0.8270	1.0000	-0.3422	0.8909
$Upper\ bound_{max}$	-0.2235	0.0233	0.0726	-0.3422	1.0000	-0.5632
$Upper\ bound_{min}$	0.7897	0.5683	0.4825	0.8909	-0.5632	1.0000

Table 8: Correlation data matching to third objective (V_p) values and its upper bound values (minimum, maximum, mean, and standard deviation (Std. dev.) for Market 2 (DAX100)

Third objective variable	V_{pmax}	V_{pmin}	V_{pmean}	$V_{pStd.Dev}$	$Upper\ bound_{max}$	$Upper\ bound_{min}$
V_{pmax}	1.0000	0.9398	0.9756	0.9825	0.8915	-0.6095
V_{pmin}	0.9398	1.0000	0.9837	0.8657	0.8322	-0.3512
V_{pmean}	0.9756	0.9837	1.0000	0.9178	0.8216	-0.5131
$V_{pStd.Dev}$	0.9825	0.8657	0.9178	1.0000	0.9141	-0.6729
$Upper\ bound_{max}$	0.8915	0.8322	0.8216	0.9141	1.0000	-0.3482
$Upper\ bound_{min}$	-0.6095	-0.3512	-0.5131	-0.6729	-0.3482	1.0000

Table 9: Correlation data matching to third objective (V_p values and its upper bound values (minimum, maximum, mean, and standard deviation (Std. dev.) for Market 3 (FTSE100)

Third objective variable	V_{pmax}	V_{pmin}	V_{pmean}	$V_{pStd.Dev}$	$Upper\ bound_{max}$	$Upper\ bound_{min}$
V_{pmax}	1.0000	0.9600	0.9977	0.7436	-0.0301	0.4669
V_{pmin}	0.9600	1.0000	0.9438	0.5269	0.2448	0.5248
V_{pmean}	0.9977	0.9438	1.0000	0.7765	-0.0700	0.4893
$V_{pStd.Dev}$	0.7436	0.5269	0.7765	1.0000	-0.6713	0.1833
$Upper\ bound_{max}$	-0.0301	0.2448	-0.0700	-0.6713	1.0000	0.4266
$Upper\ bound_{min}$	0.4669	0.5248	0.4893	0.1833	0.4266	1.0000

Table 10: Correlation data matching to third objective (V_p) values and its upper bound values (minimum, maximum, mean, and standard deviation (Std. dev.) for Market 4 (S&P100)

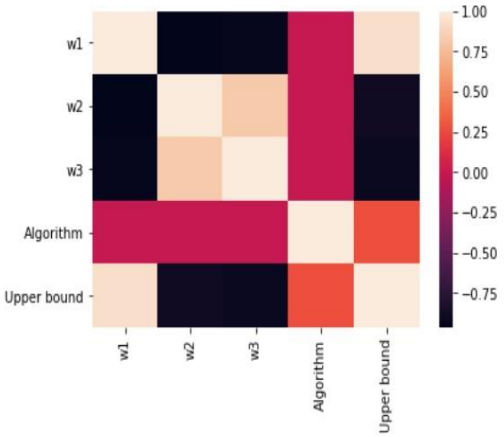
Third objective variable	V_{pmax}	V_{pmin}	V_{pmean}	$V_{pStd.Dev}$	$Upper\ bound_{max}$	$Upper\ bound_{min}$
V_{pmax}	1.0000	-0.6062	0.9914	0.9994	-0.8894	-0.0737
V_{pmin}	-0.6062	1.0000	-0.5811	-0.6323	0.8903	-0.6964
V_{pmean}	0.9914	-0.5811	1.0000	0.9908	-0.8858	-0.0510
$V_{pStd.Dev}$	0.9994	-0.6323	0.9908	1.0000	-0.9043	-0.0404
$Upper\ bound_{max}$	-0.8894	0.8903	-0.8858	-0.9043	1.0000	-0.3878
$Upper\ bound_{min}$	-0.0737	-0.6964	-0.0510	-0.0404	-0.3878	1.0000

Table 11. Correlation data matching to third objective (V_p) values and its upper bound values (minimum, maximum, mean, and standard deviation (Std. dev.) for Market 5 (Nikkei225)

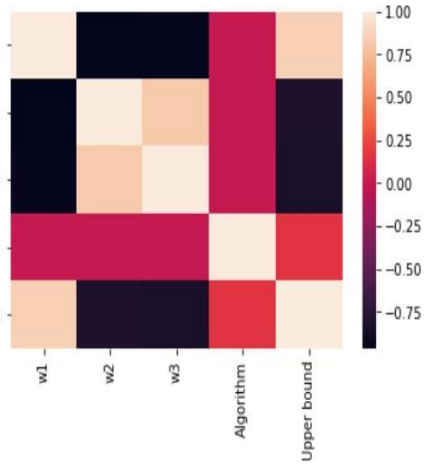
The heat-maps for five datasets of Market 1 - Market 5 are shown in Figures 9. The heat-maps provide a significant apprehension about the decision parameters and values of the upper bound of the third objective in the proposed model. It is observed that the values of the upper bound and weight (w_1) are positively correlated. Similarly, weight (w_1 and weight (w_3) are also correlated.

The upper bound values are dependent on the adapted algorithm to a small extent. Weight (w_1) is the corresponding weight for decision parameter a_i in the third objective equation used in the proposed framework. Similarly, w_2 and w_3 are the similar weights used for the decision parameters q_i and d_i, respectively, in the equation of the third objective.

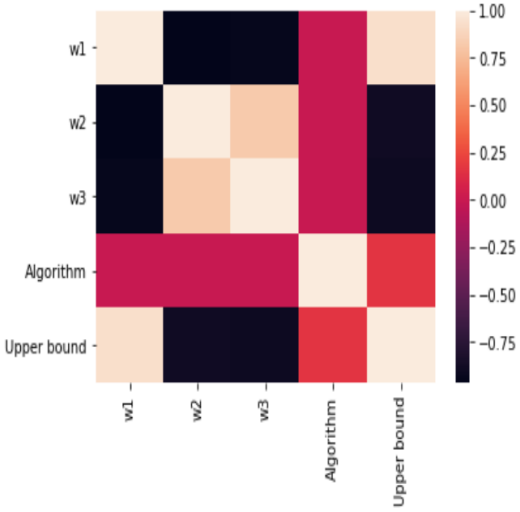
a. HS33



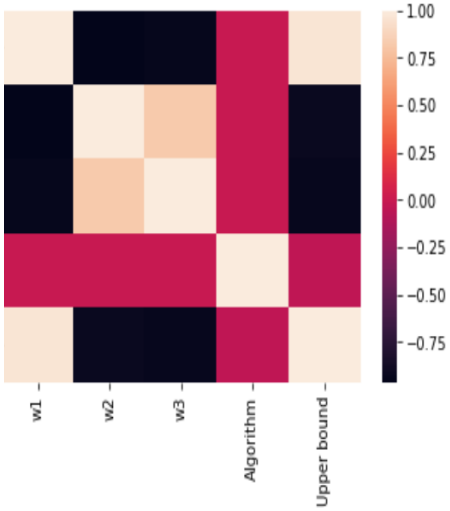
b. DAX100



c. FTSE100



d. S&P100



e. Nikkei225

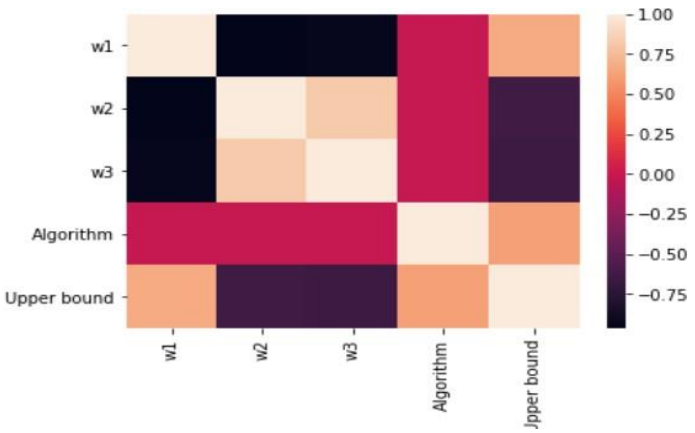


Figure 9: Correlation among weights of the decision parameters, adapted algorithm, and upper bound value of the third objective for five Markets.

6.6 Conclusions

A tri-objective portfolio optimization framework has been introduced in the chapter that is based on objectives of expected return, risk, and a composite index calculated using decision parameters. Realistic constraints like cardinality and pre-assignment are included in the framework. The use of MOEAs is needed in this approach since traditional methods are incapable of appropriately modeling this portfolio optimization framework. Generally, evolutionary algorithms face problems in suitably handling constraints. The proposed tri- objective framework uses the third objective in the form of a single index calculated by aggregating three significant decision parameters viz. a_i, q_i, and d_i. The decision parameter a_i indicates relative importance between expected return and variance, while the decision parameter q_i, indicates the proportion of the amount invested in a category of a portfolio and the decision parameter d_i is the downside risk level. The solution of the proposed framework is obtained using MOEA algorithms. This solution uses four MOEAs that are MOEA/D, NSGA-II, SPEA2, and MOPSO; these algorithms are used for finding efficient frontiers of the proposed tri-objective model. The realistic outputs are obtained by using these algorithms on five market datasets (HS33, DAX100, FTSE100, and Nikkei225). The third objective is a composite index calculated based on a weighted aggregation of the three decision parameters. The decision parameters d_i is calculated based on CVaR values in the model. A summary of the conclusion derived from this

research are given below:

1. The SPEA-2 outperforms other adapted algorithms for market 1 when the hypervolume metric is used as a performance metric. For the remaining markets datasets, NSGA-2 outperforms other algorithms in terms of hypervolume metric.
2. The upper bound values are highly correlated with values of decision parameter a_i for most of the cases. The lower value of the third objective is useful for obtaining the higher expected returns for a specified range of risk values.

The outputs indicate that a lower value of the third objective in the proposed model is preferred for achieving a higher expected return for a specified range of risk. The selection of an appropriate algorithm is dependent on the preferences of the user for either NSGA-II, SPEA2, MOPSO, or MOEA/D. The outputs obtained from these algorithms indicate that SPEA2 is a preferred choice for market 1 and NSGA-II is a preferred choice for the remaining markets. Finally, the inclusion of other significant parameters in the third objective is a fascinating direction for the extension of the proposed model, and this is regarded as the future scope of the study. Some directions for future scope of this study are:

I. The existing methods available in the literature can be utilized for initializing the weights in the composite index of the proposed model and calculate the sensitivity of decision parameters accordingly.

II. Additional realistic constraints such as Transaction cost constraint, Class constraints, and chance constraints may be included in the proposed model.

Furthermore, the selection of methods amidst adapted algorithms MOEA/D, SPEA2, NSGA-II, and MOPSO for the tri-objective model is based on the hypervolume metric and diversity of the solutions. NSGA-II is preferred choice of the algorithm based on the hypervolume metric and diversity of the solutions for the portfolio optimization problem. Since the investor is more inclined to a decision that yields a higher expected return, the choice also centers on choosing a lower value of the third objective for a given or specified range of risk. Thus, choosing a minimal value of the upper bound for the third objective value helps an investor make a decision.

Further, the value of the upper bound is highly correlated with the value of the decision parameter a_i, the choice for selecting a minimal value for the a_i parameter becomes a preferable option.

CHAPTER 7

Covid-19 Fake News Sentiment Analysis

7.1 Models Architecture a discruptive approach

In this Chapter, 39 features are created from text and these features are then used to detect the fake news regarding Covid-19 from social media using state-of-the-art deep learning models. The steps involved in propose methodology is shown in Figure 1. The dataset used in this study is collected from various website and social media cites. In order to make the dataset clean, some preprocessing are performed on dataset by removing URLs, some punctuations marks and empty columns. After text preprocesing, tokenization is performed to convert the larger text into words or in small lines.

The major part of this research is extracting the features from text and then using these features for fake news detection instead of text. After features extraction, these features are pass to state-of-the-art deep learning algorithms like RNN, LSTM and GRU to train the model While Various evaluation metrics are used to evaluate the performance of our proposed model.

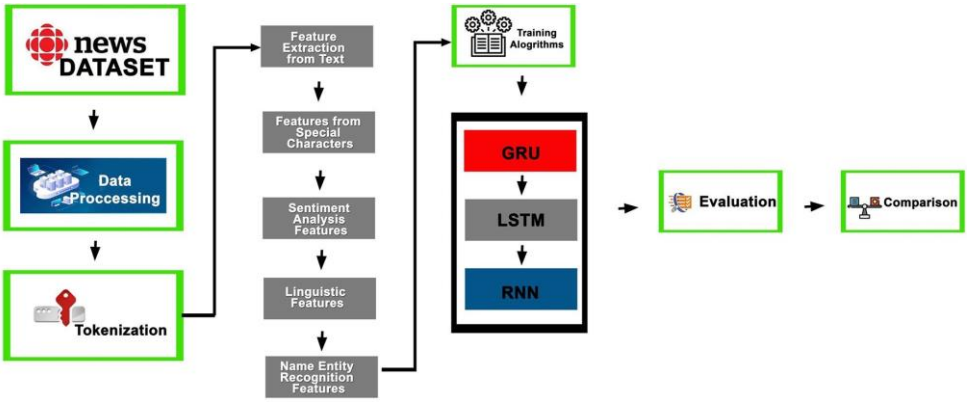


Figure 1: Proposed Model

7.1.1 Dataset Description

In this study, the covid19 dataset that is used was collected from [1]. Dataset consists of 586 true news and of which 578 are fake news. The dataset consists of

more then 1100 news articles and social media posts regrading covid19. The true news are gather from harvard Health Publishing, WHO, CDC, The New York Times and so on. The fake news ere gathered from social media (Facebook) posts and other medicine sites.

7.1.2 Feature Extraction from Text

The extraction of features is a reduction in dimensionality in which an original collection in raw data is condensed to more accessible classes for processing. By doing so we reduce the compiler's processing time and increase the rate of effectiveness in detecting the word value. A disadvantage of such broad data sets is that certain variables take a number of computing resources to handle. So for this purpose we have created our own features from text to detect fake news. These features have number of stop words, number of upper case letters in text, number of small letters in text, all upper case letters in text, number of numeric values in text, word count in text, character count in text, sentence count in text, aver- age sentence length in text, average word, length, sentence polarity score, sentiment scores, positive, negative, neutral and compound scores for text sentiments and also extracted name entity recognition features (NER) from text. Table 1, Table 2 and Table 3 represents the features that we created from the text and we used that features for classification.

7.1.3 Features from Special Characters

Features from special characters show the scenario that happens after the data collection. Our model indicates that every input vector represents a special character. For example, min-char-rnn utilizes one-hot vectors to represent different characters. Therefore, it allows our model to generate authentic training segments by predicting the next character based on previous characters. A special character allows for shorter text classification while integrating RNN, LSTM and GRU with the fully connected layers.

7.1.4 Sentiment Analysis Features

The fundamental task of applying the sentiment analysis features is the classification of polarity of our model text or sentence as Positive sentiment, negative sentiment, neutral sentiment or as compound sentiment. Here, our resultant text polarity of the data type in Table 1 is numeric. It actually represents

the features that we created from the text and we used that features for classification.

7.1.5 Linguistic Features

The classification performed in this research entails categorizing a piece of Fake news text into a category and translation. This is done by converting that piece of text into any other language. Table 2 shows the feature name arranged with their data types, displaying numeric and non-numeric characteristics.

Feature Name	Data Type
Positive Sentiment Score	Numeric
Negative Sentiment Score	Numeric
Neutral Sentiment Score	Numeric
Compound Sentiment Score	Numeric
Text Polarity	Numeric

Table 1: Sentiment Features from Text

7.1.6 Name Entity Recognition Features

This is also known as entity identification, entity chunk- and entity extraction. The objective of Name Entity Recognition (NER) is to allocate and classify tokens in texts into predefined categories. Table 3 shows the NER with the feature names and the data types.

7.1.7 Recurrent Neural Networks (RNN)

Due to the missing gradient problem, conventional neural networks (NN) do not yield satisfactory results on time series data. To deal with the above mentioned problem, RNN as proposed in 1982 by Hopfield. RNN has the strength of letting the NNs know the patterns over a period of time.
RNN will forecast the sequential data as behavior dependent on previous occurrences in a video, voice recordings, text things, etc. Figure 3 displays the RNN Working System.

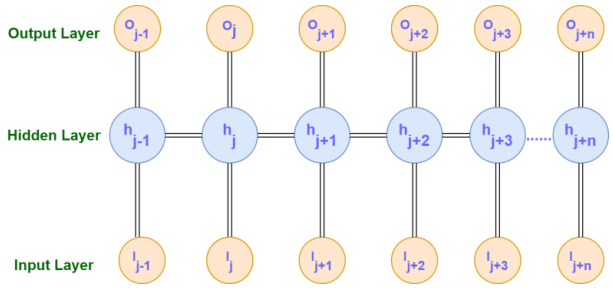


Figure 3: RNN Working Model

In Figure 4, h_j depicts the weight vector for hidden layer, O_j denotes the weight vector for output layer, I_j represents word vector which in input. The time stamp for hidden layer j is expressed using equation 1 and the Output layer final value for RNN is calculated using equation 2.

$$h_j = \sigma\left(M * O_j + N * h_{j-1}\right) \tag{1}$$

$$o_j = \sigma\left(K * h_j\right) \tag{2}$$

where $\sigma(\cdot)$ represent the activation function. Tanh, Relu sigmoid one of them can be used as activation function. After every timestamp j, the hidden state h_j is calculated by using equation 2, with the corresponding input parameters.

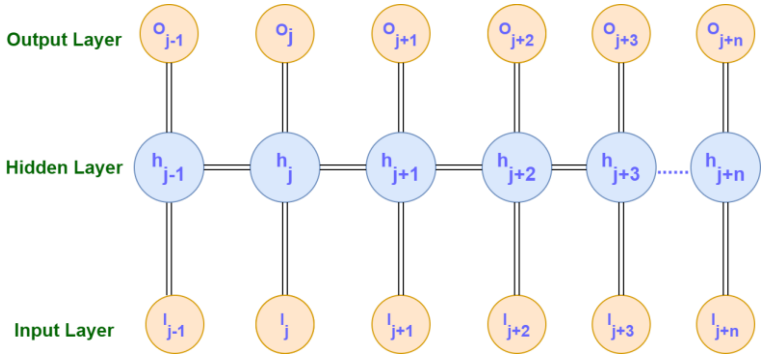


Figure 4: RNN Propose Structure

7.1.8 Long Short Term Memory

RNN suffers from short-term memory. If a series is long enough, they may find it

difficult to bring knowledge from earlier stages in time to later ones. So if you're trying to process a text paragraph to do predictions, RNN can leave crucial details out from the start. The RNN suffer from the vanishing gradient issue during back propagation. Gradients are the values used to update the weights of a neural network. The problem with the vanishing gradient is that the gradient shrinks as it propagates back over time. If a gradient value is exceedingly small, the learning does not help that much. LSTM was created as the short term memory solution. They have got internal systems called gates that can control information flow. These gates may know which data is necessary to retain or to throw away in a sequence. In doing so, it will transfer important knowledge down the long sequence chain to render predictions. With these two networks, almost all state-of-the-art results were achieved based on recurrent neural networks. LSTM consists of 3 gates such as input gate, forget gate and output gate. LSTM graphical representation is given in Figure 5. Sequence value D_i concatenated with the previous cell O_{i-1} output.

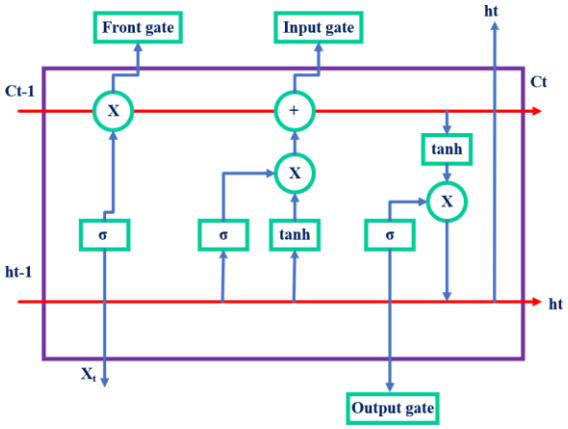


Figure 5: LSTM Working Mode

The first move for this combined input is to have it squashed through a **tano** layer. The second step is to transfer this input via an input gate.
A sigmoid function output range is in between 0 and 1, so the weights that bind the input to those nodes can be trained to output values close to zero to "delete" certain input values (or, conversely, outputs below 1 to "transfer" certain values).

Feature Name	Data Type	Feature Name	Data Type
News Source	Non-Numeric	Num of ?	Numeric
Num of Stopwords	Numeric	Num of /	Numeric
Num of @	Numeric	Num of #	Numeric
Num of numeric values	Numeric	Num of upercase characters	Numeric
Num of lowercase characters	Numeric	Num of a1l upercase characters	Numeric
Text language	Numeric	Word count	Numeric
Character count	Numeric	Sentence count	Numeric
Average word length	Numeric	Average sentence length	Numeric

Table 2: Linguistic Features from Text

Feature Name	Data Type	Feature Name	Data Type
Person	Numeric	NORP	Numeric
FAC	Numeric	Organization	Numeric
GPE	Numeric	Location	Numeric
Product	Numeric	Event	Numeric
Work of Art	Numeric	Law	Numeric
Language	Numeric	Date	Numeric
Time	Numeric	Percent	Numeric
Money	Numeric	Quantity	Numeric
Cardinal	Numeric	Ordinal	Numeric

Table 3: NER Features from Text

Layer (type)	Output Shape	Param #
lstm_13 (LSTM)	(None, None, 100)	56400
lstm_14 (LSTM)	(None, None, 50)	30200
lstm_15 (LSTM)	(None, 25)	7600
dropout_17 (Dropout)	(None, 25)	0
dense_21 (Dense)	(None, 10)	260
dense_22 (Dense)	(None, 10)	110
dense_23 (Dense)	(None, 1)	11
Total params: 94,581 Trainable params: 94,581 Non-trainable params: 0		

Table 4: LSTM Propose Tabulation

The next step is data flow through forget gate loop. LSTM cell consist of an internal state variable e_i. This variable, updated after one time phase i.e. e_{i-1} is applied to the input data in order to establish an efficient recurrence layer. Lastly, we have an output layer **tano** squashing function, whose output is controlled by an output gate. This gate decides what values are technically allowed as cell O_i output. LSTM cell mathematics looks like this: Input Next, inputs are squashed between -1 and 1 using a **tano** activation function. As shown in Table 4.
That can be articulated through equation,

$$g = (b^g + D_i W^g + O_{i-1} U^g) \tag{3}$$

b^g represents input bias, while W^g is the weight for input and U^g is the weight of previous output cell. Exponents g do not reflect an increased power, but rather imply that these are input weights and bias values. Such squashed input is then element-wise multiplied by the output of input gate, which is already mentioned above, is a set of activated sigmoid nodes:

$$I = \sigma(b^I + D_i W^I + O_{i-1} U^I) \tag{4}$$

Then, the output of the LSTM cell input section is mentioned in equation 3.

$$g \circ I \tag{5}$$

The ∘ operator depict element wise multiplication. Mathematical form for forget gate is shown in equation 4.

$$f = \sigma(b^f + D_i W^f + O_{i-1} U^f) \tag{6}$$

The output of the previous state's element-wise product and the forget gate shall be represented as

$$e_{i-1} \circ f \tag{7}$$

Output from forget gate is shown in equation 6.

$$s_t = s_{t-1} \circ f + g \circ i \tag{8}$$

Mathematical form for output gate is show in equation 7.

$$e_i = e_{i-1} \circ f + g \circ I \tag{9}$$

$$R = \sigma\left(b^R + D_i W^R + O_{i-1} U^r\right) \tag{10}$$

So the cell's final output can be seen, with the **tano** squashing, as:

7.1.9 Gated Recurrent Unit (GRU)

Many variants were built to solve the problem of the Vanishing-Exploding gradients often encountered during the development of a simple Recurrent Neural Network. One of the most popular variants is the LSTM.
The GRU is one of the lesser-known but equally powerful variants. GRU, unlike LSTM only consists of just three gates and does not retain an Internal Cell state.
Information that is contained in an LSTM recurrent unit in the Internal Cell State is incorporated into the Gated Recurrent Unit's hidden state. The cumulative information would be moved on to the next Recurrent Gated Unit. GRU with various gates functions are as outlined below:

- **Update Gate:** it dictates how much of historical information has to be transferred into the future. It is similar to the Input Gate in a recurrent cell of the LSTM.

- **Reset gate:** It determines how much past knowledge you should forget about. It is identical to combining the Input Gate and the Forget Gate in a recurrent LSTM unit.

$$U^j = \sigma\left(W_{U^J} X^j + Z_{U^J} h_{j-1} + b_{U^j}\right) \tag{11}$$

$$R^j = \sigma\left(W_{R^J} X^j + Z_{R^J} h_{j-1} + b_{R^j}\right) \tag{12}$$

Updated gate is represented by U^j , sigmoid function is depicted by $\sigma(\cdot)$. Similarly W, Z and b h_j are matrices and vector parameters. represents output vector, input vector is represented by Xj. The upgrade gate executes identical forget gate and an LSTM input gate functions.
This is the duty of choosing which information to remove and which information to add. It is the duty of the reset gate to decide how many of the previous data will be forgotten. Because GRU has less gates relative to LSTM, the cycle of training is typically quicker.

- **Current Memory Gate**: This is often overlooked during a typical Gated Recurrent Unit network discussion. This is integrated into the Reset Gate even like the Input Modulation Gate is a subpart of the Input Gate that is used to add a certain non-linearity in the data and even to render the Zero-mean response.
Another justification to make it a sub-part of the Reset gate is to which the impact of the new information being transferred into the future that prior information has been. Current Memory Gate calculation process is a bit different. F
irstly, the Reset Gate Hadmard product and the previous hidden state vector are calculated.
This function is then parameterised and then applied to the input vector of the parameterized current as shown in Table 5.

$$\bar{h}_j = tanh\ (W \odot I_j + W \odot (R_j \odot h_{j-1})) \tag{13}$$

First a vector of the ones and the same dimensions as that of the input is defined to calculate the current hidden state. This vector is name as ones and it is represented as 1 mathematically.
First measure the update gate's hadmard element and past hidden state function. Subtract the upgrade gate from ones to create a new vector and then, measure the newly created vector hadmard product using the current memory gate.

$$h_j = \ (U_j \odot h_{j-1} + (1 - U_j) \odot \bar{h}_j) \tag{14}$$

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, None, 100)	42300
gru_2 (GRU)	(None, None, 50)	22650
dropout_1 (Dropout)	(None, None, 50)	0
gru_3 (GRU)	(None, 25)	5700
dropout_2 (Dropout)	(None, 25)	0
dense_1 (Dense)	(None, 10)	260
dense_2 (Dense)	(None, 1)	11
Total params: 70,921 Trainable params: 70,921 Non-trainable params: 0		

Table 5: GRU Propose Tabulation

The back-propagation for a GRU network through time algorithm is similar to that of a LSTM network and differs only in the formation of the differential chain. Let $\bar{y}_j$be the expected performance at every stage of the period, and y_j be the actual performance at every step. The error is then given on each step of the time by:

$$E_j = -\, y_j \; log\big(\bar{y}_j\big) \tag{15}$$

Therefore the cumulative error is given at all time phases by summing up the errors.

$$E = \sum_j E_j \tag{16}$$

$$E = \sum_j -y_j log\,\left(\bar{y}_j\right) \tag{17}$$

Likewise, on every time stage, the value $\frac{\partial E}{\partial W}$ can be determined as the summation of the gradients.

$$\frac{\partial E}{\partial W} = \sum_j \frac{\partial E_j}{\partial W} \tag{18}$$

Using the rule of the chain and the fact that $\bar{y}_j$ is a feature of h_j and which is also a feature of $\bar{h}_j$ the following statement arises:

$$\frac{\partial E_j}{\partial W} = \frac{\partial E_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial h_j} \frac{\partial h_j}{\partial h_{j-1}} \frac{\partial h_{j-1}}{\partial h_{j-2}} \dots \dots \frac{\partial h_0}{\partial W} \tag{19}$$

Therefore the complete gradient of the error is supplied by:

$$\frac{\partial E}{\partial W} = \sum_j \frac{\partial E_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial h_j} \frac{\partial h_j}{\partial h_{j-1}} \frac{\partial h_{j-1}}{\partial h_{j-2}} \dots \dots \frac{\partial h_0}{\partial W} \tag{20}$$

Remember that the gradient equation requires a ∂h_j chain that appears similar to a simple Recurrent Neural Network but this method functions differently owing to the internal workings of the h_j derivatives.

7.2 Experimental Set-Up

In order to make the dataset clean, preprocessing is performed on the dataset by removing URLs, some punctuations marks and empty columns. After cleaning the text, we extracted features from text to create new features for predication. These features have a number of stop words, uppercase letters in text, small letters in text, numeric values in text, word count in text, character count in text, sentence count in text, average sentence length in text, average word, length, sentence polarity score, sentiment scores, positive, negative, neutral and compound scores for text sentiments and also extracts name entity recognition features (NER) from text. More details of these features are mentioned in Section 7.4. After extracting features from text, these features are normalized using min-max normalization, the mathematical expression for min-max normalization is given below:

$$f' = \frac{f - \min(N)}{\max(N) - \min(N)} (\text{new\_max(N)} - \text{new\_min(N)}) + \text{new\_min(N)} \tag{21}$$

In equation 21 'f' represents the current values of features while '$min(N)$' and '$max(N)$' represents the smallest and largest values features in given column. While $new\_max(N)$ and $new\_min(N)$ are the largest and smallest scales to be set for normalization.
After data normalization, data encoding was also per- formed to convert non-numeric values to numeric values, before moving data for training.

7.2.1 Data Visualization

Figure 6 and Figure 7 represents the word-cloud of most frequent words, similarly, Figure 8 represents the frequency bar-graph for frequent words.

From Figure 9, we can see that covid, coronavirus, virus and people words are most frequently used in the dataset. Health, disease, symptom, words frequency are more than 250. Infected, person, vaccine, spread and patient's words frequency are between 200 to 230.

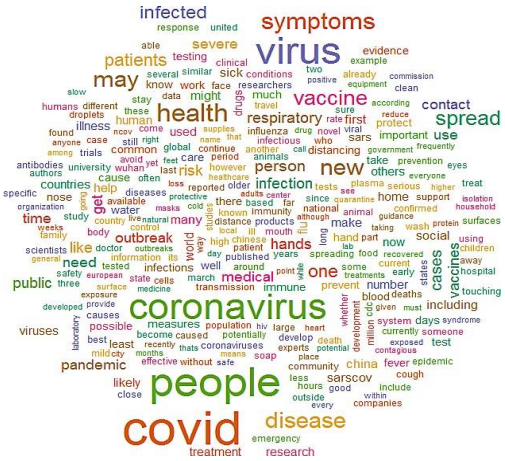


Figure 6: Real News Word Cloud

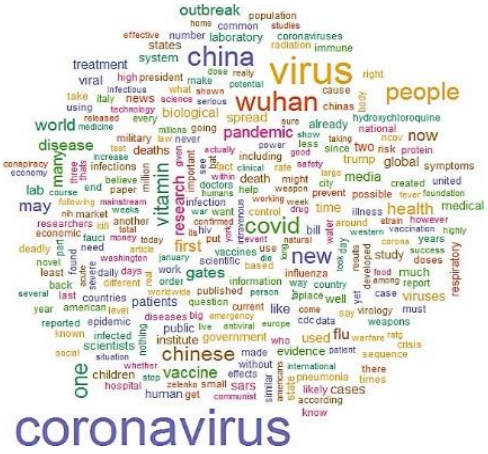


Figure 7: Fake News Word Cloud

Figure 10, Figure 11 depict the most common words used corona virus fake news. We can see from Figure 12, that corona-virus and virus are most frequently used words and these words are used more than 450 times in the text.

China, Wuhan, world, pandemic, people these words are also used frequently in the fake news text.

Figure 10 represents the emotion graph for fake news and real news.

Figure 11 and 12 represents the sources for real and fake news respectively.

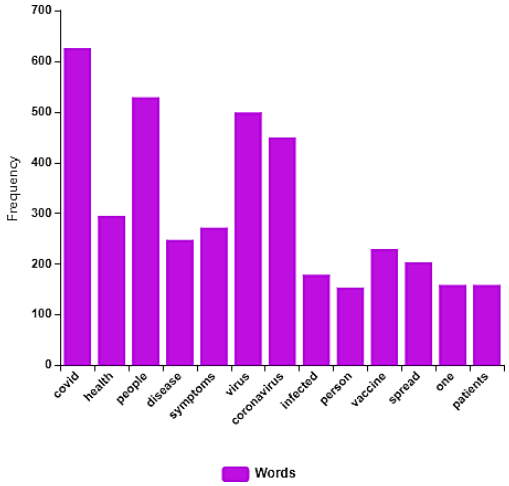


Figure 8: Frequency Bargraph for Real News in Covid-19

7.2.2 Tokenization

Tokenization relates to breaking a wider body of text into smaller sections, phrases or even terms for a language other than English. The various tokenization features are built into the nltk framework itself. We used Regex Tokenizer, which can either extract tokens by using the regex pattern given to break the text (default) Or match the regex repeatedly (if the gaps are false).

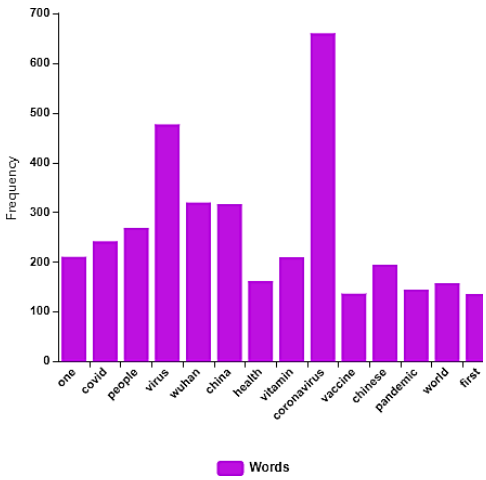


Figure 9: Frequency Bargraph for Fake News in Covid19

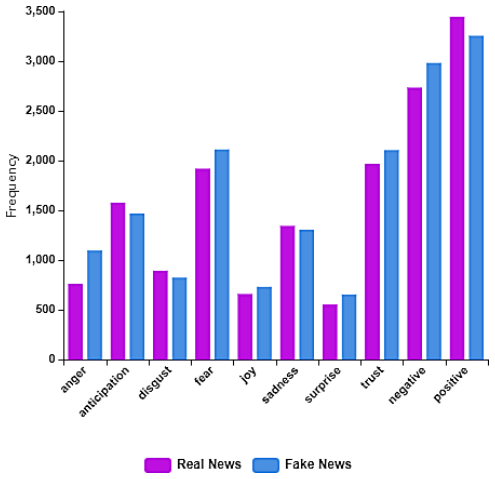


Figure 10: Emotion Mining For Covid-19 News

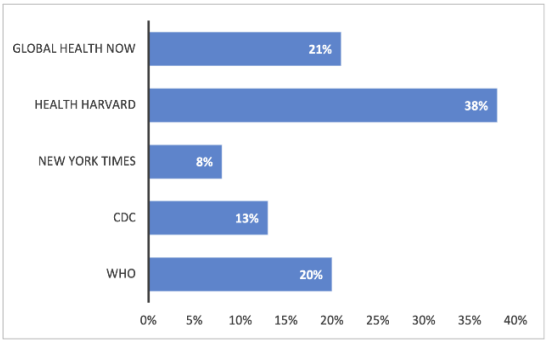


Figure 11: Real News Sources for Covid-19

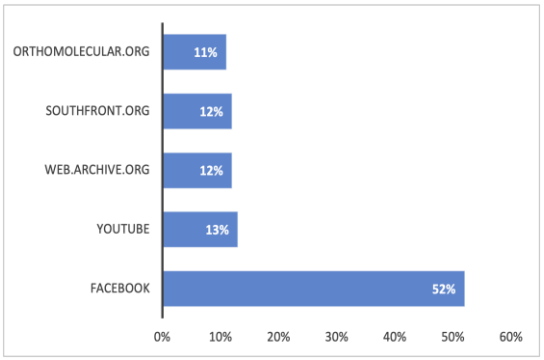


Figure 12: Fake News Sources for Covid-19

7.2.3 Evaluation metrics

Various performance metrics are used to evaluate the pro- posed solution, including precision, recall, F1-Measure and accuracy. Above mentioned performance metrics are based on Actual Positive (AP), False Positive (FP), False Negative (FN) and Actual Negative (AN).

Accuracy is used to measure how many instances are correctly classified as normal and attacks classes. Accuracy is achieved by summing correctly classify instances with dividing the total instances represented in equation 13.

$$Accurancy = \frac{AP + AN}{AP + FP + FN + AN} \tag{22}$$

$$DR = \frac{AP}{AP + FN} \tag{23}$$

Precision's objective is to evaluate the True Positive (TP) entities in relation to False Positive (FP) entities.

$$Precision = \frac{AP}{AP + FP} \tag{24}$$

The purpose of recall is to evaluate True Positive (TP) entities in relation to (FN) False Negative entities that are not at categorized. The mathematical form of recall is mentioned in the equation below

$$Recall = \frac{AP}{AP + FN} \tag{25}$$

Sometimes performance assessment may not be good with accuracy and recall, for instance, if one mining algorithm has low recall but high precision that another algorithm is needed. Then there is the question of which algorithm is better. This problem is solved by using F1-score that gives an average recall and precision. F1-score can be calculated as shown in equation.

$$F1 - score = \frac{2 \ * Precision \ * Recall}{Precision + Recall} \tag{26}$$

Algorithm	Predication Accuracy	Precision	Recall	F1-Measure
AdaBoost Classifier	79.88	76.76	86.36	81.82
Decision Tree Classifier	67.81	70.51	62.50	66.26
KNeighbors Classifier	62.06	72.91	39.77	51.47

Table 6: Performance of the ML algorithms before Feature Extraction

Classifier	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
GRU	0.05	98.29	4.11	59.20
LSTM	0.31	95.07	6.90	55.72
RNN	0.00	100.00	3.30	57.71

Table 7: Accuracy and Loss of all the classifiers

Model	Label	Precision	Recall	F1-Measure
GRU	Fake News	55	58	56
	Real News	63	60	62
LSTM	Fake News	51	60	55
	Real News	61	52	56
RNN	Fake News	53	62	57
	Real News	63	55	59

Table 8: Classification Report Before features

7.3 Results and Discussion

All these experiments are performed on google colab. System specification core I3 system with 8 GB RAM and GHz processor is used.

7.3.1 Results Without Features Extraction

From table 6 we can depict that Ada-boost classifier out- performed compare to other machine learning algorithms like Decision Tree (DT) and K Nearest Neighbour (KNN) in terms of accuracy, precision, recall and F1-Measure score 79.88 % predication accuracy is achieved by Ada-boost before feature extraction which is highest among all the machine learning classifiers. Similarly, Ada-boost also achieved 76.76% precision, 86.36% recall and 81.82% F1-Measure score respectively. Predication accuracy for DT and KNN is 67.81% and 62.06% respectively. Precision recall and F1-Measure for DT are 70.51%, 62.50% and 66.26% respectively. KNN achieved 72.91% precision, 39.77% recall and 51.47% F1-Measure respectively. We can see that the traditional machine learning algorithms performed very well compared to our proposed deep learning models before creating our own features as stipulated in section 7.3. Therefore, after the creation of our novel features from text, our proposed model is expected to outperform the machine learning models.

From Table 5 and Table 6 it can be observed that the Precision, Recall and F1-Score for GRU model without feature extraction are 55%, 58%, and 56% for fake news, 63%, 60% and 62% respectively for Real News using GRU model.

Similarly, training loss for GRU model is 0.05 and training accuracy is 98.29% respectively. This is typically expected for a testing accuracy to be lower than the training accuracy due to the refinement of the data. Also, predication loss and predication accuracy scores are 4.11 and 59.20%. Using LSTM model, we have achieved 0.31 training loss and 6.90 testing loss. Training and testing accuracy for LSTM model are 95.07% and 55.72% respectively. Precision, recall and F1-Measure for fake news using LSTM model are 51%, 60% and 55% respectively. Similarly, for real news precision, recall and F1-Measure scores are 61%, 52% and 56% respectively. For RNN model 0.00 training loss and 3.30 testing loss is recorded respectively. Training and predication accuracy for RNN model are 100% and 57.71% respectively. Overall RNN model achieved 0.00 training loss and training accuracy 100% which is better compared to GRU and LSTM training loss and training accuracy. Similarly, RNN achieved low loss for predication which is 3.30 and GRU achieved high predication accuracy 59.20% compared to LSTM and RNN. For fake news overall, high precision is 55% using GRU model. Best recall and F1-Measure for fake news recorded are 62% and 57% using RNN model respectively. Optimal precision, recall and F1-Measure for real news are 63%, 60% and 62% using GRU mode. Training accuracy is high because 70%

dataset is used for training and only 30% dataset is used for testing. It also shows that deep learning algorithms performs well on large features and large dataset.
Figure 13 represents the predication accuracy and training accuracy for GRU model. The green curve depicts the training accuracy. At 1st epoch, training accuracy is 45% and after 100th epoch, we achieved 98.29% training accuracy which is highest training accuracy. Similarly, at 1st epoch, predication accuracy using GRU model is 47% and after 100th epoch, maximum predication accuracy reported is 59.20%.

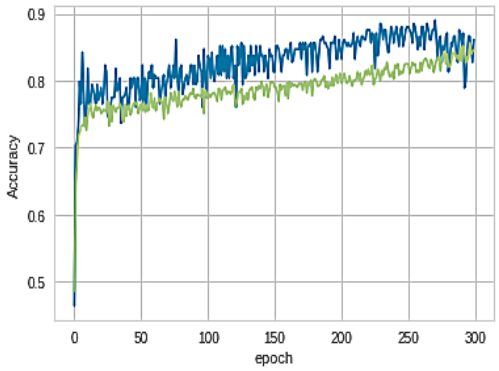


Figure 13: GRU Model Training and Testing Accuracy

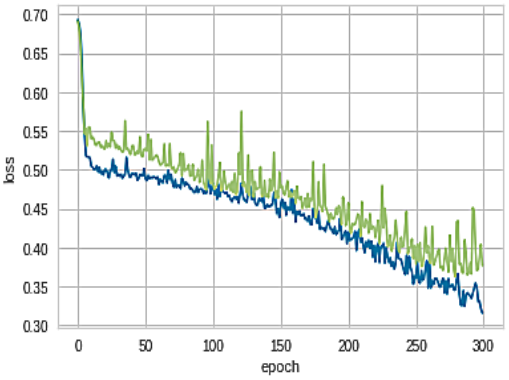


Figure 14: GRU Model Training and Testing Loss

Figure 14 represents the GRU model loss for training and testing. The blue curve depicts the training loss and the green curve represents the testing loss this time. At 1st iteration, training loss is around 0.70 and after 10th iteration is 0.05. Similarly, for predication loss at 1st epoch, predication loss is 0.5 and after 100th epoch, predication loss is 4.11 respectively using GRU model.

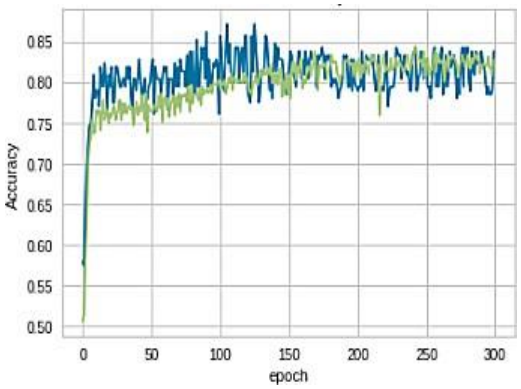


Figure 15: LSTM Model Training and Testing Accuracy

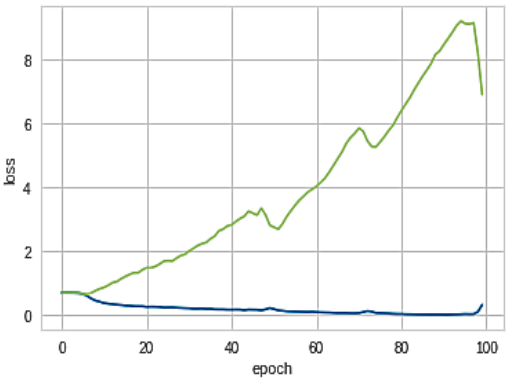


Figure 16: LSTM Model Training and Testing Loss

Figure 15 represents the predication accuracy and training accuracy for LSTM model. The green curve depicts the training accuracy, at 1st epoch training accuracy is 47% and after 100th epoch, we achieved 95.07% training accuracy. Similarly, at 1st epoch, predication accuracy using LSTM model is 56% and after 100th epoch, maximum predication accuracy reported is 55.72%.
Figure 16 represents the LSTM model loss for training and testing. The blue curve depict the training loss and green curve represent the testing loss this time. At 1st epoch training loss is around 0.68 and after 100th iteration 0.00. Similarly, for predication loss at 1st epoch, predication loss is 0.67 and after 100th epoch, predication loss is 3.30 respectively using RNN model.

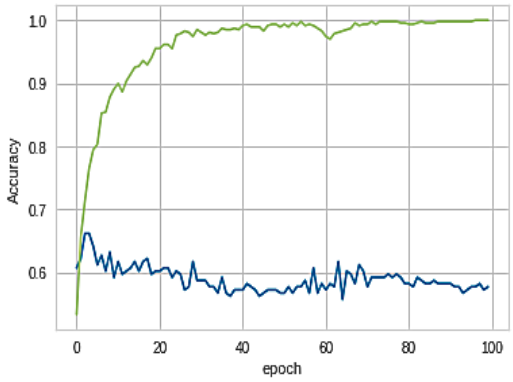


Figure 17: RNN Model Training and Testing Accuracy

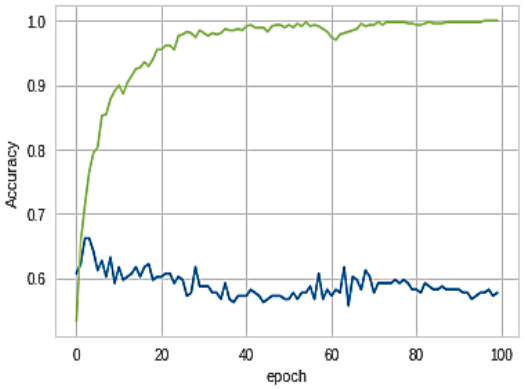


Figure 18: RNN Model Training and Testing Loss

Figure 17 represents the predication accuracy and training accuracy for RNN model. The blue curve depict the training accuracy, at 1st epoch training accuracy is 53.32% and after 100th epoch, we achieved 100% training accuracy which is highest training accuracy.
Similarly, at 1st epoch, predication accuracy using LSTM model is 60.70% and after 10th epoch, maximum predication accuracy reported is 57.71%.
Figure 18 represents the RNN model loss for training and testing. At 1st epoch training loss is around 0.70 and after 10th iteration 0.58.
Similarly, for predication loss at 1st epoch predication loss is 0.72 and after 10th epoch predication loss is 0.68 respectively using GRU model.

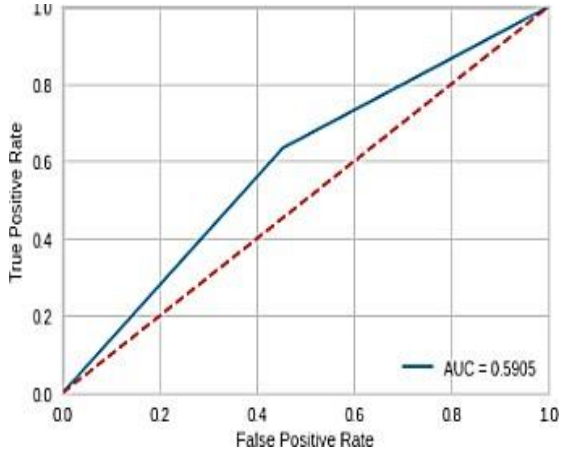


Figure 19: GRU ROC Curve

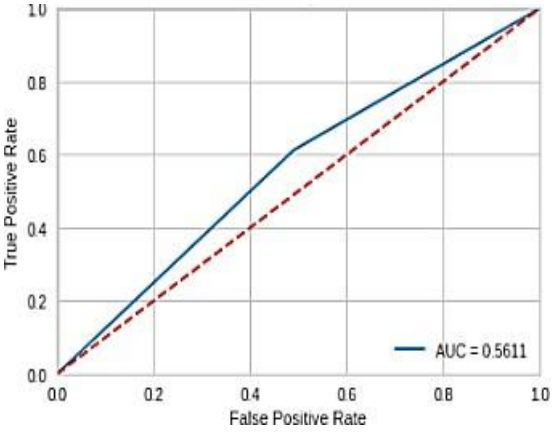


Figure 20: LSTM ROC Curve

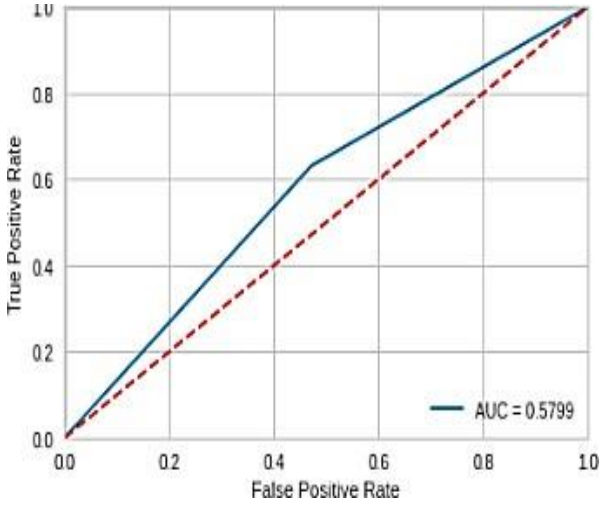


Figure 21: RNN ROC Curve

Figure 19 represents AUC curve for GRU classifier and UC for GRU is around 59.05%. Similarly figure 20 represents AUC curve for LSTM and figure 21 depicts AUC for RNN model. LSTM and RNN both have 56.11% and 57.99% AUC respectively.

GRU outperformed in terms of AUC compared to other classifiers.

Algorithm	Predication Accuracy	Precision	Recall	F1 Score
AdaBoost	82.75	79.00	89.77	84.04
Decision Tree	77.58	72.47	89.77	80.20
K Neighbors	69.54	62.41	100	76.85

Table 9: Performance of the ML algorithms after Feature Extraction

7.3.2 Results After Features Extraction

Table 9 depicts the that Ada-boost classifier achieved 82.75% predication accuracy, while DT and KNN achieved 77.58% and 69.54% predication accuracy respectively. Similarly, precision, recall and F1-Measure scores are 79%, 89.77% and 84.04% respectively. 72.47%, 89.77% and 80.20% precision, recall and F1-Measure scores are achieved using DT respectively. KNN classifier achieved 62.41% precision, 100% recall and 76.85% F1-Measure score respectively.
After features extraction on both DL and ML, we can deduce that deep learning out performed machine learning algorithms in terms of precision, recall, F1-Measure and Predication accuracy. Table 7 and 8 shows the Accuracy and Loss of all the classifiers and Classification Report Before features.
Table 10 shows Accuracy and Loss of all the classifiers an Table 11 shows Classification Report Before features.

Classifier	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
GRU	0.31	84.57	0.37	86.12
LSTM	0.38	82.72	0.45	83.73
RNN	0.31	84.57	0.34	85.65

Table 10: Accuracy and Loss of all the classifiers

Model	Label	Precision	Recall	F1-Measure
GRU	Fake News	84	83	83
	Real News	88	88	88
LSTM	Fake News	84	76	80
	Real News	84	89	86
RNN	Fake News	85	80	82
	Real News	86	90	88

Table 11: Classification Report Before features

Table 9 depicts that the Precision, Recall and F1-Score GRU model after feature extraction are 84%, 83%, and 83% for fake news, 88%, 88% and 88% respectively for Real News using GRU model. Similarly, training loss for GRU model is 0.31, training accuracy is 84.57% respectively. Predication loss and predication accuracy scores are 0.37 and 86.12%. Using LSTM model, we have achieved 0.38 training loss and 0.45 testing loss. Training and testing accuracy for LSTM model are 82.72% and 83.73% respectively.

Precision, recall and F1-Measure for fake news using LSTM model are 84%,76% and 80% respectively. Similarly, for real news precision, recall and F1-Measure scores are 84%, 89% and 86% respectively. For RNN model 0.31 training loss and 0.34 testing loss is recorded respectively. Training and predication accuracy for RNN model are 84.57% and 85.65% respectively.

Overall GRU and RNN model achieved 0.31 training loss and training accuracy 84.57% which is better compared to LSTM training loss and training accuracy, respectively. Similarly, RNN achieved low loss for predication which is 0.34 and GRU achieved high predication accuracy of 86.12% compared to LSTM and RNN.

For fake news high precision is 85% using RNN model.

Best recall and F1- Measure for fake news recorded are 83% using GRU model. 88% optimal precision is recorded using GRU for real news. RNN model achieved 90% recall for real news while 88% F1-measure is recorded using GRU and RNN.

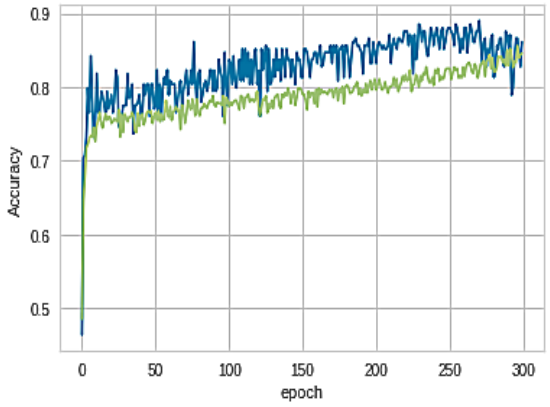


Figure 22: GRU Model Training and Testing Accuracy

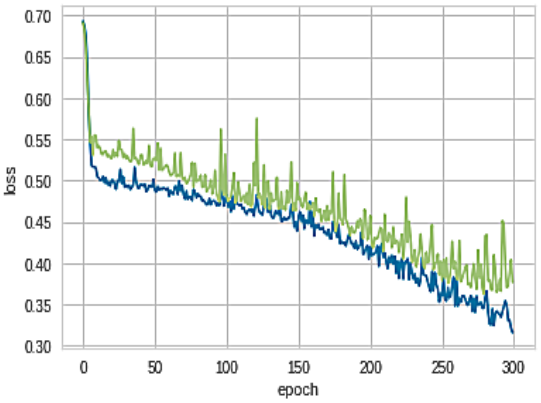


Figure 23: GRU Model Training and Testing Loss

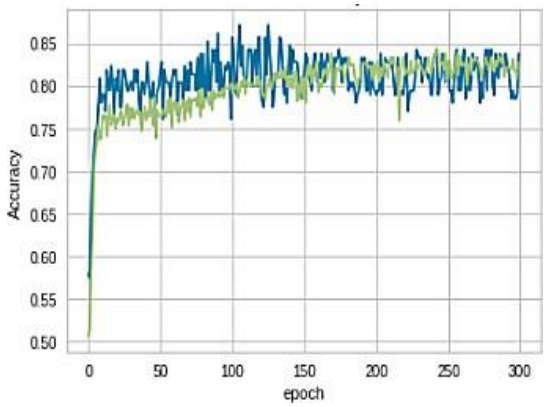


Figure 24: LSTM Model Training and Testing Accuracy

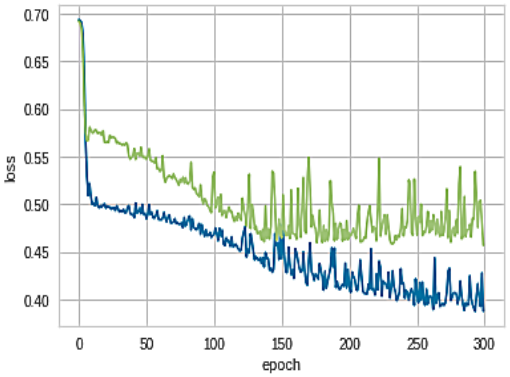


Figure 25: LSTM Model Training and Testing Loss

Figure 22 represents the predication accuracy and training accuracy for GRU model. The green curve depicts the training accuracy. At 1st epoch training, accuracy is 45% and after 100th epoch, we achieved 84.57% training accuracy which is highest training accuracy.

Similarly, at 1st epoch, predication accuracy using GRU model is 44% and after 100th epoch, maximum predication accuracy reported is 86.12%. Figure 23 represents the GRU model loss for training and testing. The blue curve depicts the training loss and the green curve represents the testing loss this time. At 1st epoch, training loss is around 0.55 and after 100th iteration is 0.31. Similarly, for predication loss at 1st epoch, predication loss is 0.70 and after 100th epoch, predication loss is 0.37 respectively using GRU model.

Figure 24 represents the predication accuracy and train-accuracy for LSTM model. The green curve depicts training accuracy. At 1st epoch, training accuracy is 52% and after 100th epoch, we achieved 82.72% training ac- curacy which is highest training accuracy. Similarly, at 1st epoch, predication accuracy using GRU model is 54% and after 100th epoch, maximum predication accuracy reported is 83.73%. Figure 25 represents the LSTM model loss for training and testing. The blue curve depicts the training loss and the green curve represents the testing loss this time. At 1st epoch, training loss is around 0.68 and after 100th iteration is 0.38. Similarly, for predication loss at 1st epoch, predication loss is 0.67 and after 100th epoch, predication loss is 0.45 respectively using LSTM model.

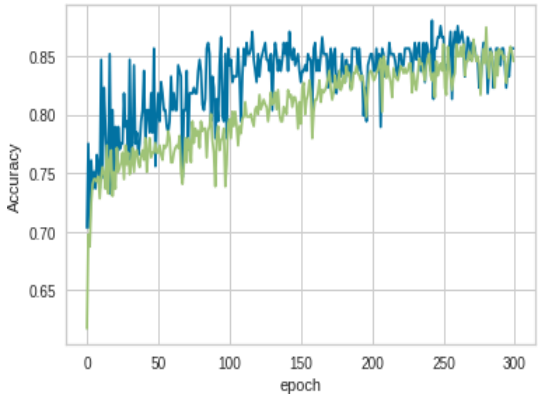


Figure 26: RNN Model Training and Testing Accuracy

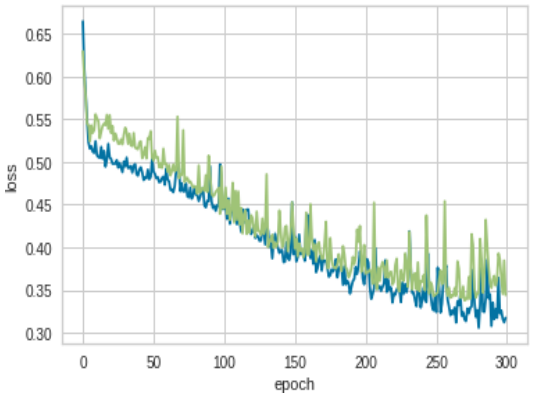


Figure 27: RNN Model Training and Testing

Figure 26 represents the predication accuracy and training accuracy for RNN model. The green curve depicts the training accuracy. At 1st epoch, training accuracy is 61% and after 100th epoch, we achieved 84.57% training accuracy which is highest training accuracy. Similarly, at 1st epoch, predication accuracy using RNN model is 71% and after 100th epoch, maximum predication accuracy reported is 85.65%. Figure 27 represents the RNN model loss for training and testing.

The blue curve depicts the training loss and the green curve represents the testing loss this time.

At 1st epoch, training loss is around 0.68 and after 100th iteration is 0.31. Similarly, for predication loss at 1st epoch, predication loss is 0.65 and after 100th epoch, predication loss is 0.34 respectively using RNN model.

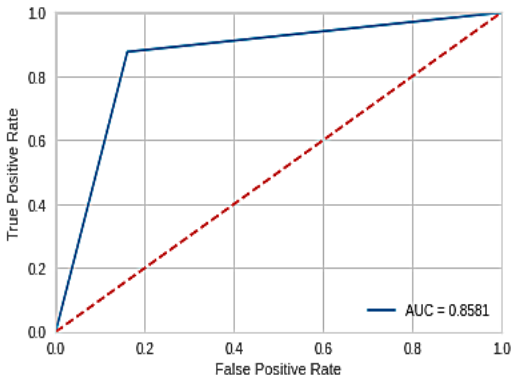


Figure 28: GRU ROC Curve

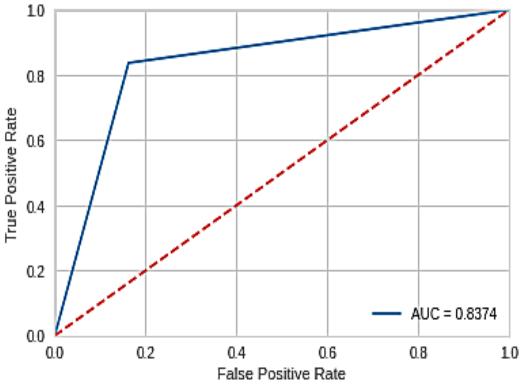


Figure 29: LSTM ROC Curve

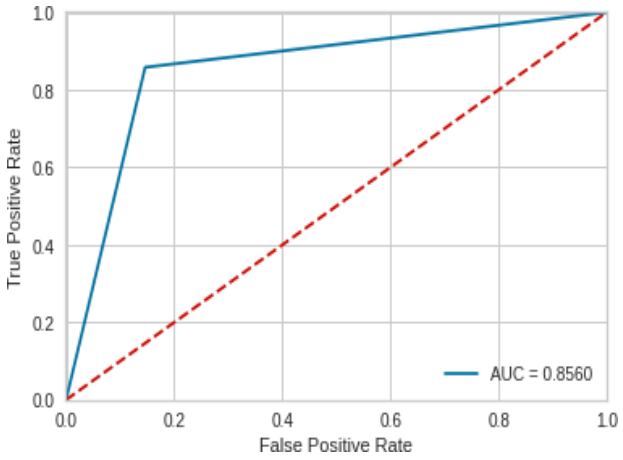


Figure 30: RNN ROC Curve

Figures 28, 29 and 30 depicts that GRU has highest Area under the curve (AUC) which is 85.81% while LSTM and RNN both have 83.74% and 85.60%. Comparison with before feature extraction AUC after feature extraction is much higher, with a 20% increase in AUC after feature extraction.

7.4 Conclusion

At a time where the use of masks and adherence to social distancing policy due to the Covid-19 pandemic has a pro- found power to slow down or even eliminate the coronavirus outbreak, many people are still adamant and flout the policy. Why has it been so hard to get people to adopt these simple measures? There are many reasons, but this research has identified misinformation on the internet as a major factor. Consequently, people are quite unsure and struggle to understand the exponential spread of the virus. The outbreak of the virus was paralleled by an outbreak of misinformation about the virus. This misinformation ranges from false origin of the virus, conspiracy theories, fake cures to harmful health advice. Misinformation jeopardises public health responses. This research presented a novel system utilizing 39 features used in detecting fake news about Covid-19. The model uses information fusion process to obtain social media data and applies state-of-the-art deep learning models such as GRU, LSTM and RNN. We have proposed sentiment features, linguistic features, and name entity-based features. After extracting features from text, our new features detect fake news in Covid-19 with the accuracy of 86.12%. Thus, accuracy is increased by 20%. Overall high precision is 85% using RNN model. The results further show the best recall and F1-Measure for fake news recorded to be 83% using GRU model. Similarly, for real news precision, recall and F1-Measure for real news are 88%, 90% and 88% using the GRU, RNN and LSTM models, respectively. Our results were compared

with the standard machine learning algorithms with same dataset and confirms the superiority of our deep learning model. In the future, we shall optimize our idea to include automation with instant review and comment of users justifying their news or totally removing them from the communication chain.

CHAPTER 8

A Hybrid Learning Approach for the Stage-wise Classification and Prediction of Covid-19 X-ray Images

8.1 Materials and methods

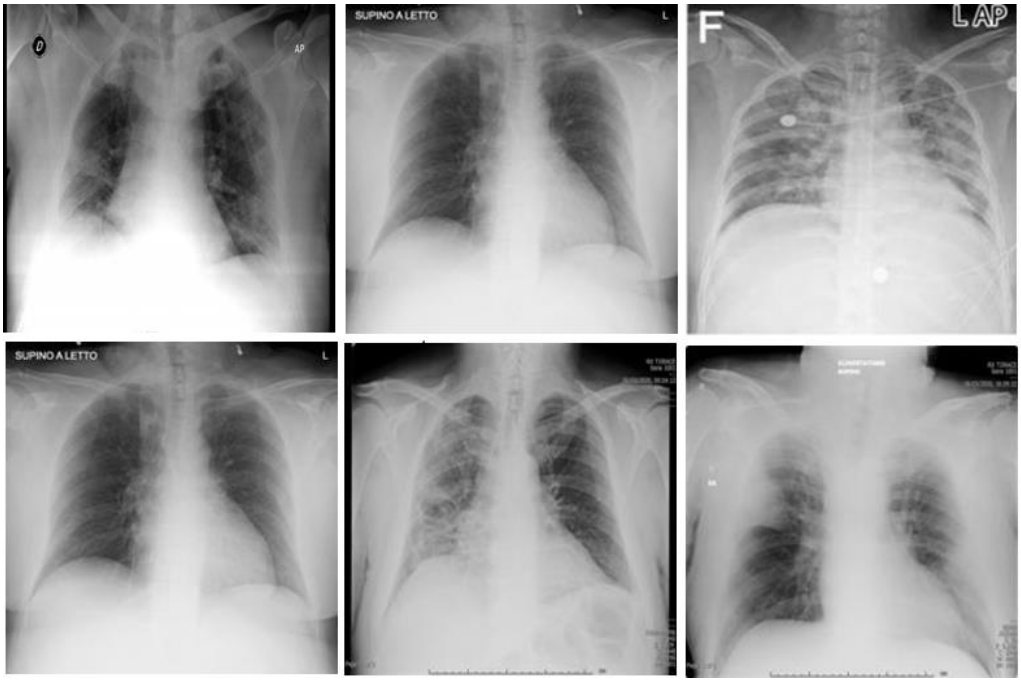
In this Chapter, a hybrid deep learning approach with textual and morphological features has been proposed in order to classify and predict Covid-19 cases by utilizing X-ray images of both affected and normal cases. Sections 8.1.1 and 8.1.2 show the dataset details and the proposed work description.

8.1.1 Materials: Covid-19 Datasets

S. No	Dataset References	Covid-19 Images	Normal Images
1	Covid-19 Image Database [1]	1200	1200
2	github.com - COVID Images [2]	700	700 (collected from various sources)
3	Combined Datasets [1- 9]	2500	1500

Table 1: Dataset Information

The datasets consist of Covid-19 affected images as well as normal images.
The lung images affected by Covid-19 were gathered from various websites and research organizations. The affected as well as normal datasets have been included in the reference list [1, 3, 4, 6, 7, 8].
Table 1 presents the information regarding the training and testing images utilized to predict the images that were Covid-19 positive. The samples Covid-19 affected and normal images are shown in Figure1.



A. Sample of COVID-19 Affected Images

B. Sample of Normal Lung Images

Figure 1: Sample of Covid-19 Affected Images and Normal Lung Images

8.1.2 Methodology

The proposed prediction consists of pre-processing, segmentation of affected and non-affected images, training and testing, prediction of Covid-19, and classification of the various stages of prediction. The architecture of the prediction using X-ray lung images has been illustrated in Figure 2. The proposed work is consisted of four steps such as:

i. Initializing the sample images or visualizing X-ray images from the patient data.

ii. Pre- Processing of datasets.

iii. Predicting Covid-19 (positive) as well as non-Covid (negative) images using VGG-16.

v. Classification of various stages of Covid-19 images.

A. Pre- Processing of Dataset:

The pre-processing procedure converts the image into digital images. The main aim of digital image processing is to improve the features and suppress any distortions in the image features. The images in the datasets have non-uniform shapes and sizes, therefore, before initiation of training, there was a need to resize the images. The pre-processing procedures were performed using texture and morphological features. Energy, Entropy, Correlation, Contrast, Homogeneous are some of the features that were calculated using the textural properties and are represented in the Equations 1 to 5 and these Equations are for Energy, Entropy, Contrast, Correlation and Homogeneity features respectively. The Gray-level matrix used to calculate the textural structures of the image have been included [11].

$$Energy = \sum_{i,j=1}^{N-1} (P_{i,j})^2 \tag{1}$$

$$Entrophy = \sum_{i,j=1}^{N-1} -\ln(P_{i,j})P_{i,j} \tag{2}$$

$$Contrast = \sum_{i,j=1}^{N-1} (P_{i,j})(i-j)^2 \tag{3}$$

$$Correlation = \sum_{i,j=1}^{N-1} (P_{i,j}) \frac{(i-\mu)(j-\mu)}{\sigma^2} \quad (4)$$

$$Homogeneity = \sum_{i,j=0}^{N-1} (P_{i,j}) \frac{P_{i,j}}{(i-j)^2} \quad (5)$$

Where $P_{i,j}$ is normalized symmetrical, N is number of grey levels, μ is mean intensity, σ is variance of intensity. The textural features were calculated using various equations used in [12-18]. The prediction size, radius of the prediction area, image equivalence, and dispersion were calculated using morphological features shown in Equations 6 and 7. The morphological features were measured using various equations used in [11]. The varying scale morphological analysis [11] was utilized in order to forecast the morphological structures of Covid-19 X-ray lung images [19-23].

$$Difference = \{w | w \in A, w \notin B\} \quad (6)$$

$$Equalance = \{w | w \in A, w \in B\} \quad (7)$$

In the proposed work three types of datasets images were resized. In the first dataset [1], 1200 Covid-19 positive as well as 1200 negative images were checked and resized into 256*256 pixels. The rectangular lung images were resized into 256 pixels. For better prediction, the images were augmented in different ways. The various phases of Covid classification from the dataset is described as follows which could be used for training and testing phases. Further various stages of Covid images are also trained and tested. It is represented as 5 steps as follows:

1. The proposed learning-based hybrid model helped improve the efficiency and prediction accuracy by utilizing X-ray Images.
2. 3400 Covid affected images and 4400 non-affected images were used during implementation. In total, 7800 images were used for training, testing, and evaluation.
3. The X-ray image-based predictions incorporated texture and morphological features using hybrid deep learning techniques.
4. The Prediction and classification of positive as well as negative Covid-19 images were undertaken using VCC-16 and light GBM. The use of Light GBM significantly decreased the complexity and computation time involved.

5. The various classifications and predictions derived from the affected images were carried out by using the layer-wise features of Deep Belief Network (DBN).

B. Segmentation of Covid-19:

The VGG-16 was used to segment the Covid-19 (positive) as well as the Non-Covid-19 (negative) images. VGG-16 is a deep conventional neural network technique for classifying or segmenting large-scale datasets or images. The previous works [24-27] aimed to only predict, but this work has endeavoured to predict as well as classify the various stages of Covid-19 images. The VGG-16 had an accuracy of 92.7 in the top five tests of images using 138,357,544 parameters, whereas the VGG-19 had an accuracy of 90.2 in the top five tests of images using 143,667,240 parameters. This work used VGG-16 as a means of using fewer parameters and achieving higher accuracy of classification and prediction.

The VGG-16 contained fixed-sized images such as 224*224 as an input and had an output of 1000 values. But in this work, the output of the images were classified as positive and negative. The general representation of the output of VGG-16 is shown in Equation 8.

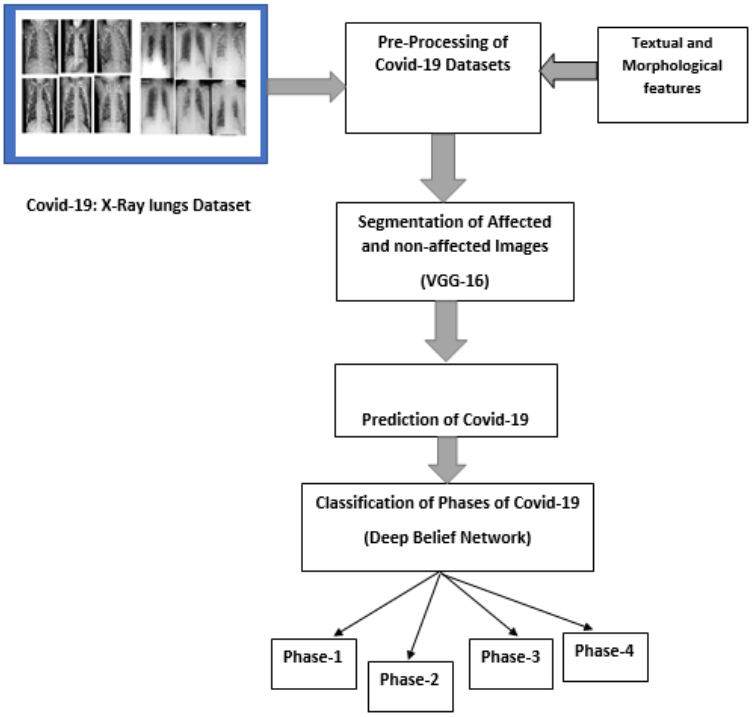


Figure 2: Architecture of Prediction using X-ray Lung Images

$$Y = \begin{pmatrix} Y_0 \\ . \\ . \\ Y_{999} \end{pmatrix} \tag{8}$$

In this work, the output prediction of the proposed work is either Negative (Y_0) or Positive (Y_1). The output using VGG-16 representation is shown in Equation 9.

$$Y = \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} \tag{9}$$

The VGG-16 was used to segment the given input images as positive or negative. The diagrammatic representation of the proposed work with VGG-16 and light GBM has been shown in Figure 3.

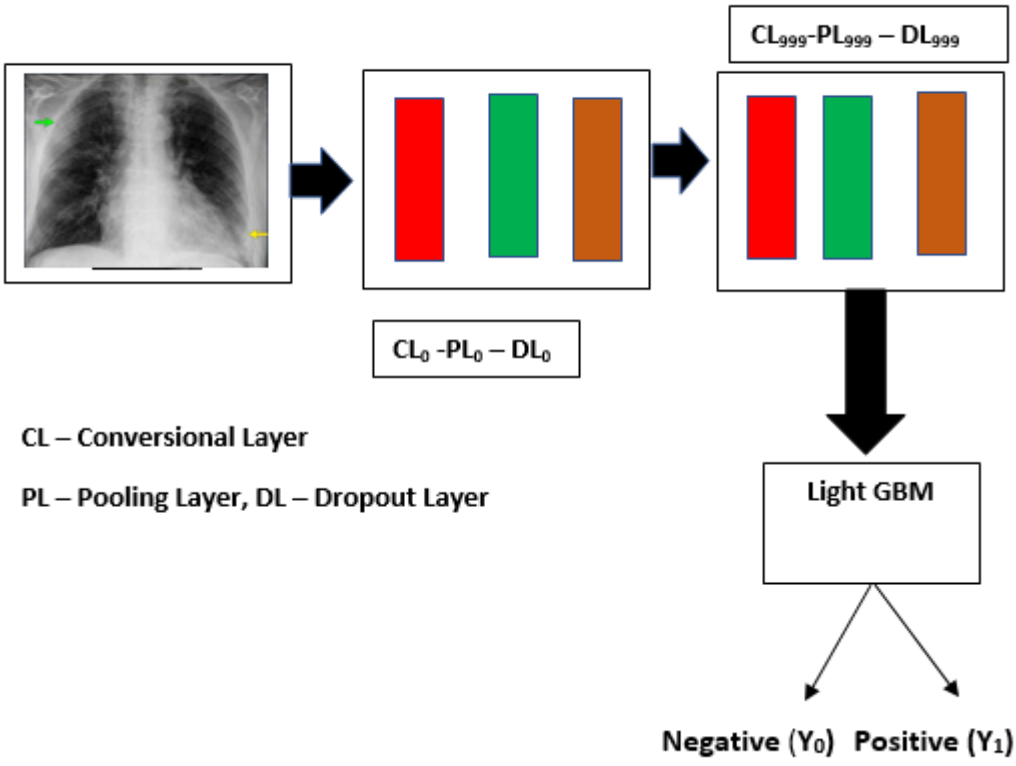


Figure 3: Segmentation of Images using VGG-16 and Light GBM

Light Gradient Boosting was used to boost the outputs as it supported the huge size of datasets. Compared to other ensemble learning methods, it is 10 times faster in the computation of a huge number of datasets.
The combination of VGG-16 and light GBM decreased the computation time and increased the accuracy of predictions. VGG-16 and light GBM algorithm are better combination in image-based prediction technique.
Even though light GBM is combined with other prediction algorithms, the accuracy and predictions are considerably less.

C. Stages of COVID-19 X-ray Images

The various stages of the predicted Covid-19 images were performed using Deep Belief Network (DBN). The DBN is a multiplicative graphical representative model and is another class of deep neural networks. It has multiple hidden layers from the observation and tinning layers. The normal DBN performs the classification with each layer having a set of features classifications. One such a belief-based monitoring on real time application and compulsion was discussed for image data driven method [28]. The work was that the effective useful features of images were extracted to guide the statistical calculation to monitor further. Yet another work using belief network was for breast cancer prediction from image and it was discussed [29] as follows. The unsupervised and supervised algorithm has been utilized for pre-training and fine-tuning processes respectively for automation feature extraction using this belief network model. The images of dataset were patched and logistic regression algorithm was used to support for gaining probability matrix for positive and negative sample prediction. The feature classifications are performed using encoders. The supervised learning data was predicted using gradient functions and unsupervised learning was performed using Restricted Boltzmann machine network or autoencoder. The DBN started from the training layer and extracted the different features in each layer. The main properties of the DBN included training the features of Covid-19 images, pixels, and signals of images directly in each layer. From the second layer onwards, the features increased and every layer's classification features increased in the belief network. The DBN network is shown in Figure 4 and each layer was well interconnected to make the classification of the given dataset.

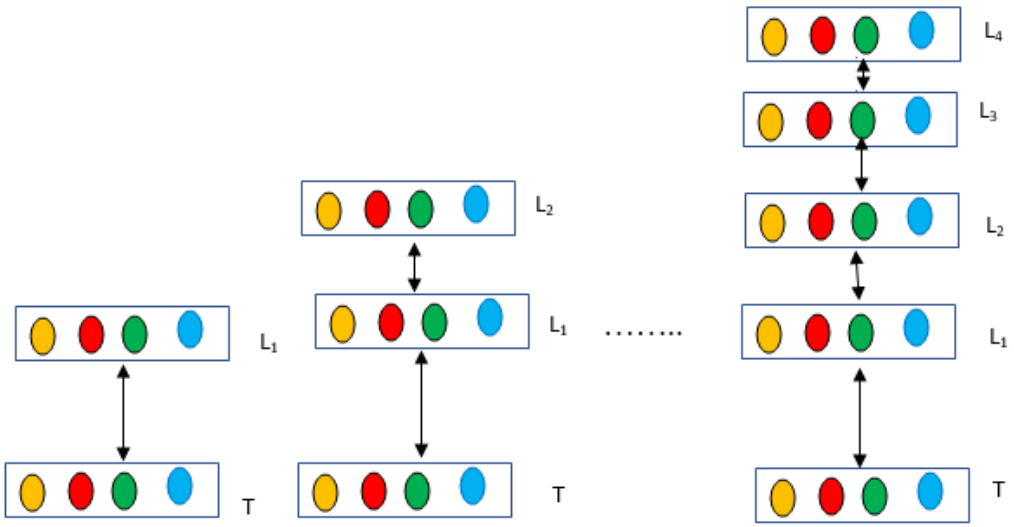


Figure 4: Classification of Various Stages of Images using Deep Belief Network

The supervised model had three parameters for training the nets such as weight (W_{AB}) and related properties of images A_n and B_n, represented in Equation 10 and 11.

$$A_n = [a_1, a_2 \dots a_3] \tag{10}$$

$$B_n = [b_1, b_2 \dots b_n] \tag{11}$$

The energy equation represented for training and computation is represent in Equation 12.

$$E(x, y, \theta) = \sum_{x}^{y} w_{ij}\, x_i y_i - \sum_{x=1}^{x=n} a_i\, w_i - \sum_{y=1}^{y=n} b_i y_i \tag{12}$$

The continuous random prediction of COVID X-ray image characteristics and multi-variants of image distribution was performed using Probability Density Function (PDF) and is represented in Equation 13.

$$(x, y) = \frac{e^{-E(x,y)}}{\sum_{x.y}^{n} e^{-E(x,y)}} \tag{13}$$

The weight as well as the learning rate of training of X-ray images were performed and is represented in Equation 14.

$$P(wl) = \sum_{i=1}^{n} \left(w_{ij} x_1 + A_i\right) + \sum_{j=1}^{n} (w_{ij}\, y_1 + B_i) \tag{14}$$

Generally, the sample data prediction and classification was performed using supervised and unsupervised learning. The supervised learning was performed using Gradient Descent method and its representation is the shown Equation 15. The supervised learning network was fine tuned from top to bottom. The associated representation of the gradient function with different parameters can be represented as

$$O = W_1 A_1 B_1 + w_2 A_2 B_2 + \cdots + W_n A_n B_n \tag{15}$$

Similarly, the unsupervised learning was performed using Boltzmann machine network. It was performed from the bottom to the top layer and matched the features to the maximum possible limit using Equation 16.

$$O = \sum_{i=1}^{n} (P\frac{x}{v})(\frac{A_n B_n}{w})) + \sum_{j=1}^{n} (P(\frac{y}{v})(\frac{A_n B_n}{w})) \tag{16}$$

D. The Working of Hybrid Approach Algorithm

The hybrid approach featured selection was performed using texture and morphological parameters. A huge number of datasets were processed and predicted using VGG-16 and light GBM. The various stages of Covid-19 affected images were classified with the use of DBN. Initially, the collected datasets were noisy and features were predicted with the help of feature selection methods. The texture and morphological features are mentioned in the Equations (1-7).

After features selection, the Y_0 to Y_n number of classification and prediction the images were predicted, such as positive (Y_1) or negative (Y_0). The predicted images Y_1 and Y_0 images were classified as per the impact of the Covid infection as per Table 1 (see chapter 3). The workflow process is presented in Algorithm 1.

Algorithm 1:

Input: Various X-ray Images.

Output: i. COVID-19 Predicted images.

ii. Various stages of Predicted Images.

Hybrid Deep Learning Method:

Step 1: Initialize images.

Step 2: Extract each image size (224*224).

Step 3: Extract the texture features [Equations 1-5].

Step 4: Extract the morphological features [Equations 6-7].

Step 5: Initialize the VGG-16 and train the various features.

Step 6: Compute the datasets using GBM.

Step 7: Classify and predict ((Y_1) (Y_0)).

Step 8: Compute the predicted images.

Step 9: Train the various classification phases.

Step 10: Extract the features from the layers.

Step 11: Extract the training and testing feature of phases.

Step 12: Result (O) = Classify (training feature, T).

Step 13: Output: Result (O) = Phases of COVID-19.

For the purpose of continuous predictions, the methods were looped into the input images. As per the looping, the images were examined and the predicted images were sent for classification of various stages of Covid-19 images. These Phases are mild cell incursion and viral duplication of nose and lungs, moderate amount replication which affect the lungs and the immune system, the replication of consolidated sections of the lungs threatening, and critically affected failure of multiple organs respectively.

8.2 Experimental results and discussion

8.2.1 Datasets and Performance Parameters

The proposed approach detected and classified the various stages of Covid-19. The implementation of various datasets used has been mentioned in Table 2. All the

datasets were classified for training, testing and validations. The 70%, 15% and 15% of the datasets were used for training, testing and validations, respectively. The number of normal and Covid affected images used for training and testing are shown in the Table 4.

In total, 7800 images were utilized in the datasets.

The 4400 Covid affected images were used and 3400 normal lung images underwent classifications and predictions. The testing and validation images were not used for training and similarly, the lungs images used for training were not used for testing and validations. The sample predicted images with labels from the huge number of datasets are shown in the Figure 5.

Datasets	Normal Images	Covid-19 Images
Total Images	4400	3400
Training	3080 (70%)	2380 (70%)
Testing	660 (15%)	510 (15%)
Validation	660 (15%)	510 (15%)

Table 4: Training, Testing and Validation Ratios of Images

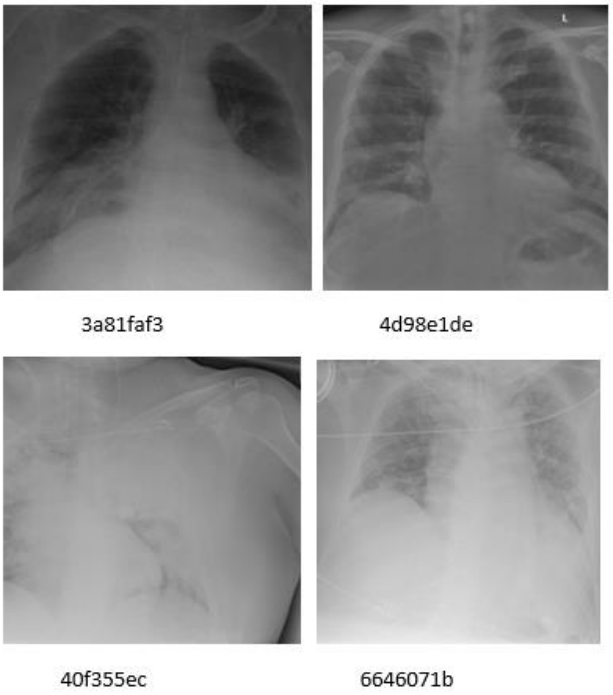


Figure 5: Covid Predicted Images using the Hybrid Approach

The performance was calculated using various parameters such as accuracy, sensitivity, and specificity. These parameters were automatically calculated using other supporting parameters such as true negative, true positive, false negative, as well as false positive, which were used to compute the metrics automatically using the hybrid method. The predicted results were compared with other methods such as the Scratch Model, AlexNet, and ResNet50 [30]. Compared to the previous models, the proposed model produced better accuracy (Acc), sensitivity (Sen), and specificity (Spc).

8.2.2 Classification and Prediction Performance

The proposed classification and prediction methods are shown in Figure 6. The proposed hybrid method, the feature extraction, classification and prediction results were received using a combination of VGG-16 and DBN. Each input was trained using the bottom training layer and features were extracted from each layer. Initially, the basic features of the images were extracted from each layer with the help of techniques used for extracting texture and morphological features.

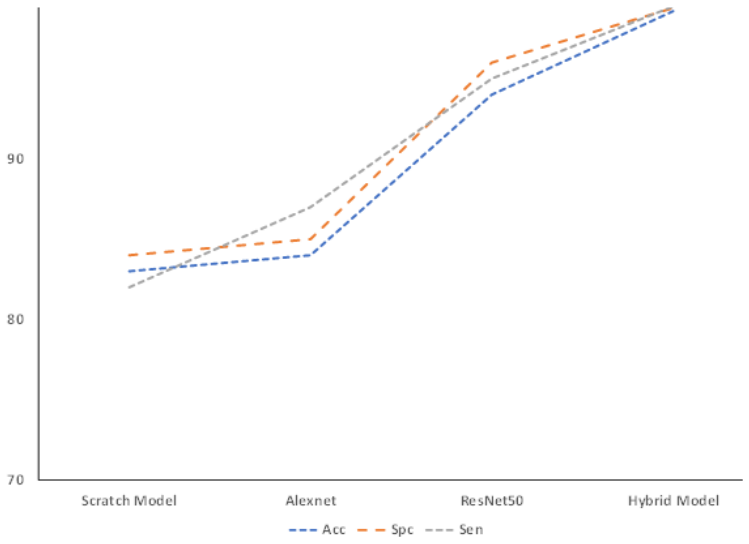


Figure 6: Features Classification and Prediction

The initial screening and supporting features were seen to produce better accuracy for predictions. Using VGG-16 Net, 0-999 layers were created with each layer having different features for predictions and classifications. This automatically resulted in increased prediction and classification features compared to the other methods. In this proposed hybrid method, a total of 7800 X-ray lung images were utilized for testing, validation as well as training. This automatically resulted in decreased the

time complexity of computations. To avoid increased computation time, light GBM was used. The aforementioned methods such as Scratch Model, AlexNet, and ResNet50 used lesser percentage of the dataset, and therefore led to lesser computation cost. However, in this proposed hybrid work, light GBM produced overall better classification, accuracy, and used less storage capacity in terms of memory and computations.

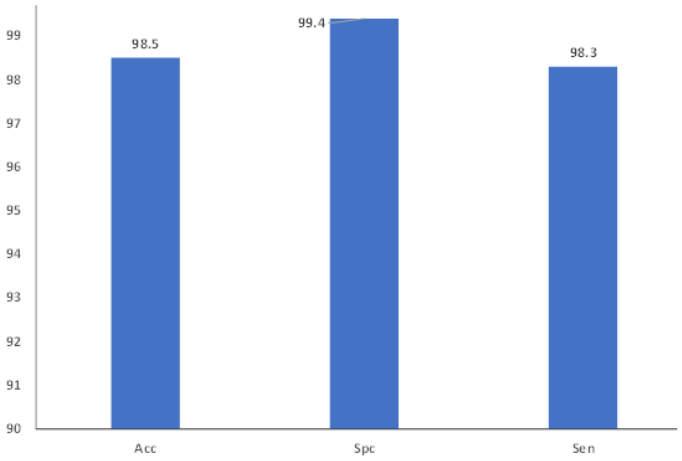


Figure 7: Bacterial Pneumonia Prediction Rates

The presence of Bacterial Pneumonia was used to find the influence of certain bacteria in the affected X-ray images.

Figure 7 illustrates the Bacterial Pneumonia prediction rates. Accuracy, sensitivity, as well as specificity parameters were utilized for the purpose of measuring performance. The different labelled and unlabelled features in the images such as smoke, Viral Pneumonia etc. were used to find the influence of the virus.

This hybrid method introduced the Bacterial Pneumonia prediction rates, which were not introduced in previous works.

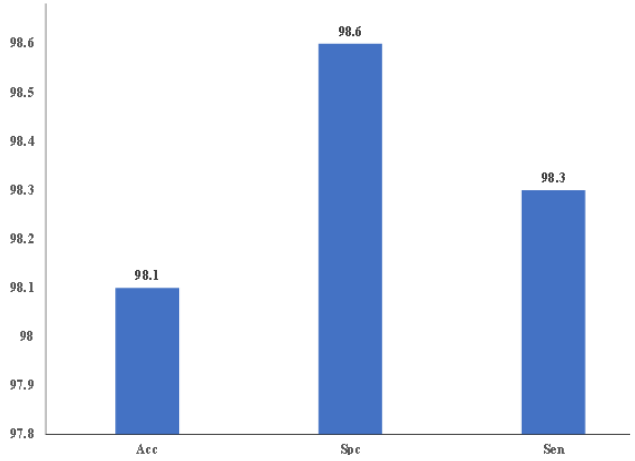


Figure 8: Various Stages of Prediction of Covid-19

The Figure 8 shows the stage-wise prediction performance and corresponding parameters. The same parameters of accuracy, sensitivity as well as specificity were utilized for the performance calculation and the predicted results were found to be 98.1%, 98.6% and 98.3%, respectively. The VGG-16 layers and BDN provided better classification of the affected images. This classification was used for better decision-making. Table 1 shows the stage-wise identification and sample images of various stages. The DBN layers were used to classify the stages of the Covid images. The combination of 999 layers of VGG-16 and DBN layers produced better classifications of the Covid stages. The number of affected images and corresponding classification of predicted images are shown in Table 5. Similarly, the prediction of total Covid cases, cases of Bacterial Phenomena, the classification of stages and the various overall predictions are shown in Table 6.

Total COVID Images	Accuracy % of Stage Wise Prediction	Total Predicted Images	No of Stage-1 Covid Images	No of Stage-2 COVID Images	No of Stage-3 COVID Images	No of Stage-3 COVID Images
3400	98.1	3332	1569	1078	613	72

Table 5: Classification of Various Stages of COVID Images

Predictions	Accuracy	Sensitivity	Specificity
Prediction of COVID-19	99.2	99.4	99.5
Bacterial Pneumonia Prediction	98.5	99.4	98.3
Classification of Stages	98.1	98.6	98.3

Combined Multi-classification Prediction	98.6	99.1	98.7

Table 6: Various Predictions of the Proposed Method

8.3 Conclusion and future work

Covid-19 has increased fears of infection and mortality rates in families and societies the world over. The healthcare workers are under tremendous pressure to accurately predict the stage of the disease in those affected by it. Therefore, timely predictions play an important role and can save lives. Covid X-ray images are a good resource for determining the status of affected people and can help provide better decision making for treatment protocols. In this work the lung X-ray images were classified into four stages for the purpose of predicting and supporting such decision making. This proposed hybrid method combined various feature extraction methods, prediction and classification methods. The Texture and Morphological methods are used to determine the basic features of the Covid-19 images.

The Light GBM and VGG-16 were used for classifying and predicting the Covid images. The light GBM helped reduce the computation time and VGG-16 increased the prediction rates. The Deep Belief Network was used to classify the various predicted images to determine the rate of infection in the lung X-ray images.

In this proposed work, a huge number of datasets were used in the implementation. The proposed hybrid method used three parameters for measuring prediction performance such as accuracy, sensitivity and specificity. The Covid-19 images were also predicted in three distinct ways, such as Covid affected and non-affected images, Bacterial influence rates and various stages of the affected images. The overall predicted results were respectively, 98.6, 99.1 and 98.7 percent accurate. Therefore, the proposed work was observed to produce improved results in comparison to previous methods, with the stage-wise predictions adding to the novelty of this work. The future direction of this research would benefit from finding ways of further increasing the speed of prediction and reducing the time complexity of the proposed method. Further research can be proposed for the application of multiple prediction models that can lead to increased accuracy of predictions. More fine-tuned prediction-based parameter could be introduced in proposed model in feature to automate the process for prediction.

That would be feature span of extension in learning-based research direction. If the more feature extraction specific parameters are included in future, then the proposed algorithm would be modified considerably in all prediction aspects of Covid image. The proposed hybrid algorithm will support future Covid-19 X-ray image predictions

even if version changes to be happened. In such a cased some parameters may be included as per future version of Covid-19.

The impact of X-ray prediction of Covid-19 is essential in reality since every covid affected sample images had been appeared different in prediction using proposed algorithm. The proposed algorithm may be extended to automate and predict false positive and false negative predictions in near future.

CHAPTER 9

New Machine Learning Features: Real-world model for bitcoin price prediction

9.1 Introduction

Cryptocurrencies are virtual or digital currencies that are used in today's financial systems. Because no government, central authority, or bank issues these virtual currencies, these are decentralized. All cryptocurrencies are based on Blockchain technology, which is incredibly complicated and tries to store data in such a way that it is tough and beyond expectations to hack and modify. Cryptography further secures these currencies, making it hard to create fraudulent cryptocurrencies. Cryptocurrencies are still in their infancy, and it is impossible to say if these can ever be extensively adopted in global markets, but El Salvador became the first country to use bitcoin as legal money alongside the US Dollar in November 2021.

There are a bunch of cryptocurrencies flowing in the digital market. In such cryptocurrencies Bitcoin is the most well-known, and it is impacted and influenced by external variables such as social sites, digital data, market analysis and so on. It was first introduced by Nakamoto [1, 2]. Peer-to-peer transactions are possible thanks to blockchain technology. Blockchain, the technology that underpins the Bitcoin cryptocurrency system, is extensively critical for assuring enhanced security and privacy across a variety of sectors, comprising the IoT [3]. It is basically a distributed digital record of transactions that spans the whole network of computer systems that make up the blockchain. The first component of the blockchain is a transaction, and the second is a block [4, 5]. The participant's activity is represented by the transaction, and the block is a data collection that stores the transaction as well as extra information like the right sequence and creation timestamp.

The cryptocurrency market is currently the most attractive domain for financial speculation. Many individuals have gained a lot of money by speculating in the digital markets, and also marketing investment process which are filled with concealed pitfalls still endowing in bitcoins, ethereum etc. [2, 3].The number of machine learning algorithms to use to reduce these risks, as well as a comprehensive set of possible market-predictive characteristics [6]. Due to the capacity to dynamically pick from a potentially enormous number of characteristics and understand complicated, high-dimensional correlations between features and targets, machine learning approaches have become more popular in this sector [5, 7]. Artificial Intelligence with machine learning techniques is attractive due to variances in forecasting capacity per coin [8]. Low-volatility cryptocurrencies are more predictable than high-volatility cryptocurrencies.

Because of price fluctuations and instability, cryptocurrency prices are tough to

predict. This vacuum in the field is filled by comparing several machine learning models for forecasting market movements of the most relevant cryptocurrency - bitcoin. Investing in Bitcoin is comparable to investing in stocks: none of the risk factors that can identify negotiations and changes in stock price related to cryptocurrencies.

In case of stocks, there are various parameters available like P/E value, ROE (return on equity), ROCE, EBITDA, etc. to predict the prices of stocks but in case of crypto currencies there are only few parameters available, price movement is highly volatile and dynamic in nature, trends and pattern are complex and changes dynamically within a short period of time and there are multiple seasonality with hourly, daily and weekly data. Most of trading algorithms requires a lot of parameter, they highly depends on historical data, they cannot handle high volatility and strong multiple seasonality. Due to highly dynamic and volatile nature, there might have missing observations and large outliers. There are various algorithms available for them like ARIMA and LSTM-based recurrent neural network but all these requires many parameter, LSTM based RNNs are difficult to interpret and it is challenging to gain intuition into their behavior, these algorithms are good for long term pattern but in case of crypto pattern and trends changes within few hours. So, there is a need of algorithm which can handle all these problems and can even deal with holidays known in advance and missing observation and large outliers and seasonal effect cause by human behavior and can provide more accurate prediction.

In this chapter, the aims is to achieve the a model to predict closing price of Bitcoin along with Bitcoin Opening Price, Bitcoin Day High Price, Bitcoin Day Low Price, Bitcoin Day Volume and Market Capitalization of Bitcoin on particular Day by using deep leaning algorithms and various concepts of machine learning, which can find hidden patterns in data, combine them, and make considerably more accurate predictions. To achieve the aim the following task will be performed:

a) Descriptive Analysis
b) Exploratory Data Analysis Along with Data Pre-Processing
c) Statistical Test to remove seasonality and make it stationary {using Ad fuller Test}
d) Data Transformation
e) Preparing the data, smoothing data and try to adjust seasonality. The concept like Differencing is often used.
f) Building a Time-Series Model for which Fbprophet Library is being used.
g) And at last, cross-validation is performed on model.

After applying all these steps, the generated model is:

1. That can perform all features of previous models.
2. That doesn't require much prior knowledge or experience of forecasting time series data since it automatically finds seasonal trends beneath the data and

offers a set of ‘easy to understand’ parameters.

3. That can Deal with holidays known in advance, missing observations, and large outliers.
4. That can encounter means hourly, daily, or weekly observations with strong multiple seasonality's.

The fundamental goal of these models is to create a trustworthy prediction model based on previous bitcoin prices that investors can trust. The article also attempts to address few issues like 1. Utilizing the machine learning techniques in prediction of cryptocurrency prices for investors and decision makers 2. Selection of best fit model for predicting prices of cryptocurrency.

Next section discusses about the study related to crypto currency. Section 9.3 provides the details of the dataset. Detailed methodology is discussed in detail in Section 9.4 which is followed by section 9.5 explaining the implementation and results. Further section 9.6 is related to discussion of the results followed by conclusion.

9.2 Literature survey

There is very less availability of price prediction models for Bitcoins as it a modern technology in current scenario [9]. Time series models interacts with data from daily time series, 10-minute, and 10-second intervals. These models constructed three time series data sets for 30, 60, and 120 min, then used GLM/Random Forest to generate three linear models from the datasets. To estimate the price of Bitcoin, these three models are linearly integrated [10] handles with data from daily time series, 10-minute, and 10-second intervals. To estimate the price of Bitcoin, these three models are linearly integrated. Instead of directly anticipating the stock's future price, the writers in anticipate the stock's trend. A pattern may be drawn from the trend. These can make both small (day or week-long) and large forecasts (months). It can be discovered that the long forecasts gave superior outcomes, with an accuracy rate of 79 percent. Another intriguing method reflected in the study is the network's performance evaluation criteria which functions on projected output. The authors applying machine learning approaches handles both deep learning technique and regression techniques for prediction of Bitcoin integrating with gradient descent and linear search. According to a study published applied for high dimensional data which is related to Bitcoin daily price prediction, shows that logistic regression and linear discriminant analysis reach a 66 percent accuracy rate. Outpacing (a complex technique which is based on machine learning), on the other hand, outperforms the standard results for everyday prediction for price, with 66 percent and 65.3 percent accuracies for statistical approaches and machine learning algorithms, respectively. The study demonstrated two types of prediction models created using Bayesian optimized RNN and LSTM to forecast the price of BTC [11]. The study found that LSTM performed better, with a 52 percent accuracy and an RMSE of 8%.

Time series are a specific instance of LSTM-based recurrent neural networks, which

are perhaps the most powerful way to learning from sequential data. When learning from large datasets with complex patterns, the potential of LSTM-based models is fully realized. They do not rely on certain assumptions about the data, such as time series stationarity or the availability of a Date field, like ARIMA or Prophet do. However, the downside is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior is tough. In order to produce decent outcomes, rigorous hyperparameter adjustment is also essential. In the case of cryptocurrencies, substantial multiple seasonality has a significant impact on LSTM accuracy.

Most of the investment process is based on a cryptocurrency's previous pricing. Building Markov chains is one of the most essential tactics used by investors. This technique entails using numerous decision trees to select the cryptocurrency that is expected to produce a higher return when sold, as well as comparing the anticipated return to the actual amount. is an example. ANN is also very effective in optimizing problems [11, 12, 13]. According Markov chain models, the transitioning from one state to another state solely depends on the current state, in these models we can see an ignorance of all previous trends and other than that in case of cryptocurrencies is might possible that the current state is a result of seasonality and volatility, at this conditions the efficiency of model might compromised. Whereas proposed prophet model manages the trends as well it able to predict the trends at the currents states. Prophet model is also able to deal with missing data values and outlier along with the seasonality affects or trends. From few years, in time series forecasting a ARIMA model is widely spread i.e. known as autoregressive integrated moving average [14]. ANN also an option and substitute for forecasting [15]. The superiority of ARIMA models and ANNs in forecasting performance is frequently compared, with inconsistent results [16]. The study suggested a hybrid model applying ARIMA and ANN [17]. The combined model can be an effective technique to increase forecasting accuracy attained by any of the models used alone, according to experimental findings with real data sets. The results signifies that the Bitcoin prediction can be accurate by applying ML ensemble method [4]. Decision making should be done at right time interval while reducing the risk. Over few decades Autoregressive integrated moving average (ARIMA) is one of the most widely used linear models in time series forecasting [18]. Recent research into artificial neural networks (ANNs) for forecasting suggests that ANNs might be a viable alternative to standard linear approaches. The superiority of ARIMA models and ANNs in forecasting performance is frequently compared, with inconsistent results. To take use of the distinctive strengths of linear and nonlinear concept in ARIMA and ANN, a hybrid technique including both ARIMA and ANN models is suggested in this study [19]. Experimental findings signify, for improving forecasting accuracy the integrated model is more efficient in comparison to other models if applied separated and isolated.

However, when it comes to cryptocurrency, where data is restricted in terms of characteristics, seasonality can occur on a weekly, daily, or even hourly basis, data has big outliers, and does not entirely rely on historical data, the market in cryptocurrencies moves dynamically within a short period. All of these may have a

negative impact on the ARIMA model's performance. Further tuning is sometimes required for algorithms like ARIMA to generate respectable results, which is out of reach for many people who are not properly qualified specialists. To address the constraints of the ARIMA model in terms of crypto trading, a new model based on Facebook Prophet is developed in this study. Because Prophet is primarily built to find patterns in business time series, it requires minimal hyperparameter tweaking. Prophet is robust to missing data and trend shifts, and it can usually manage outliers and trend shifts caused by new items and market events. Unlike auto. arima, Prophet shows a realistic seasonal pattern, even if the absolute numbers are a little off from the actual data. Prophet is unique in that it requires no prior knowledge or expertise in forecasting time series data since it automatically detects seasonal trends underlying the data and provides a set of 'simple to comprehend' parameters. Prophet is also built to deal with holidays that are known ahead of time, missing data, and significant outliers. As a result, even non-statisticians may use it and achieve pretty decent results that are often on par with, if not better than, those generated by specialists.
The patterns of time series are complex and vary dynamically over time, but Prophet only pays attention to such changes when the trend shifts. The seasonality prior scale is ineffective, however the greater trend prior scale performs better. However, because Prophet, unlike other models, does not directly consider recent data points, if there are some seasonality patterns in the dataset and these patterns are not consistent or smooth (as in the case of Cryptocurrencies), this can severely hurt performance when prior assumptions do not fit. To overcome all of Facebook Prophet's limitations, first and foremost, data must be subjected to Quantitative forecasting, which will include trend projection, the Naive technique, assessing seasonality using the Adfuller Test, moving averages, and exponential smoothing. After all of this, the FbProphet model will be able to outperform all other models by a large margin.

Comparison of LSTM and ARIMA model

In Figure 1, we had used data Bajaj Finserv Ltd., an Indian Financial Company in Order to compare the two models in Figure 1. The Data spans the period from 2008 until end of 2021. From the Figure 1, we can clearly see ARIMA model yields better performance than LSTM model.

9.3 Datasets

The data for this study came from an open-access website - https://www.kaggle.com/team-ai/bitcoin-price-prediction/version/1.
Used data is historical data to form Naïve model and to search for historical trends. Data can be use at real time using Big data concepts like Spark Streaming and Kafka.

Data is made up of a single.csv file which consists of date, open, high, low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. Data can also be downloaded from https://in.tradingview.com/chart/?symbol=COINBASE%3ABTCUSD from where data can easily export data in form of .csv file. The dataset used in project do not have outliers, but if dataset have some outliers and missing values they can be removed using averaging, replacing null value with mean similar type of data, by adding and subtracting the variance in outliers etc.

The data for this study came from an open-access source. It is made up of a single.csv file which consists of date, open, high, low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. This model can be applied on any other Data of similar type. Table 1 shows sample of raw data.

Raw Data contains "Open" which represents opening price, "High" represents highest price, "Low" represents lowest price and "Close" represents closing price of Bitcoin on particular "Date". The available data is raw data which might have some outliers, so firstly data need to be prepared. For data preparation, libraries like Pandas (extremely used for data manipulation and data cleaning), Numpy (to perform numerical operation on data) and for data visualization libraries like matplotlib and seaborn are used.

Description of Data is as follows.

The datatype of Raw Data is As Follows.

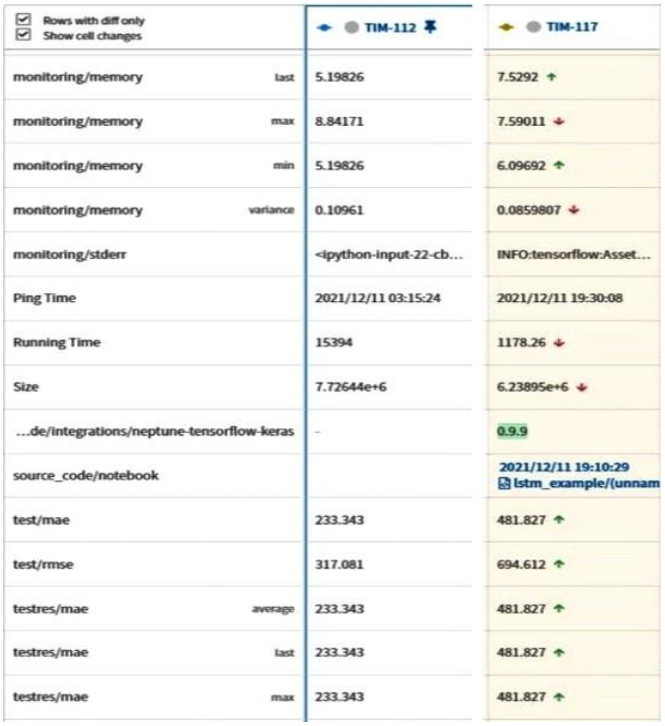
Table 1 shows a sample of the raw datasets, Table 2 shows description of the dataset like mean, standard deviation, minimum, etc. and Table 3 shows the datatype of columns in the Dataset.

In above figure "Date" have object datatype, but "Date" feature must support something known as timestamp nature because it is the must condition for a Time Series case so firstly convert it to date time format and then to maintain hierarchy, sort the data according to Date. For Time Series, it is must to make "Date" as index feature to avoid key Error.

9.4 Methodology

To achieve the aim of study, firstly fetch raw data from third party API's or the data can be extracted from some big data bases like MongoDB or web Scrapping. Then a lots of data cleaning is performed on raw data and must perform Exploratory Data Analysis (EDA) on this data. Now from this data to build a model, in such case, Multiple algorithms can be used. One of them is Naive Model which can also be termed as Base Line Model, then there are some amazing models like auto regressive model, then there are some moving average model, then there is ARIMA model, so there are tons such models available there for data. If there exist some seasonal data, then there is something known as SARIMAX. During data cleaning, once have

cleaned data over here there is one more thing which is exactly known as feature engineering which will almost take 70% for this entire project. In this feature engineering it can be detected that whether data is Stationary or not. So to check, there are a lots of steps like some statistical test from this the trend of the feature can be understood by basically using Line plot function or Line plot curve, then there are some statistical test like Augmented Dickey Fuller Test (adfuller Test) one of the famous Test to detect whether data is Stationary or not because stationarity is crucial term relating to time series analysis that affects robustly the interpretation and analysis of data. Every data point is assumed independent in time series models for prediction and forecasting. Stationary data is the crucial concept, it should be consistent and not vary with time. It can be concluded that values may be different by avoiding the consistency in general. For sustaining the stationarity mean and variance are key values. Time graphs are generated for price of bitcoin. Very first there is to work with multiple libraries like Pandas which is extensively used in case of data manipulation or data cleaning then there is NumPy which is exactly Numerical python which will be used when there have to perform some numerical stuff on data, then to deal with data visualization stuff Matplotlib will be used even sometime Seaborn which will return some interactive visualization compare to matplotlib and there is one more library which is Sklearn which is extensively used in case of Data Modelling and with respect to time series use cases library known as stats models will be used. Figure 2 shows raw sample data.



Rows with diff only / Show cell changes		TIM-112	TIM-117
monitoring/memory	last	5.19826	7.5292
monitoring/memory	max	8.84171	7.59011
monitoring/memory	min	5.19826	6.09692
monitoring/memory	variance	0.10961	0.0859807
monitoring/stderr		<ipython-input-22-cb...	INFO:tensorflow:Asset...
Ping Time		2021/12/11 03:15:24	2021/12/11 19:30:08
Running Time		15394	1178.26
Size		7.72644e+6	6.23895e+6
...de/integrations/neptune-tensorflow-keras		-	0.9.9
source_code/notebook			2021/12/11 19:10:29 lstm_example/(unnam
test/mae		233.343	481.827
test/rmse		317.081	694.612
testres/mae	average	233.343	481.827
testres/mae	last	233.343	481.827
testres/mae	max	233.343	481.827

Figure 1: The mean square error and the mean average error ARIMA and LSTM models can be seen next to each other.

	Date	Open	High	Low	Close	Volume	Market Cap
0	Jul 31, 2017	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000
1	Jul 30, 2017	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000
2	Jul 29, 2017	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000
3	Jul 28, 2017	2679.73	2897.45	2679.73	2809.01	1380,100,000	44,144,400,000
4	Jul 27, 2017	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000

Table 1: Sample of Raw Data.

	Open	High	Low	Close
count	1556.000000	1556.000000	1556.000000	1556.00000
mean	582.625328	597.992847	567.851446	584.239396
std	523.137312	542.992855	505.877401	525.904442
min	68.500000	74.560000	65.530000	68.430000
25%	254.287500	260.327500	248.835000	254.320000
50%	438.600000	447.560000	430.570000	438.855000
75%	662.437500	674.525000	646.735000	663.402500
max	2953.220000	2999.910000	2840.530000	2958.110000

Table 2. Description of Raw Data.

```
Date        object
Open       float64
High       float64
Low        float64
Close      float64
Volume      object
Market Cap  object
dtype: object
```

Table 3: Datatype of Raw Data.

	Date	Open	High	Low	Close	Volume	Market Cap
0	Jul 31, 2017	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000
1	Jul 30, 2017	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000
2	Jul 29, 2017	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000
3	Jul 28, 2017	2679.73	2897.45	2679.73	2809.01	1,380,100,000	44,144,400,000
4	Jul 27, 2017	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000

Figure 2: Raw Sample Data

In above Figure 3 date have object datatype, but date feature must support something known as timestamp nature because it is the must condition for a Time Series case so firstly convert it to datetime format and then will sort the data according to Date. Then make Date as index feature to avoid key Error.
Now, when it comes to predicting, there are two basic types of forecasting: quantitative and qualitative forecasting.

1. Quantitative forecasting makes use of data that can be measured. It makes use of previous data that is both dependable and accurate, which removes the possibility of forecasting inaccuracy and bias.

2. Qualitative forecasting relies on non-quantifiable data. It is excellent for new firms that do not have any or much historical data because it is based on views and expert recommendations. Qualitative approaches are beneficial, but it's critical to consider the data in a nonjudgmental and unbiased manner.

Because historical data is accessible, this research will concentrate on quantitative forecasting.

```
Date           object
Open          float64
High          float64
Low           float64
Close         float64
Volume         object
Market Cap     object
dtype: object
```

Figure 3: Datatype of different column of raw data.

The four techniques to quantitative forecasting are trend projection, the nave

approach, moving averages, and exponential smoothing. We will use all of the ways to improve forecast accuracy and will continue to expand our model by using a few additional approaches to improve performance.

Trend projection: One of the most common types of business forecasting is trend projection. It's quite straightforward to comprehend, since it analyses prior trends in a time-series and predicts that the same trends will occur in the future.

1. a) Now there must performed exploratory data analysis considering the "Close" feature the end goal is to predict what is the closing price of Bitcoin, so a copy of data in 'Data' will be created and analyze the close feature deeply to understand it is trend. To under- stand it is trend line plot will be used and get Figure 4.

From Figure 5-line plot there can be seen in 2014 there is a spike in closing price of Bitcoin and then in 2018 again there is a spike. b) Now, resample the data in a particular date range and after resampling the sum of closing price is found, for average closing price the mean function will be used and use various such functions and plot all these values with respect to date feature in Figure 6. In Figure 6 plot sum on yearly basis.

In Figure 7 there can see the exact closing price with respect to different year using mean function. Similarly, it can be applied to distinct functions on yearly, monthly, quarterly bases etc.

3: a) Analyze average weekly closing price: for this group by 'Close' feature of data will be used and take mean according to week and plot them in Figure 8. b) Analyze average closing price by day: From this, find the average closing price per day and plot in Figure 9, similarly analyze the trend according to the quarter and plot it as in Fig. 10.

2. Analyzing the trend of closing price in Weekdays & weekends: for this create a function in which from weekofdays function if day<5, then it will be considered it as weekday else weekend as in Figure. 11 and plot in Figure. 12 and from this it can be seen a minor difference in two graphs.

5: Now build Baseline Model or Naive Model and using this model prediction will be performed:

The naive method takes into account what happened in the previous period and predicts that the same thing will happen again. The sole basis for Naive forecasting models is historical observation. They don't try to explain the underlying causal links that result in

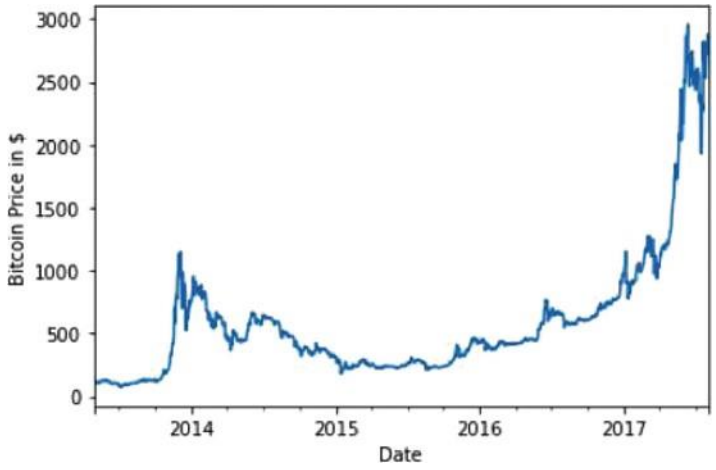


Figure 4: Close Feature Plot.

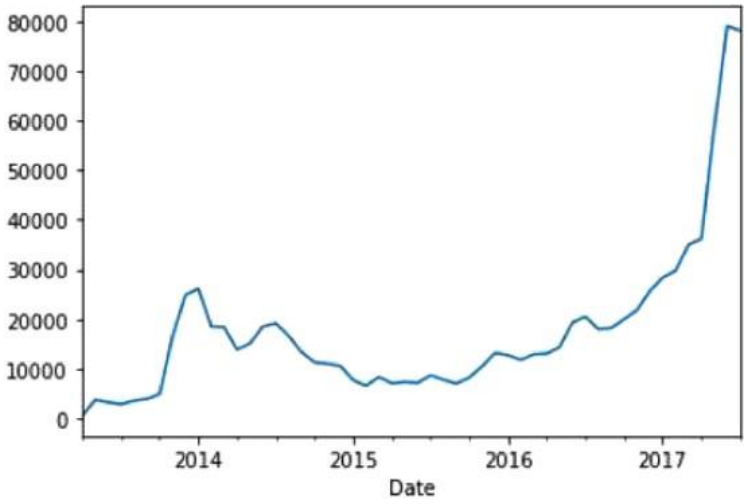


Figure 5: Plot resample data sum on weekly basis.

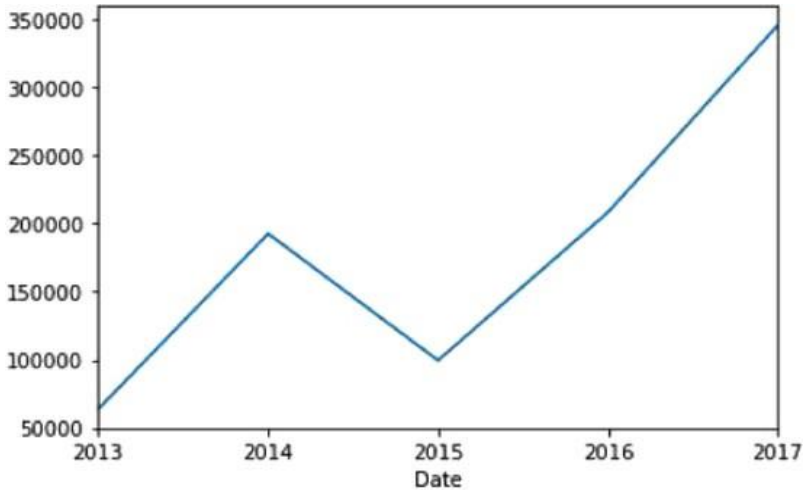


Figure 6: Resample of Data on Yearly Basis.

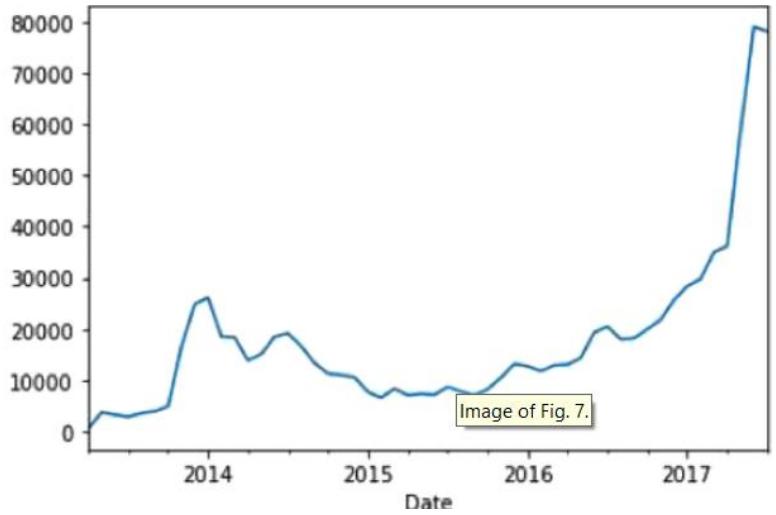


Figure 7: Mean Closing Price in Different Year

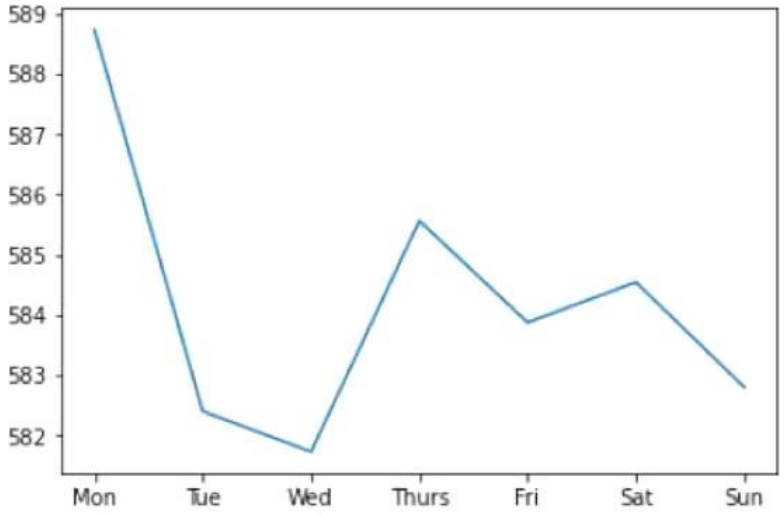


Figure 8: Mean of 'Close' by week.

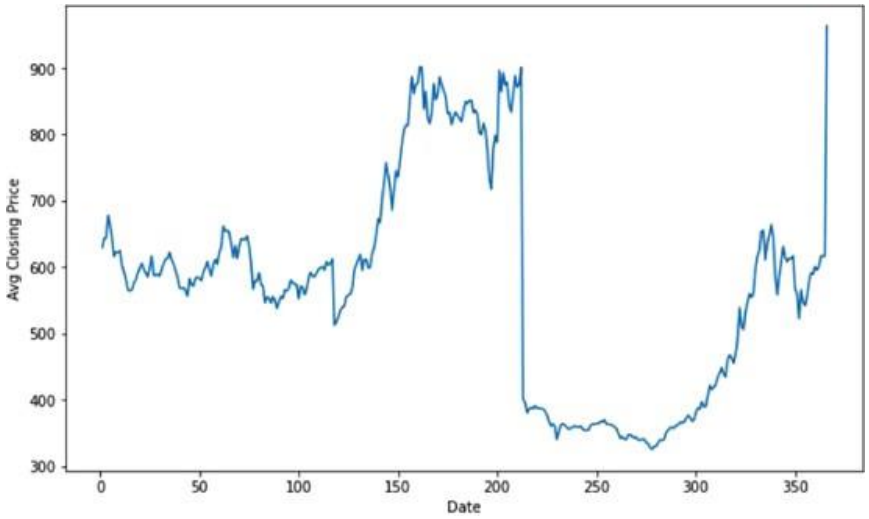


Figure 9: Average Closing Price Each Day.

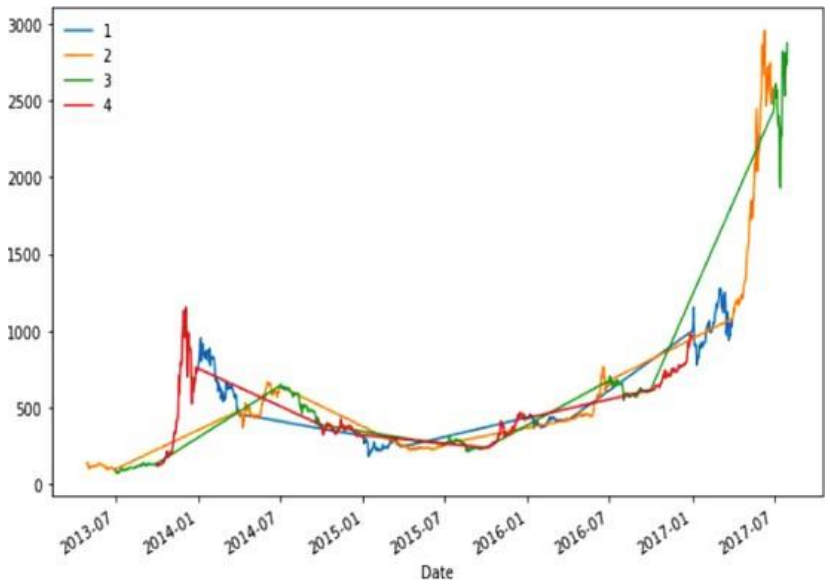


Figure 10. Average Closing Price Each Day

the forecasted variable. This model is all about "the previous value is the best reflector of the next Value " or we can say that next value completely depends on previous value.
A simple example of a naive type:

1. Use the actual sales of the current period as the forecast for the next period. Let us the symbol

$$Y'_{t+1} = Y_t$$

Y'_{t+1} as the forecast value and the symbol Y_t as the actual value. Then

2. If you consider trends, then

$$Y'_{t+1} = Y_t + (Y_t - Y_{t-1})$$

Date	Open	High	Low	Close	Volume	Market Cap	dayofweek
2017-07-30	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000	weekends
2017-07-29	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000	weekends
2017-07-23	2808.10	2832.18	2653.94	2730.40	1,072,840,000	46,231,100,000	weekends
2017-07-22	2668.63	2862.42	2657.71	2810.12	1,177,130,000	43,929,600,000	weekends
2017-07-16	1991.98	2058.77	1843.03	1929.82	1,182,870,000	32,767,600,000	weekends

Figure 11. Weekdays and Weekends.

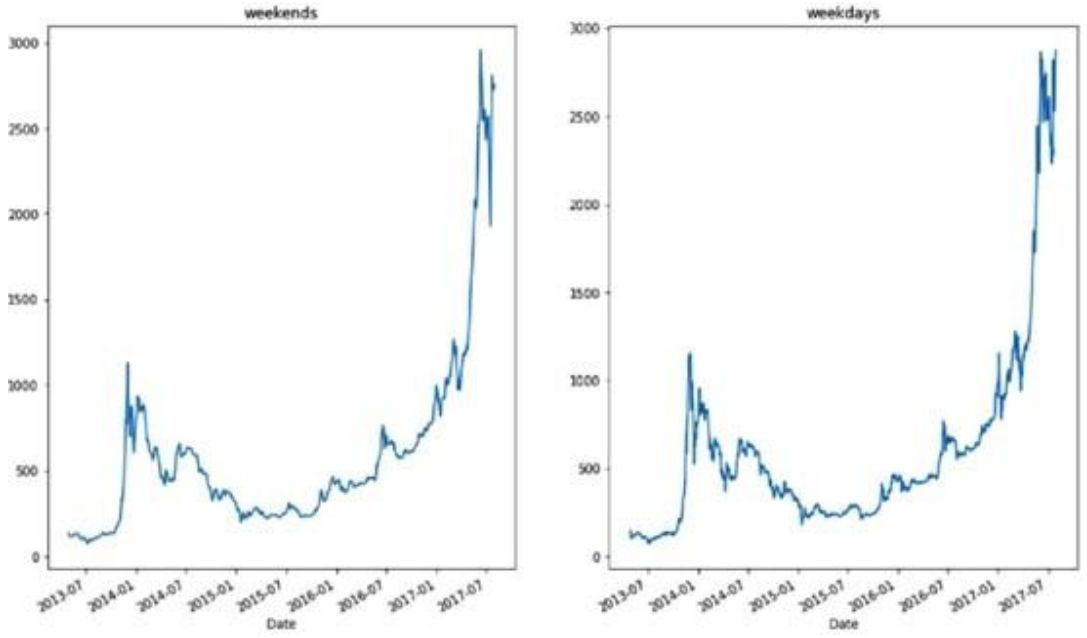


Figure 12. Weekdays and Weekends Plot.

This model adds the latest observed absolute period-to-period change to the most recent observed level of the variable.

BaseLine model or Naive Model: This model is all about "the previous value is the best reflector of the next Value" or it can be stated that next value completely depends on previous value. This is the basic summary behind this baseline model. For this it will use inbuilt function of pandas' library using Shift operations on data and store this data in "prediction_naive" in Figure 13.

Now plot the different between my actual value and my predictive trend value in Figure 14.

From the plot Figure 14 this can be seen that the prediction not much differ from actual values, so it almost overlaps the actual value graph.

Date	Open	High	Low	Close	Volume	Market Cap	dayofweek	prediction_naive
2017-07-31	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000	weekdays	NaN
2017-07-30	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000	weekends	2875.34
2017-07-29	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000	weekends	2757.18
2017-07-28	2679.73	2897.45	2679.73	2809.01	1,380,100,000	44,144,400,000	weekdays	2726.45
2017-07-27	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000	weekdays	2809.01

Figure 13. Naïve Prediction.

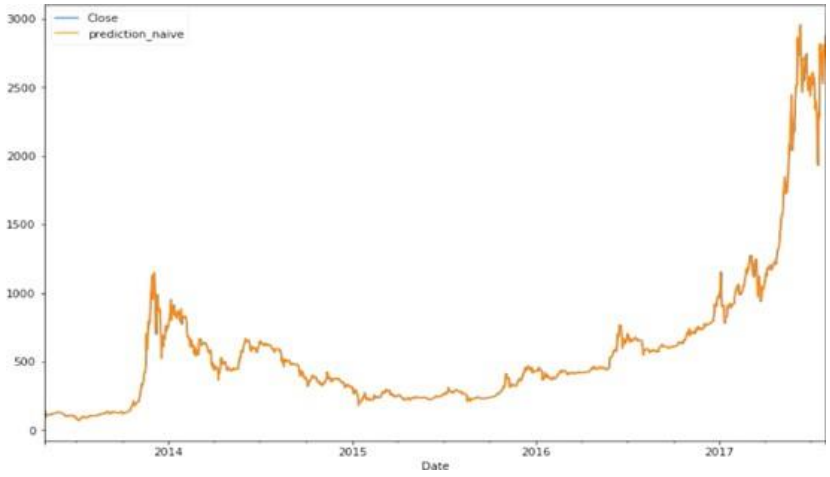


Figure 14. Naïve Prediction Vs Actual Value Plot.

Now to check how exact model is performing, use sklearn to find the mean_square_error (ignoring NaN values). An error rate of 37.23363264835875 will be got which means that there is a difference of approx. +−37.2 in prediction and actual values. This is a good model but still can't stay with this model as in real world use cases because there might have seasonality, data might not be smooth, that previous data is not true reflector of future data. So it can't considered as baseline model, even it have a good performance.

6: Examine if there exists seasonality in the data or not: When the mean and the variance of the data is constant that is basically stationary nature of data and for a time series model data must be stationary and if it is not stationary then make it stationary so that can apply other advance algorithms like SARIMA which can give a better prediction result. So apply Rolling(moving Average) on the data at window

period of 7 and compute mean and standard deviation to eliminate the seasonality of curve in Figure 4 and plot the new curve as in Figure 15.
In above graph the orange one is exactly the given series and blue which get overlapped here on the green curve is mean values. From this, it has been computed that rolling mean is not stationary and is varying with time. So now must remove its seasonality also and make it stationary. But firstly, lets prove that there is some seasonality in data for which use some statistical approach which is Adfuller Test.
The Dickey-Fuller test is a unit root test that examines the null hypothesis that $\alpha = 1$ in the following model equation. alpha is the coefficient of the first lag on Y. Null Hypothesis $(H0)$: $alpha = 1$

$$y_t = c + \beta t + \alpha y_{t-1} + \Phi \Delta Y_{t-1} + e_t$$

where, $y(t-1) = lag\ 1$ of time series delta $Y(t-1) = first$ difference of the series at time $(t-1)$

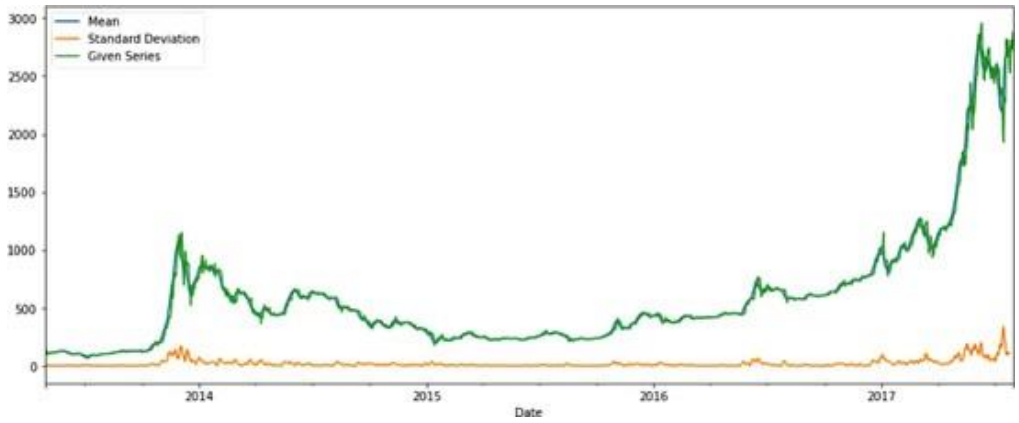


Figure 15: Mean and Standard Deviation Plot.

Fundamentally, it has a similar null hypothesis as the unit root test. That is, the coefficient of $Y(t-1)$ is 1, implying the presence of a unit root. If not rejected, the series is taken to be non-stationary.
The Augmented Dickey-Fuller test evolved based on the above equation and is one of the most common form of Unit Root test. Adfuller test is mostly depends on one parameter, p-value. There are two types of Hypotheses:

1. Null Hypothesis is represented by H0

2. Alternate Hypothesis which denoted by H1

There will be 2 case either the p-value is greater than 0.05 (Strong evidence against

null hypothesis or can reject null hypothesis and data is stationary) or less than 0.05 (weak evidence against null hypothesis, hence data is not stationary).

Lets consider an simple example: We have data which have mean as: $\mu = 120$ (H0 Null hypothesis), i.e. alternative hypothesis $H\alpha$: $\mu > 120$ or $\mu < 120$, and assuming that $\alpha(level\ of\ sig.) = 0.05$. The sample values that u took are as $n(sample\_size) = 40$, $\sigma = 32.17$ and $x(sampling\_mean) = 105.37$. What is the conclusion for this hypothesis, i.e. what about P-value?
We know that,
$\sigma\bar{x} = \sigma\sqrt{n}$
Now substitute the given values
$\sigma\bar{x} = 32.17\sqrt{40} = 5.08$
As per the test static formula, we get
$t = (105.37 - 120) / 5.0865 = -2.8762$
Now we have to Use Z-Score table, finding the value of $P\ (t > -2.8762)$ *we get,*
$P\ (t < -2.8762) = P(t > 2.8762) = 0.003$
Therefore,
If $P(t > -2.8762) = 1 - 0.003 = 0.997$
$P - value = 0.997$ which is > 0.05
As value of p > 0.05, we are failed to reject Null hypothesis
Therefore, the null hypothesis is accepted, i.e. The
result is not statistically significant The Null &
Alternate hypothesis regarding this use-case is
Null Hypo->> closing price is stationary in nature
Alt. Hypo->> closing price is not stationary in nature

There will be two cases 1. $P - value > 0.05$ 2. $P - value < 0.05$. First case signifies null hypothesis can be rejected as there is convincing evidence against it. Second case signifies the acceptance of hypothesis as there is unconvincing evidence in its against hence data is not stationary. From statsmodel library directly import Adfuller and use it in program on data of "close". From this, $p - value = 0.0002154535155876224$ which is less than 0.05 so null hypothesis rejected, hence data is stationary. There is still some seasonality and outliers in the data, which may be reduced by applying Log Transformation on the data. The Log Transformation is used to remove outliers from data that are either very high or very low. There is still some seasonality in data now try to eliminate this by using Log Transformation on data. Log Transformation is used for removing Either extremely High or exceptionally Low outliers from the data, use log function which is available in NumPy for this and plot it as in Figure 16 after removing seasonality from data.
Now, Smoothening and moving average (M.A) have to be performed on data. In Smoothening, rather than just getting an average and using it as the next forecast, it

decreasingly weights exponents depending on outside factors, like season or age of a product. This is done to 'smooth' the averages and create a reliable forecast. Smoothening is done by using moving average. Moving Average is universally used in financial market. Rolling is a window that have considered for moving average (M. A). Now plot this Rolling Trans- formation as Figure 17.

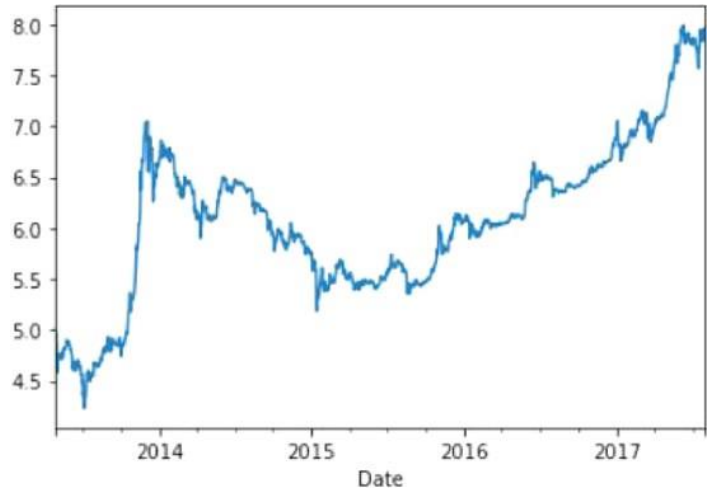


Figure. 16. After Removing Seasonality Factor.

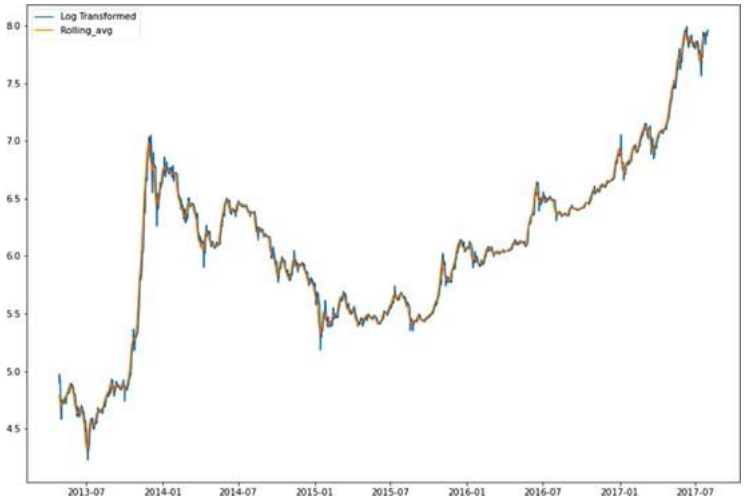


Figure 17: Log Transformation and Moving Average.

Now can compute the difference between log_data and rolling_average and checks its stationarity also as in Figure. 18.

From Above Graph there are convincing results and proof against null hypothesis, so null hypothesis will get rejected hence data is Stationary. From the series, time series is approx. Stationary with constant interval. Now apply differencing using shift and

plot it as in Figure 19 below.
With Respect to seasonality there is the trend as shown in Figure 20. Now if wants to check its stationarity and want to check whether have made some Seasonality or not, similarly check it for this and then plot it as in Figure 20.
From this it can be concluded that dicky fuller Test is very much less than 1% critical Value. Now, prepared data for forecasting using FbProphet.

Forecast using prophet model and plot forecasting

"Facebook Prophet" is a Facebook-developed open-source toolkit for univariate time series forecasting that use a Bayesian-based curve fitting approach to forecast time series data. Prophet is unique in that it requires no prior knowledge or expertise in forecasting

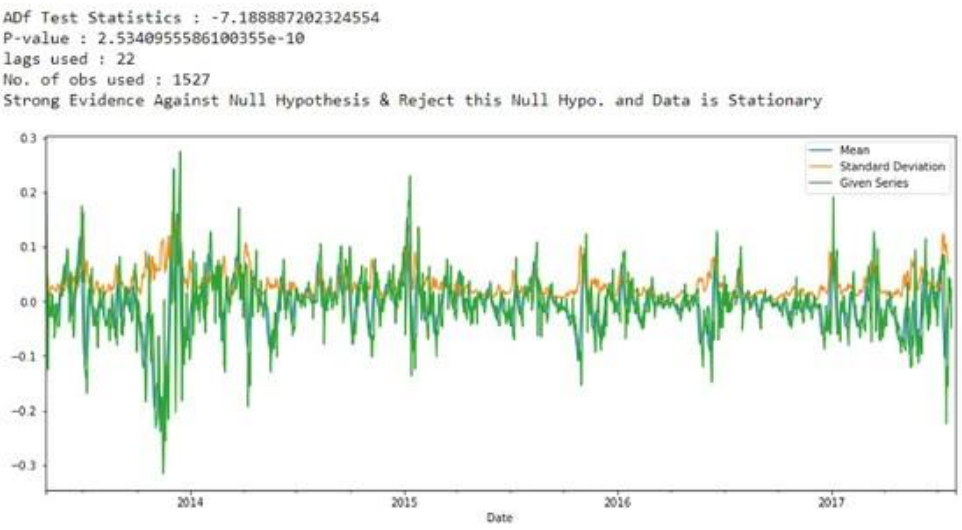


Figure 18: Rolling and Moving Average Difference is Stationary.

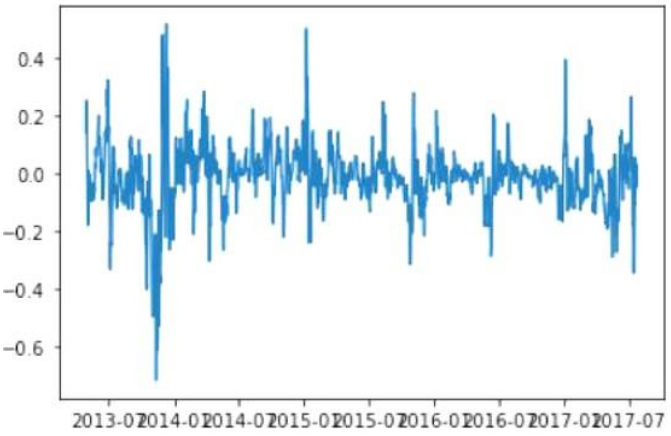


Figure 19. Differencing

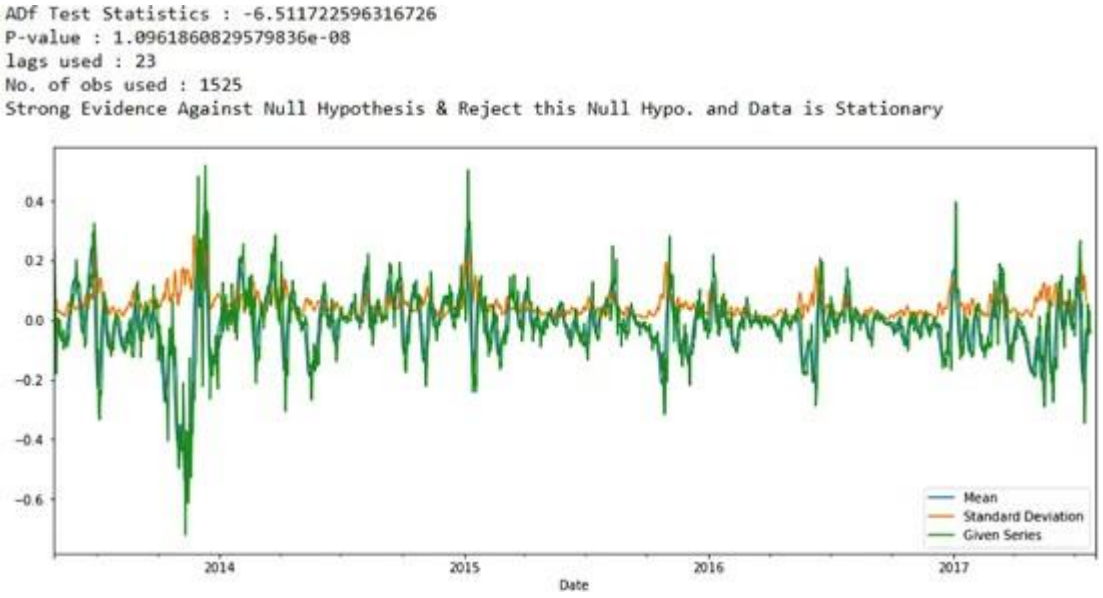


Figure 20. ADf Test Statistics

time series data since it automatically detects seasonal trends underlying the data and provides a set of 'simple to comprehend' parameters. Prophet is also built to deal with holidays that are known ahead of time, missing data, and significant outliers. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

The package employs an easily interpreted, three components additive model whose Bayesian posterior is sampled using *STAN*. In contrast to some other approaches, the user of Prophet might hope for good performance without tweaking a lot of parameters. In-stead, hyper-parameters control how likely those parameters are a priori, and the Bayesian sampling tries to sort things out when data arrives. Prophet's default settings produce forecasts that are often [as] accurate as those produced by

skilled forecasters, with much less effort.

Prophet is a model that is exactly available in FbProphet library. It manages irregular intervals or irregular holidays. If there are some noises in data or are some outliers in data, then that scenario also gets managed by this Fb prophet module. Adjusting with non- linear patterns may be historical or seasonal or weekly, a time series model that is prophet applies additive model. It has capability to deal with outliers and missing values.

Before fitting prophet model there is need to prepare data according to prophet model documentation. It must make sure that data will follows all its protocols, and its protocols are as follows. Make date as "ds" and the output feature as "y"'.

Now fit the model for a period of 500 Days with frequency of Day. The forecast value will look like as in Figure 21.

Where yhat is actual prediction and yhat_upper is prediction for upper bound and yhat_lower is prediction for lower bound. Now to plot this forecast for which use inbuild feature of Fbprophet library which is forecast and plot it using plot function as in Figure 22.

In the graph the black dot are actual data, blue line is prediction curve and light blue line is trend. Similarly, can plot for weekly basis, yearly basis, monthly basis, etc. as in Figure 24.

	ds	yhat	yhat_lower	yhat_upper
2046	2018-12-04	5587.349564	3155.666046	8010.214200
2047	2018-12-05	5586.897199	3145.849320	8052.950010
2048	2018-12-06	5590.384867	3085.462484	8043.696604
2049	2018-12-07	5587.862758	3103.062760	8078.466456
2050	2018-12-08	5587.249932	3133.705219	8141.700373
2051	2018-12-09	5586.850155	3069.009790	8076.865174
2052	2018-12-10	5590.897150	3082.779889	8047.689207
2053	2018-12-11	5591.548155	3089.374667	8102.190623
2054	2018-12-12	5588.748898	3069.402777	8070.284320
2055	2018-12-13	5590.484927	3030.567550	8169.309468

Figure 21: Forecast Values.

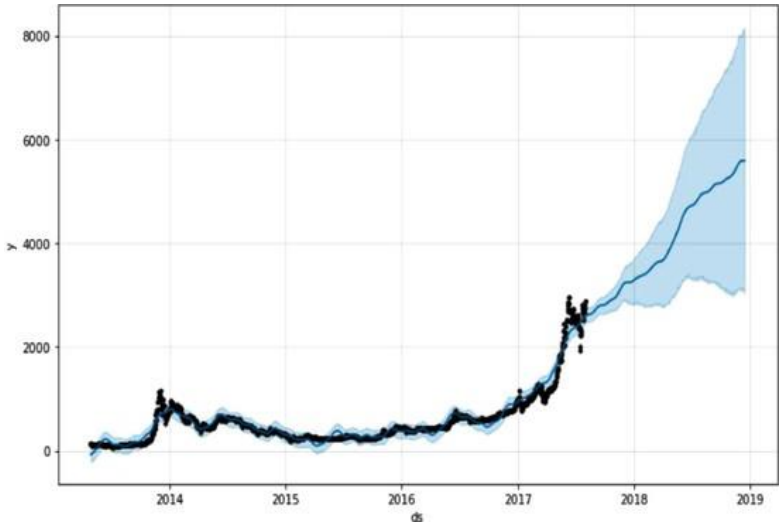


Figure 22: Forecast using FbProphet.

Now to cross validate forecast model for which calculate forecast error. To compute forecast error will be compared to actual values to predicted values. In Fbprophet there is an in build cross validation techniques. Inside this cross-validation technique. There are some parameters which are Horizon parameter, size of Initial training period, spacing between cutoff period and plot it as in Figure 23. In Figure 23 plot for root-mean-square error, similarly, can plot for mean-square error and further check for errors.

9.5 Results and analysis

This project is created on Jupyter Notebook (anaconda 3) using python. The data for this study came from an open-access website - https://www.kaggle.com/team-ai/bitcoin-price-prediction/version/1. Used data is real-world or actual historical data to form Naïve model and to search for historical trends. Data can be use at real time using Big data concepts like Spark Streaming and Kafka. Using these big data tools we can collect data at real time and analyze them and apply this model at real time. Data is made up of a single.csv file which consists of date, open, high, low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. For real time collection of Data, we can collect data from https://in.tradingview.com/chart/?symbol=COINBASE%3ABTCUSD from where data can easily exported to big data tools and it will be easy to apply this model at real time. In this section the output of Naïve prediction is shown in Table 4 and then plot this Naïve model prediction in Figure 25.

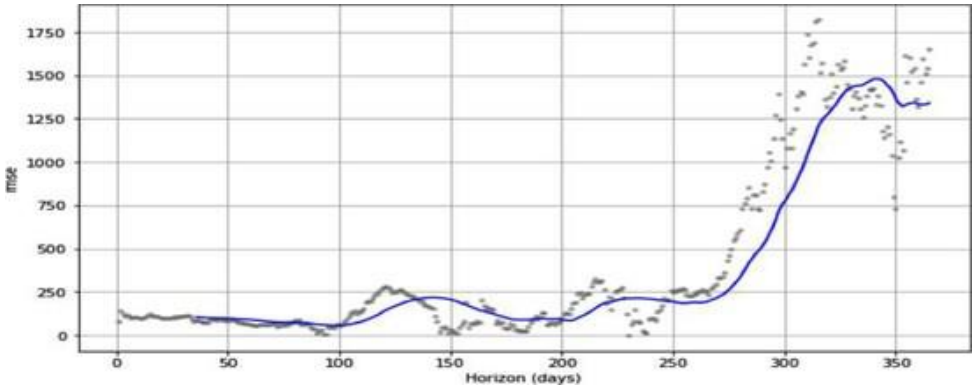


Figure 23: Root_Mean_Square Error.

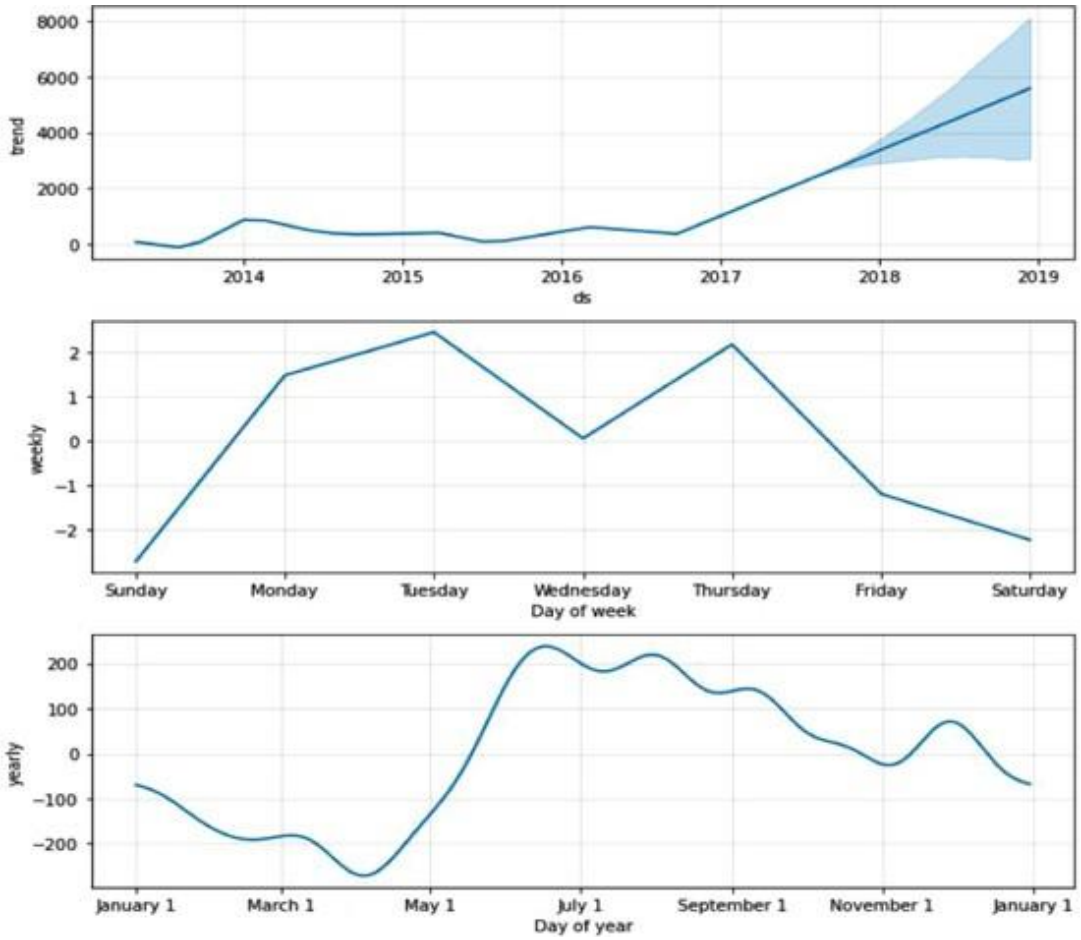


Figure 24: Weekly, yearly and monthly plot.

As Naïve model was based on historical data, and it is not possible to predict the future on bases of historical data in real world. So created a new model with help of Fbprophet library, forecasted it in Table 5 and plot it in Figure 26. After prediction cross validated model and find the root_mean_square error in model and plot it in Figure 27.
Figure 27 shows the difference between actual and predicted valued for datasets. In Figure 26 the black dots show actual data, blue line shows the predicted curve and blue shaded region shows the trend of Bitcoin.
As Naïve model was based on historical data and it is not possible to predict the future on bases of historical data in real world. The historical data might have seasonality, so we checked for seasonality using AdFuller and remove seasonality. Figure 27 shows seasonality present in data. Figure 28 shows the naïve model after removing seasonality. After removing seasonality, there might be some utliers so to eliminate those outliers log transformation is applied on data and there is a need to make data smooth using Rolling transformation, Figure 29 shows data after applying log transformation and rolling.

Date	Open	High	Low	Close	Volume	Market Cap	dayofweek	Prediction_naive
2017–07–31	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000	weekdays	NaN
2017–07–30	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000	weekends	2875.34
2017–07–29	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000	weekends	2757.18
2017–07–28	2679.73	2897.45	2679.73	2809.01	1380,100,000	44,144,400,000	weekdays	2726.45
2017–07–27	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000	weekdays	2809.01

Table 4: Naïve Prediction Table.

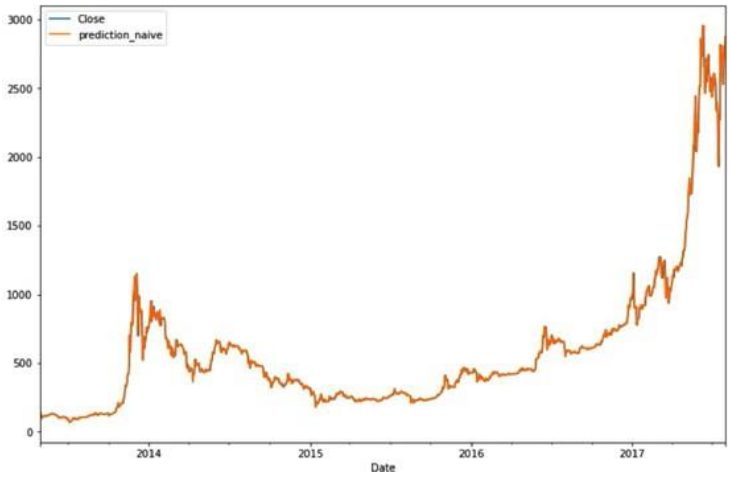


Figure 25. Naïve Prediction Plot.

	ds	yhat	yhat_lower	yhat_upper
2046	2018–12–04	5587.349564	3155.666046	8010.214200
2047	2018–12–05	5586.897199	3145.849320	8052.950010
2048	2018–12–06	5590.384867	3085.462484	8043.696604
2049	2018–12–07	5587.862758	3103.062760	8078.466456
2050	2018–12–08	5587.249932	3133.70529	8141.700373
2051	2018–12–09	5586.850155	3069.009790	8076.865174
2052	2018–12–10	5590.897150	3082.779889	8047.689207
2053	2018–12–11	5591.548155	3089.374667	8102.190623
2054	2018–12–12	5588.748898	3069.402777	8070.284320
2055	2018–12–13	5590.484927	3030.567550	8169.309468

Table 5: FbProphet Forecast Table.

Now we further check for seasonality in our processed data. From Figure 28 we can say our data is Stationary. Now finally, we can create a time series model using Facebook Prophet. Forecasted values are in Table 4 and plot it in Figure 26. After prediction there is a need of cross validate for errors, so we cross validate model and find the root_mean_square error in model and plot it in Figure 27.
Figure 27 shows the difference between actual and predicted valued for datasets. In Figure 26 the black dots shows actual data, blue line shows the predicted curve and blue shaded region shows the trend of Bitcoin.

9.6 Discussion

This study's proposed model for cryptocurrency prediction can be recognized a consistent and suitable model. Previous models such as BTC rates 43 percent. Whereas model based on LTC Multi-linear regression as R2 rates percent of 43. Whereas its precision for Bitcoin is 57 percent. The study shows Bitcoin logistic regression rates 67 whereas LDA have 64 percent accuracy. Linear discriminant analysis as same like bitcoin regression 67 percent. Models occurred in existence before such as LTC Multi-linear regression model R2 have 42 percent for LTC. But have accuracy of 56 percent for Bitcoin. Bitcoin Logistic regression and linear discriminant analysis rates LR have 65 percent. Methods are often based on previous data, and in real-world issues, future outcomes cannot be anticipated only on the basis of historical data. Seasonality in historical data may exist, or pattern accuracy in other models may be impaired. But in this model here created a function for removing seasonality and used advanced modules like Fbprophet which are one of the best

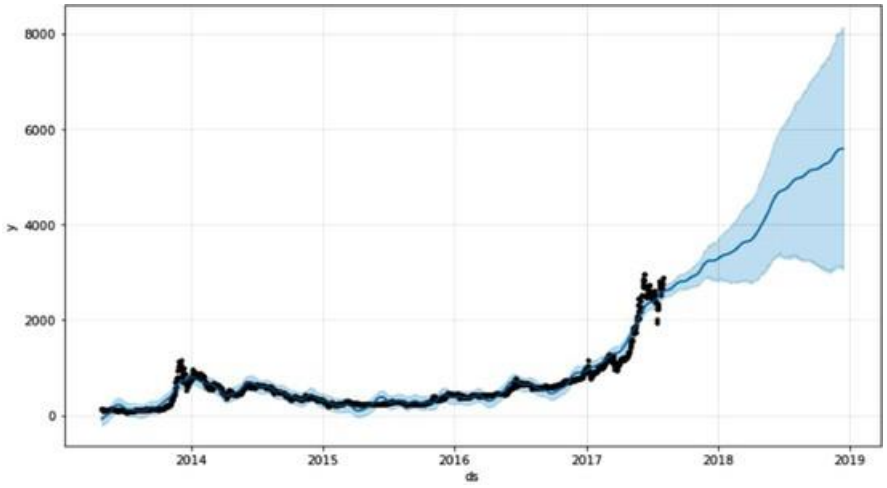


Figure 26. Fbprophet prediction plot.

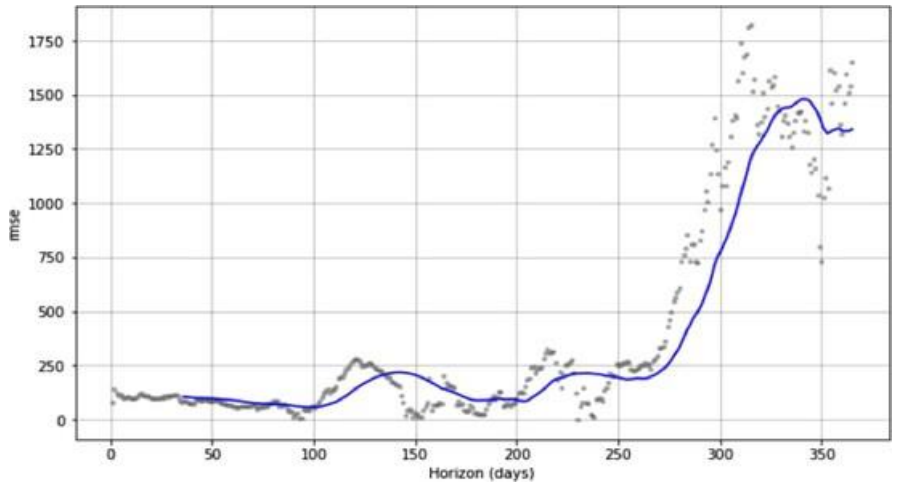


Figure 27. Cross validation using Root_Mean_Square Error.

modules for real time data and time series model and at last cross-validated the model in Figure 27 and can clearly see accuracy of this model is far better than other models. ARIMA is a strong model that, as we've seen, produced the greatest results for stock data. One difficulty is that it may need meticulous hyperparameter adjustment and a thorough comprehension of the data. When comparing ARIMA to LSTM and Fbprophet Models in the Stock Market, it can produce superior results. However, when it comes to cryptocurrencies, where there is severe seasonality, ARIMA can handle data with trend but does not support time series with trend and seasonality. Unlike ARIMA and Prophet, LSTM does not require on certain assumptions about the data, such as time series stationarity or the presence of a Date field. One problem is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior might be difficult. In order to produce decent outcomes, rigorous hyperparameter adjustment is also essential. So there was a need in the case of bitcoin;

a model that could dynamically track changes in trends and data, manage severe seasonality, and handle a large number of outliers and changes in trends owing to market occurrences was required. Unlike some other techniques, the user of Prophet may expect decent results without adjusting a lot of settings. Instead, hyper-parameters determine how likely certain parameters are a priori, and Bayesian sampling attempts to sort things out once the data is collected. It works best with time series with substantial seasonal influences and historical data from several seasons. However, when there is a lot of seasonality and seasonality patterns in a dataset, and these patterns aren't consistent or smooth, and trends change quickly, it can hurt the model's performance. To address this and other limitations of Fbprophet, we used quantitative forecasting on the data to get stationary data that is free of outliers, consistent, and smooth. Now, data may be used to use the Prophet time series model to provide a reliable forecast.

9.7 Conclusion

Cryptocurrencies are volatile, trends are dynamic, data is neither consistent nor smooth, there is a lot of seasonality in data, and trends in Cryptocurrencies can't be totally based on previous data since they might change dynamically. Majorly two machine learning algorithms are proposed and implemented for forecasting Bitcoin values in this article. The accuracy of several models was evaluated using performance metrics, as illustrated in Figure 27. The methodology included in the sudy majorly focuses on fbprophet model

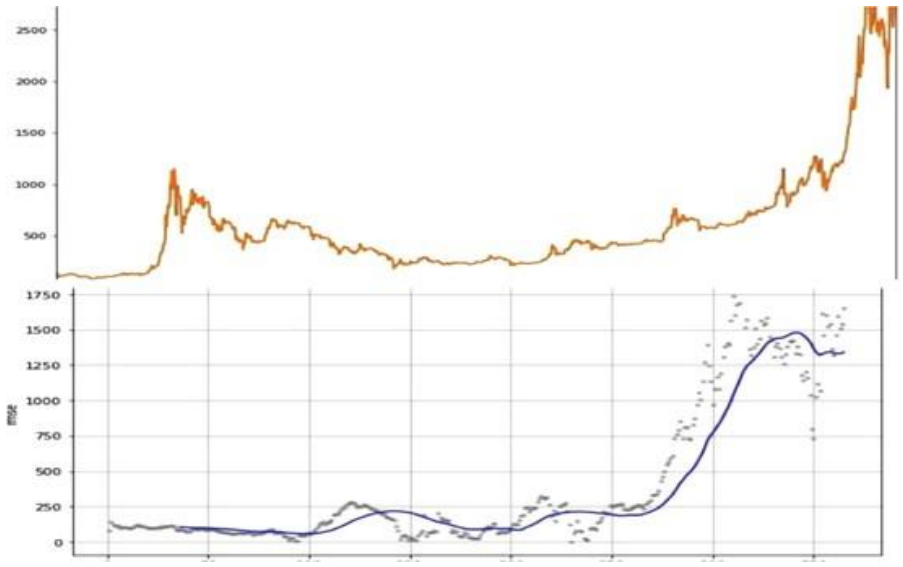


Figure 28: Naïve and Fb Prophet model curve comparision.

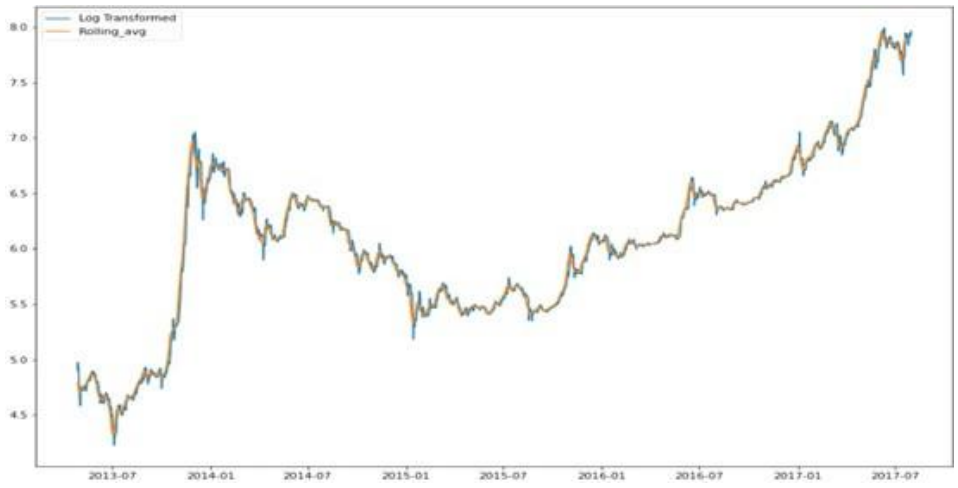


Figure 29: Data After Log transformation.

applied for Cryptocurrency. The outcome of the methodology in the work shows that in the case of crypto currencies, time series patterns are complex and vary dynamically over time, but Prophet only tracks such changes when the trend shifts. Prophet is forgiving of missing data and trend changes, and it usually handles outliers well. Prophet is built to deal with holidays that are known ahead of time, missing observations, and huge outliers. The key aspect of the outcome is that, because cryptocurrency patterns are not consistent and fluctuate over time, Facebook's Prophet is the best solution in this situation. Prophet is unique in that it does not require much prior knowledge or expertise with time series data forecasting since it automatically detects seasonal trends underlying the data and provides a set of "simple to understand" parameters, therefore utilized Fbprophet to predict real-world outcomes for cryptocurrencies after eliminating the seasonality effect and smoothening of data.

The real and forecasted prices were then compared. The precision of the model, as shown in Figure 27, is significantly superior to that of previous models. Based on these findings, discovered that the Naive model is only useful with historical data, and while its accuracy is excellent, still cannot rely on past data completely for real-world scenarios, therefore utilized Fbprophet to predict real-world outcomes after eliminating the seasonality effect. The results of the conducted experiments signify that according to the trial findings: 1. The AI algorithm is accurate and suitable for bitcoin prediction 2. Fbprophet Model is better than Naïve model for real world test cases.

Fbprophet Model can give better result when data is consistence, stationary and smooth therefore there is need for applying Quantitative forecasting on data before Fbprophet.

In future research, it can be looked into other factors that could influence

cryptocurrency prices. Model can be further evolved by applying Qualitative forecasting on data. Unlike Quantitative methods, Qualitative forecasting uses data that can't be measured. It relies on opinions and expert advice and is useful for new companies that don't have any or much historical data. Qualitative methods are useful, but it's important to take the information into account in a nonjudgmental and unbiased manner. In future research, it can be investigated other factors that could influence cryptocurrency prices, with a particular focus on the impact that specifically on tweets by applying the concept of natural language processing and sentiment analysis. There are only few API's are available because storage and analysis the trading sites have to provide the real time data and generally companies do not provide and this data need to be processed in real time. For real time analysis, we can use Big data technologies like Kafka and Spark Streaming.

Conclusions

At the end of this dissertation it is important to point out how my own research activity has been essentially focused on the concept of "Dynamic Modeling".
In particular, there were introduced three lines of research - different from each other in the mathematical formulation but closely linked to the concept of dynamics and the role that uncertainty plays within each problem - described and explored in the various original works arranged into this work.
The most crucial aspects of dynamics have been modeled by using of mathematical tools that fall within the context of dynamic systems with the important novelty of the introduction of a fuzzy measure aimed at broadening the spectrum of convergence of the solutions of the systems proposed for the analysis of the various problems covered by the case studies of this work.
In many practical contexts, such as those analyzed within the thesis, there are inputs that act on the real system of interest with characteristics of irregularity, complexity, uncertainty, non-reproducibility and unpredictability. In describing these signals, it is easy to adopt the paradigm of stochastic processes. Furthermore, in a dynamic and fast world like today's one, a “cyclical” approach is increasingly necessary to manage the various decision-making processes. In fact, each data analysis, even the most careful and scrupulous, if it is not updated risks becoming obsolete in a short time, especially in those contexts in which decisions must be made and strategies implemented in real (or almost) time.
Therefore, traditional models are not enough to model and solve these problems - as repeatedly reiterated in the present study - but new techniques and algorithms of Artificial Intelligence are needed, in particular of Machine Learning and Deep Learning to maintain high performance analytics of predictive models.
Therefore, Dynamical Systems, Stochastic Programming and Artificial Intelligence represented a horizontal combination in which it is possible to plan a scientifically robust forecasting analysis and prediction activity with a good scientific result in line with the complexity of our times.
The pandemic phenomenon has revolutionized the traditional paradigms of classic decision-making platforms.
In this thesis, on the basis of a series of reflections, under the guidance of my Supervisor Massimiliano Ferrara, by the publication of the original results obtained, we wanted to add milestones to a potential discussion to be started in the coming years in the international scientific community.

References Chapter 1

[1] Podlubny I., (1999). Fractional Differential Equations, Mathematics in Science and Engineering, Academic Press, New York.

[2] Lakshmikantham V., Leela S., Vasundhara J. (2009). Theory of Fractional Dynamic Systems, Cambridge Academic Publishers, Cambridge, UK, 2009.

[3] Hilfer, R. (2000). Fractional calculus and regular variation in thermodynamics. In *Applications of fractional calculus in physics* (pp. 429-463).

[4] Rossikhin, Y. A., & Shitikova, M. V. (1997). Applications of fractional calculus to dynamic problems of linear and nonlinear hereditary mechanics of solids.

[5] Seadawy, A. R., & El-Rashidy, K. (2018). Dispersive solitary wave solutions of Kadomtsev-Petviashvili and modified Kadomtsev-Petviashvili dynamical equations in unmagnetized dust plasma. *Results in Physics*, *8*, 1216-1222.

[6] Arshad, M., Seadawy, A. R., & Lu, D. (2017). Modulation stability and optical soliton solutions of nonlinear Schrödinger equation with higher order dispersion and nonlinear terms and its applications. *Superlattices and Microstructures*, *112*, 422-434.

[7] El-Nabulsi, R. A. (2017). Induced gravity from two occurrences of actions. *The European Physical Journal Plus*, *132*(7), 1-8.

[8] Lu, D., Seadawy, A. R., & Ali, A. (2018). Applications of exact traveling wave solutions of modified Liouville and the symmetric regularized long wave equations via two new techniques. *Results in Physics*, *9*, 1403-1410.

[9] Helal, M. A., Seadawy, A. R., & Zekry, M. H. (2014). Stability analysis of solitary wave solutions for the fourth-order nonlinear Boussinesq water wave equation. *Applied Mathematics and Computation*, *232*, 1094-1103.

[10] Lu, D., Seadawy, A. R., & Ali, A. (2018). Dispersive traveling wave solutions of the equal-width and modified equal-width equations via mathematical methods and its applications. *Results in Physics*, *9*, 313-320.

[11] Samko, S. G., Kilbas, A. A., & Marichev, O. I. (1993). *Fractional integrals and derivatives* (Vol. 1). Yverdon-les-Bains, Switzerland: Gordon and breach science publishers, Yverdon.

[12] Toledo-Hernandez, R., Rico-Ramirez, V., Iglesias-Silva, G. A., & Diwekar, U. M. (2014). A fractional calculus approach to the dynamic optimization of biological reactive systems. Part I: Fractional models for biological reactions. *Chemical Engineering Science*, *117*, 217-228.

[13] Toledo-Hernandez, R., Rico-Ramirez, V., Iglesias-Silva, G. A., & Diwekar, U. M. (2014). A fractional calculus approach to the dynamic optimization of biological reactive systems. Part I: Fractional models for biological reactions. *Chemical Engineering Science*, *117*, 217-228.

[14] Kilbas, A. A., Srivastava, H. M., & Trujillo, J. J. (2006). *Theory and applications of fractional differential equations* (Vol. 204). elsevier.

[15] Wang, G., Yang, Z., Zhang, L., & Baleanu, D. (2020). Radial solutions of a nonlinear k-Hessian system involving a nonlinear operator. *Communications in Nonlinear Science and Numerical Simulation*, *91*, 105396.

[16] Wang, G., & Ren, X. (2020). Radial symmetry of standing waves for nonlinear fractional Laplacian Hardy–Schrödinger systems. *Applied Mathematics Letters*, *110*, 106560.

[17] Zhang, L., & Hou, W. (2020). Standing waves of nonlinear fractional p-Laplacian Schrödinger equation involving logarithmic nonlinearity. *Applied Mathematics Letters*, *102*, 106149.

[18] Wang, G., Ren, X., Bai, Z., & Hou, W. (2019). Radial symmetry of standing waves for nonlinear fractional Hardy–Schrödinger equation. *Applied Mathematics Letters*, *96*, 131-137.

[19] Al-Refai, M., & Abdeljawad, T. (2017). Analysis of the fractional diffusion equations with fractional derivative of non-singular kernel. *Advances in Difference Equations*, *2017*(1), 1-12.

[20] Hasan, S., El-Ajou, A., Hadid, S., Al-Smadi, M., & Momani, S. (2020). Atangana-Baleanu fractional framework of reproducing kernel technique in solving fractional population dynamics system. *Chaos, Solitons & Fractals*, *133*, 109624.

[21] Shah, K., Alqudah, M. A., Jarad, F., & Abdeljawad, T. (2020). Semi-analytical study of Pine Wilt Disease model with convex rate under Caputo–Febrizio fractional order derivative. *Chaos, Solitons & Fractals*, *135*, 109754.

[22] Haq, F., Shah, K., ur Rahman, G., & Shahzad, M. (2018). Numerical solution of fractional order smoking model via laplace Adomian decomposition method. *Alexandria Engineering Journal*, *57*(2), 1061-1069.

[23] Haq, F., Shah, K., Rahman, G. U., & Shahzad, M. (2017). Numerical analysis of fractional order model of HIV-1 infection of CD4+ T-cells. *Computational Methods for Differential Equations*, *5*(1), 1-11.

[24] ur Rahman, M., Arfan, M., Shah, K., & Gómez-Aguilar, J. F. (2020). Investigating a nonlinear dynamical model of COVID-19 disease under fuzzy caputo, random and ABC fractional order derivative. *Chaos, Solitons & Fractals*, *140*, 110232.

[25] Ahmad, S., Ullah, A., Arfan, M., & Shah, K. (2020). On analysis of the fractional mathematical model of rotavirus epidemic with the effects of breastfeeding and vaccination under Atangana-Baleanu (AB) derivative. *Chaos, Solitons & Fractals*, *140*, 110233.

[26] Iserles, A. (1996). *A first course in the numerical analysis of differential equations*. Cambridge University Press.

[27] Butcher J. C. (2003). Numerical Methods for Ordinary Differential Equations, John Wiley, New York.

[28] Baidosov, V. A. (1990). Fuzzy differential inclusions. *Journal of Applied Mathematics and Mechanics*, *54*(1), 8-13.

[29] Chen, M., Fu, Y., Xue, X., & Wu, C. (2008). Two-point boundary value problems of undamped uncertain dynamical systems. *Fuzzy Sets and Systems*, *159*(16), 2077-2089.

[30] Kaleva, O. (1987). Fuzzy differential equations. *Fuzzy sets and systems*, *24*(3), 301-317.

[31] Liu, B. (2008). Fuzzy process, hybrid process and uncertain process. *Journal of Uncertain systems*, *2*(1), 3-16.

[32] Zadeh, L. A. (1996). Fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh* (pp. 394-432).

[33] Chang, S. S., & Zadeh, L. A. (1996). On fuzzy mapping and control. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh* (pp. 180-184).

[34] Dubois, D., & Prade, H. (1982). Towards fuzzy differential calculus part 1: Integration of fuzzy mappings. *Fuzzy sets and Systems*, *8*(1), 1-17.

[35] Goetschel Jr, R., & Voxman, W. (1986). Elementary fuzzy calculus. *Fuzzy sets and systems*, *18*(1), 31-43.

[36] Kaleva, O. (1987). Fuzzy differential equations. *Fuzzy sets and systems*, *24*(3), 301-317.

[37] Babolian, E., Goghary, H. S., & Abbasbandy, S. (2005). Numerical solution of linear Fredholm fuzzy integral equations of the second kind by Adomian method. *Applied Mathematics and Computation*, *161*(3), 733-744.

[38] Khan, Z. H., & Khan, W. A. (2008). N-transform properties and applications. *NUST journal of engineering sciences*, *1*(1), 127-133.

[39] Belgacem, F. B. M., & Silambarasan, R. (2012). Maxwell's equations solutions by means of the natural transform. *International Journal of Mathematics in Engineering, Science and Aerospace*, *3*(3), 313-323.

[40] Asiru, M. A. (2002). Further properties of the Sumudu transform and its applications. *International journal of mathematical education in science and technology*, *33*(3), 441-449.

[41] Rahman, N. A., & Ahmad, M. Z. (2017). Solving fuzzy fractional differential equations using fuzzy Sumudu transform. *Journal of Nonlinear Sciences and Applications*, *10*(1), 2620-2632.

[42] Spiegel, M. R. (1965). *Laplace transforms* (p. 249). New York: McGraw-Hill.

[43] Watugala, G. (1993). Sumudu transform: a new integral transform to solve differential equations and control engineering problems. *Integrated Education*, *24*(1), 35-43.

[44] Belgacem, F. B. M., Karaballi, A. A., & Kalla, S. L. (2003). Analytical investigations of the Sumudu transform and applications to integral production equations. *Mathematical problems in Engineering*, *2003*(3), 103-118.

[45] Belgacem, F. B. M., & Silambarasan, R. (2012). Maxwell's equations solutions by means of the natural transform. *International Journal of Mathematics in Engineering, Science and Aerospace*, *3*(3), 313-323.

[46] Loonker, D., & Banerji, P. K. (2013). Natural transform for distribution and Boehmian spaces. *Mathematics in Engineering, Science & Aerospace (MESA)*, *4*(1).

[47] Zhang, Y. Z., Yang, A. M., & Long, Y. (2014). Initial boundary value problem for fractal heat equation in the semi-infinite region by Yang-Laplace transform. *Thermal Science*, *18*(2), 677-681.

[48] Eltayeb, H., & Kılıçman, A. (2008). A note on solutions of wave, Laplace's and heat equations with convolution terms by using a double Laplace transform. *Applied Mathematics Letters*, *21*(12), 1324-1329.

[49] Povstenko, Y. Z. (2004). Fractional heat conduction equation and associated thermal stress. *Journal of Thermal Stresses*, *28*(1), 83-102.

[50] Antes, H. (1985). A boundary element procedure for transient wave propagations in two-dimensional isotropic elastic media. *Finite Elements in Analysis and Design*, *1*(4), 313-322.

[51] Haberman, R. (1983). *Elementary applied partial differential equations* (Vol. 987). Englewood Cliffs, NJ: Prentice Hall.

[52] Rizzo, F. J., & Shippy, D. J. (1970). A method of solution for certain problems of transient heat conduction. *AIAA Journal*, *8*(11), 2004-2009.

[53] Iqbal, M., Seadawy, A. R., & Lu, D. (2018). Construction of solitary wave solutions to the nonlinear modified Kortewege-de Vries dynamical equation in unmagnetized plasma via mathematical methods. *Modern Physics Letters A*, *33*(32), 1850183.

[54] Ali, A., Seadawy, A. R., & Lu, D. (2018). Computational methods and traveling wave solutions for the fourth-order nonlinear Ablowitz-Kaup-Newell-Segur water wave dynamical equation via two methods and its applications. *Open Physics*, *16*(1), 219-226.

[55] Iqbal, M., Seadawy, A. R., Khalil, O. H., & Lu, D. (2020). Propagation of long internal waves in density stratified ocean for the (2+ 1)-dimensional nonlinear Nizhnik-Novikov-Vesselov dynamical equation. *Results in Physics*, *16*, 102838.

[56] Seadawy, A. R., Iqbal, M., & Lu, D. (2019). Nonlinear wave solutions of the Kudryashov–Sinelshchikov dynamical equation in mixtures liquid-gas bubbles under the consideration of heat transfer and viscosity. *Journal of Taibah University for Science*, *13*(1), 1060-1072.

[57] Özkan, Y. S., Yaşar, E., & Seadawy, A. R. (2020). A third-order nonlinear Schrödinger equation: the exact solutions, group-invariant solutions and conservation laws. *Journal of Taibah University for Science*, *14*(1), 585-597.

[58] Ahmad, H., Seadawy, A. R., Khan, T. A., & Thounthong, P. (2020). Analytic approximate solutions for some nonlinear Parabolic dynamical wave equations. *Journal of Taibah University for Science*, *14*(1), 346-358.

[59] Farah, N., Seadawy, A. R., Ahmad, S., Rizvi, S. T. R., & Younis, M. (2020). Interaction properties of soliton molecules and Painleve analysis for nano bioelectronics transmission model. *Optical and Quantum Electronics*, *52*(7), 1-15.

[60] Arfan, M., Shah, K., Ullah, A., Salahshour, S., Ahmadian, A., & Ferrara, M. (2022). A

novel semi-analytical method for solutions of two dimensional fuzzy fractional wave equation using natural transform. *Discrete & Continuous Dynamical Systems-S*, *15*(2), 315.

[61] Gottwald, S. (1992). Fuzzy set theory and its applications.: Kluwer Academic Publishers, Dordrecht, 1991, viii+ 399 pages, DFL 150, ISBN 0-7923-9075-X.

[62] Zadeh, L. A. (1996). Fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh* (pp. 394-432).

[63] Salahshour, S., Allahviranloo, T., & Abbasbandy, S. (2012). Solving fuzzy fractional differential equations by fuzzy Laplace transforms. *Communications in Nonlinear Science and Numerical Simulation*, *17*(3), 1372-1381.

[64] Miller, K. S., & Ross, B. (1993). *An introduction to the fractional calculus and fractional differential equations*. Wiley.

[65] Perfilieva, I. (2006). Fuzzy transforms: theory and applications. *Fuzzy sets and systems*, *157*(8), 993-1023.

[66] Allahviranloo, T., & Ahmadi, M. B. (2010). Fuzzy laplace transforms. *Soft Computing*, *14*(3), 235-243.

References Chapter 2

[1] Rockafellar, R. T., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, *26*(7), 1443-1471.

[2] Eichhorn, A., & Römisch, W. (2005). Polyhedral risk measures in stochastic programming. *SIAM Journal on Optimization*, *16*(1), 69-95.

[3] Artzner, P., Delbaen, F., Eber, J. M., & Heath, D. (1999). Coherent measures of risk. *Mathematical finance*, *9*(3), 203-228.

[4] Artzner, P., Delbaen, F., Eber, J. M., Heath, D., & Ku, H. (2007). Coherent multiperiod risk adjusted values and Bellman's principle. *Annals of Operations Research*, *152*(1), 5-22.

[5] Eichhorn, A., & Römisch, W. (2005). Polyhedral risk measures in stochastic programming. *SIAM Journal on Optimization*, *16*(1), 69-95.

[6] Ruszczyński, A., & Shapiro, A. (2006). Conditional risk mappings. *Mathematics of operations research*, *31*(3), 544-561.

[7] Rockafellar, R. T. & Wets R., J.-B. (1998). Variational Analysis, Springer, Berlin.

[8] Follmer, H., & Schied, A. (2002). Stochastic Finance: An Introduction in Discrete Time, Vol. 27.

References Chapter 3

[1] Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine Learning.

[2] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[3] Schalkoff, R. J. (1997), "Artificial neural networks" (Vol. 1). New York: McGraw-Hill.

[4] Fauske, K. M. (2006). Example: Neural network. *Accessed April*, *5*, 2018.

[5] Goh, A. T. (1995). Back-propagation neural networks for modeling complex systems. *Artificial intelligence in engineering*, *9*(3), 143-151.

[6] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

[7] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[8] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, *13*(3), 55-75.

[9] Petchot, K., (2020). "fake news" in the time of covid-19. URL:https://bangkok.unesco.org/index.php/content/press-provides-antidote-fake-news-time-covid-19. accessed: 2020-07-10.

[10] Shariq, M., Singh, K., Bajuri, M. Y., Pantelous, A. A., Ahmadian, A., & Salimi, M. (2021). A Secure and reliable RFID authentication protocol using digital schnorr cryptosystem for IoT-enabled healthcare in COVID-19 scenario. *Sustainable Cities and Society*, *75*, 103354.

[11] Bedi, A., Pandey, N., & Khatri, S. K. (2019, October). A framework to identify and secure the issues of fake news and rumours in social networking. In *2019 2Nd International Conference On Power Energy, Environment and Intelligent Control (PEEIC)* (pp. 70-73). IEEE.

[12] Qi, P., Cao, J., Yang, T., Guo, J., & Li, J. (2019, November). Exploiting multi-domain visual information for fake news detection. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 518-527). IEEE.

[13] Bonato, A., Nazareth, A., Facts or fake news: Revealing patterns in the covid-19 tweets of trudeau and trump. URL: https://theconvers ation.com/facts-or-fake-news-revealing-patterns-in-the-covid-1 9-tweets-of-trudeau-and-trump-142139. accessed: 2020-07-10.

[14] Li, G., Zheng, Q., Zhang, L., Guo, S., & Niu, L. (2020, November). Sentiment Infomation based Model for Chinese text Sentiment Analysis. In *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)* (pp. 366-371). IEEE.

[15] Yang, J., & Yang, J. (2020, October). Aspect based sentiment analysis with self-attention and gated convolutional networks. In *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)* (pp. 146-149). IEEE.

[16] Ding, J., Sun, H., Wang, X., & Liu, X. (2018, June). Entity-level sentiment analysis of issue comments. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering* (pp. 7-13).

[17] Snell, N., Fleck, W., Traylor, T., & Straub, J. (2019, December). Manually classified real and fake news articles. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1405-1407). IEEE.

[18] Uppal, A., Sachdeva, V., & Sharma, S. (2020, January). Fake news detection using discourse segment structure analysis. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 751-756). IEEE.

[19] Manzoor, S. I., & Singla, J. (2019, April). Fake news detection using machine learning approaches: A systematic review. In *2019 3rd international conference on trends in electronics and informatics (ICOEI)* (pp. 230-234). IEEE.

[20] Traylor, T., Straub, J., & Snell, N. (2019, January). Classifying fake news articles using natural language processing to identify in-article attribution as a supervised learning estimator. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)* (pp. 445-449). IEEE.

[21] Ramezani, M., Rafiei, M., Omranpour, S., & Rabiee, H. R. (2019, August). News labeling as early as possible: Real or Fake?. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 536-537). IEEE.

[22] Zhang, J., Dong, B., & Philip, S. Y. (2019, December). Deep diffusive neural network based fake news detection from heterogeneous social networks. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 1259-1266). IEEE.

[23] Bhoir, S. V. (2020, January). An efficient fake news detector. In *2020 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-9). IEEE.

[24] Mayakul, T., Srisawat, W., & Kiattisin, S. (2019, December). Fake news and online disinformation. a perspectives of Thai government officials. In *2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)* (pp. 1-4). IEEE.

[25] Barua, R., Maity, R., Minj, D., Barua, T., & Layek, A. K. (2019, July). F-NAD: an application for fake news article detection using machine learning techniques. In *2019 IEEE Bombay Section Signature Conference (IBSSC)* (pp. 1-6). IEEE.

[26] Das, D., & Clark, A. J. (2019, September). Satire vs Fake News: You Can Tell by the Way They Say It. In *2019 First International Conference on Transdisciplinary AI (TransAI)* (pp. 22-26). IEEE.

[27] Xu, K., Wang, F., Wang, H., & Yang, B. (2019). Detecting fake news over online social media via domain reputations and content understanding. *Tsinghua Science and Technology*, *25*(1), 20-27.

[28] Kang, S., Hwang, J., & Yu, H. (2020, January). Multi-modal component embedding for fake news detection. In *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)* (pp. 1-6). IEEE.

[29] Poddar, K., & Umadevi, K. S. (2019, March). Comparison of various machine learning models for accurate detection of fake news. In *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)* (Vol. 1, pp. 1-5). IEEE.

[30] Khan, S. A., Alkawaz, M. H., & Zangana, H. M. (2019, June). The use and abuse of social media for spreading fake news. In *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)* (pp. 145-148). IEEE.

[31] Bhutani, B., Rastogi, N., Sehgal, P., & Purwar, A. (2019, August). Fake news detection using sentiment analysis. In *2019 twelfth international conference on contemporary computing (IC3)* (pp. 1-5). IEEE.

[32] Balestrucci, A., & De Nicola, R. (2020, May). Credulous users and fake news: a real case study on the propagation in twitter. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)* (pp. 1-8). IEEE.

[33] Sear, R. F., Velásquez, N., Leahy, R., Restrepo, N. J., El Oud, S., Gabriel, N., ... & Johnson, N. F. (2020). Quantifying COVID-19 content in the online health opinion war using machine learning. *Ieee Access*, *8*, 91886-91893.

[34] Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., ... & Zheng, C. (2020). A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. *IEEE transactions on medical imaging*, *39*(8), 2615-2625.

[35] Chamola, V., Hassija, V., Gupta, V., & Guizani, M. (2020). A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact. *Ieee access*, *8*, 90225-90265.

[36] Rustam, F., Reshi, A. A., Mehmood, A., Ullah, S., On, B. W., Aslam, W., & Choi, G. S. (2020). COVID-19 future forecasting using supervised machine learning models. *IEEE access*, *8*, 101489-101499.

[37] Liu, H. (2019, December). A location independent machine learning approach for early fake news detection. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 4740-4746). IEEE.

[38] Cui, L., Wang, S., & Lee, D. (2019, August). Same: sentiment-aware multi-modal embedding for detecting fake news. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining* (pp. 41-48).

[39] Shakibaei, S., De Jong, G. C., Alpkökin, P., & Rashidi, T. H. (2021). Impact of the COVID-19 pandemic on travel behavior in Istanbul: A panel data analysis. *Sustainable cities and society*, *65*, 102619.

[40] Najar, F., Zamzami, N., & Bouguila, N. (2019, July). Fake news detection using bayesian inference. In *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)* (pp. 389-394). IEEE.

[41] Basu, B., Murphy, E., Molter, A., Basu, A. S., Sannigrahi, S., Belmonte, M., & Pilla, F. (2021). Investigating changes in noise pollution due to the COVID-19 lockdown: The case of Dublin, Ireland. *Sustainable Cities and Society*, *65*, 102597.

[42] Han, W., & Mehta, V. (2019, November). Fake news detection in social networks using machine learning and deep learning: Performance evaluation. In *2019 IEEE International Conference on Industrial Internet (ICII)* (pp. 375-380). IEEE.

[43] Gupta, V., Jain, N., Katariya, P., Kumar, A., Mohan, S., Ahmadian, A., & Ferrara, M. (2021). An emotion care model using multimodal textual analysis on COVID-19. *Chaos, Solitons & Fractals*, *144*, 110708.

[44] Hirlekar, V. V., & Kumar, A. (2020, June). Natural language processing based online fake news detection challenges–A detailed review. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 748-754). IEEE.

[45] Jain, N., Jhunthra, S., Garg, H., Gupta, V., Mohan, S., Ahmadian, A., ... & Ferrara, M. (2021). Prediction modelling of COVID using machine learning methods from B-cell dataset. *Results in physics*, *21*, 103813.

[46] Iwendi, C., Jalil, Z., Javed, A. R., Reddy, T., Kaluri, R., Srivastava, G., & Jo, O. (2020). Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks. *IEEE Access*, *8*, 72650-72660.

[47] Senthilkumar Mohan, J. A., Abugabah, A., Adimoolam, M., Singh, S. K., kashif Bashir, A., & Sanzogni, L. An approach to forecast impact of Covid-19 using supervised machine learning model. *Software*.

[48] https://www.sciencedaily.com/releases/2020/04/200423130420.htm

[49] Singh, N., Suthar, B., Mehta, A., Nema, N., & Pandey, A. (2020). Corona virus: An immunological perspective review. *Int J Immunol Immunother*, *7*, 050.

[50] Sharma, A., Rani, S., & Gupta, D. (2020). Artificial intelligence-based classification of chest X-ray images into COVID-19 and other infectious diseases. *International journal of biomedical imaging*, *2020*.

[51] Bharati, S., Podder, P., & Mondal, M. R. H. (2020). Hybrid deep learning for detecting lung diseases from X-ray images. *Informatics in Medicine Unlocked*, *20*, 100391.

[52] Mahdy, L. N., Ezzat, K. A., Elmousalami, H. H., Ella, H. A., & Hassanien, A. E. (2020). Automatic x-ray covid-19 lung image classification system based on multi-level thresholding and support vector machine. *MedRxiv*.

[53] Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Soufi, G. J. (2020). Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Medical image analysis*, *65*, 101794.

[54] Jain, R., Gupta, M., Taneja, S., & Hemanth, D. J. (2021). Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Applied Intelligence*, *51*(3), 1690-1700.

[55] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020).

Covid-19 image data collection: Prospective predictions are the future. *arXiv preprint arXiv:2006.11988*.

[56] Shuja, J., Alanazi, E., Alasmary, W., & Alashaikh, A. (2021). COVID-19 open source data sets: a comprehensive survey. *Applied Intelligence*, *51*(3), 1296-1325.

[57] Sharma, A., Rani, S., & Gupta, D. (2020). Artificial intelligence-based classification of chest X-ray images into COVID-19 and other infectious diseases. *International journal of biomedical imaging*, *2020*.

[58] Dubey, R. K. (2020). Deep learning Based hybrid models for prediction of COVID-19 using chest X-ray.

[59] Al-Waisy, A. S., Al-Fahdawi, S., Mohammed, M. A., Abdulkareem, K. H., Mostafa, S. A., Maashi, M. S., ... & Garcia-Zapirain, B. (2020). COVID-CheXNet: hybrid deep learning framework for identifying COVID-19 virus in chest X-rays images. *Soft computing*, 1-16.

[60] Cheng, S. C., Chang, Y. C., Chiang, Y. L. F., Chien, Y. C., Cheng, M., Yang, C. H., ... & Hsu, Y. N. (2020). First case of Coronavirus Disease 2019 (COVID-19) pneumonia in Taiwan. *Journal of the Formosan Medical Association*, *119*(3), 747-751.

[61] Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., & Gloaguen, R. (2020). COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach. *Mathematics*, *8*(6), 890.

[62] Hussain, L., Nguyen, T., Li, H., Abbasi, A. A., Lone, K. J., Zhao, Z., ... & Duong, T. Q. (2020). Machine-learning classification of texture features of portable chest X-ray accurately classifies COVID-19 lung infection. *BioMedical Engineering OnLine*, *19*(1), 1-18.

[63] Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, *10*(1), 1-12.

[64] Sahlol, A. T., Yousri, D., Ewees, A. A., Al-Qaness, M. A., Damasevicius, R., & Abd Elaziz, M. (2020). COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Scientific reports*, *10*(1), 1-15.

References Chapter 4

[1] Chan, J. F. W., Kok, K. H., Zhu, Z., Chu, H., To, K. K. W., Yuan, S., & Yuen, K. Y. (2020). Genomic characterization of the 2019 novel human-pathogenic coronavirus isolated from a patient with atypical pneumonia after visiting Wuhan. *Emerging microbes & infections*, *9*(1), 221-236.

[2] Lu, H., Stratton, C. W., & Tang, Y. W. (2020). Outbreak of pneumonia of unknown etiology in Wuhan, China: The mystery and the miracle. *Journal of medical virology*, *92*(4), 401.

[3] Ji, W., Wang, W., Zhao, X., Zai, J., Li, X. (2020). Homologous recombination within the spike glycoprotein of the newly identified coronavirus may boost cross-species

transmission from snake to human. J. Med Virol. 268.

[4] Fahmi, I. (2019). World Health Organization coronavirus disease 2019 (Covid-19) situation report. *DroneEmprit*.

[5] Chen, Y., & Guo, D. (2016). Molecular mechanisms of coronavirus RNA capping and methylation. *Virologica Sinica*, *31*(1), 3-11.

[6] Wang, L. F., Shi, Z., Zhang, S., Field, H., Daszak, P., & Eaton, B. T. (2006). Review of bats and SARS. *Emerging infectious diseases*, *12*(12), 1834.

[7] Ge, X. Y., Li, J. L., Yang, X. L., Chmura, A. A., Zhu, G., Epstein, J. H., ... & Shi, Z. L. (2013). Isolation and characterization of a bat SARS-like coronavirus that uses the ACE2 receptor. *Nature*, *503*(7477), 535-538.

[8] Lu, R., Zhao, X., Li, J., Niu, P., Yang, B., Wu, H., ... & Tan, W. (2020). Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. *The lancet*, *395*(10224), 565-574.

[9] Zhou, P., Yang, X. L., Wang, X. G., Hu, B., Zhang, L., Zhang, W., ... & Shi, Z. L. (2020). A pneumonia outbreak associated with a new coronavirus of probable bat origin. *nature*, *579*(7798), 270-273.

[10] Tian, X., Li, C., Huang, A., Xia, S., Lu, S., Shi, Z., ... & Ying, T. (2020). Potent binding of 2019 novel coronavirus spike protein by a SARS coronavirus-specific human monoclonal antibody. *Emerging microbes & infections*, *9*(1), 382-385.

[11] Ahmed, S. F., Quadeer, A. A., & McKay, M. R. (2020). Preliminary identification of potential vaccine targets for the COVID-19 coronavirus (SARS-CoV-2) based on SARS-CoV immunological studies. *Viruses*, *12*(3), 254.

[12] Chaudhury, S., Berrondo, M., Weitzner, B. D., Muthu, P., Bergman, H., & Gray, J. J. (2011). Benchmarking and analysis of protein docking performance in Rosetta v3. 2. *PloS one*, *6*(8), e22477.

[13] Khan, M. A., & Atangana, A. (2020). Modeling the dynamics of novel coronavirus (2019-nCov) with fractional derivative. *Alexandria Engineering Journal*, *59*(4), 2379-2389.

[14] Agarwal, R. P., Lakshmikantham, V., & Nieto, J. J. (2010). On the concept of solution for fractional differential equations with uncertainty. *Nonlinear Analysis: Theory, Methods & Applications*, *72*(6), 2859-2862.

[15] Asjad, M. I., Aleem, M., Ahmadian, A., Salahshour, S., & Ferrara, M. (2020). New trends of fractional modeling and heat and mass transfer investigation of (SWCNTs and MWCNTs)-CMC based nanofluids flow over inclined plate with generalized boundary conditions. *Chinese Journal of Physics*, *66*, 497-516.

[16] Aleem, M., Asjad, M. I., Chowdhury, M. S., & Hussanan, A. (2019). Analysis of mathematical model of fractional viscous fluid through a vertical rectangular channel. *Chinese Journal of Physics*, *61*, 336-350.

[17] Aleem, M., Asjad, M. I., Ahmadian, A., Salimi, M., & Ferrara, M. (2020). Heat transfer analysis of channel flow of MHD Jeffrey fluid subject to generalized

boundary conditions. *The European physical journal plus*, *135*(1), 1-15.

[18] Imran, M. A., Aleem, M., Riaz, M. B., Ali, R., & Khan, I. (2019). A comprehensive report on convective flow of fractional (ABC) and (CF) MHD viscous fluid subject to generalized boundary conditions. *Chaos, Solitons & Fractals*, *118*, 274-289.

[19] Kaleva, O. (1987). Fuzzy differential equations. *Fuzzy sets and systems*, *24*(3), 301-317.

[20] Lupulescu, V. (2015). Fractional calculus for interval-valued functions. *Fuzzy Sets and Systems*, *265*, 63-85.

[21] Arshad, S., & Lupulescu, V. (2011). Fractional differential equation with the fuzzy initial condition. *Electronic Journal of Differential Equations (EJDE)[electronic only]*, *2011*, Paper-No.

[22] Benchohra, M., Cabada, A., & Seba, D. (2009). An existence result for nonlinear fractional differential equations on Banach spaces. *Boundary Value Problems*, *2009*, 1-11.

[23] Belmekki, M., Nieto, J., & Rodriguez-Lopez, R. (2009). Existence of periodic solution for a nonlinear fractional differential equation. *Boundary Value Problems*, *2009*, 1-18.

[24] Park, J. Y., & Han, H. K. (1999). Existence and uniqueness theorem for a solution of fuzzy Volterra integral equations. *Fuzzy Sets and Systems*, *105*(3), 481-488.

[25] Ali, N., & Khan, R. A. (2016). Existence of positive solution to a class of fractional differential equations with three point boundary conditions. *Mathematical Sciences Letters*, *5*(3), 291-296.

[26] Khan, R. A., & Shah, K. (2015). Existence and uniqueness of solutions to fractional order multi-point boundary value problems. *Commun. Appl. Anal*, *19*(3-4), 515-525.

[27] Lakshmikantham, V., & Leela, S. (2009). Nagumo-type uniqueness result for fractional differential equations. *Nonlinear Analysis*, *7*(71), 2886-2889.

[28] Lakshmikantham, V., & Vatsala, A. S. (2008). Basic theory of fractional differential equations. *Nonlinear Analysis: Theory, Methods & Applications*, *69*(8), 2677-2682.

[29] Allahviranloo, T., Salahshour, S., & Abbasbandy, S. (2012). Explicit solutions of fractional differential equations with uncertainty. *Soft Computing*, *16*(2), 297-302.

[30] Zhu, Y. (2010). Stability analysis of fuzzy linear differential equations. *Fuzzy Optimization and Decision Making*, *9*(2), 169-186.

References Chapter 5

[1] Werner, T. G., & Remberg, R. (2008, April). Technical, economical and regulatory aspects of Virtual Power Plants. In *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies* (pp. 2427-2433). IEEE.

[2] Martin-Martínez, F., Sánchez-Miralles, A., & Rivier, M. (2016). A literature review of

Microgrids: A functional layer based classification. *Renewable and sustainable energy reviews*, *62*, 1133-1153.

[3] Beraldi, P., Violi, A., Carrozzino, G., & Bruni, M. E. (2018). A stochastic programming approach for the optimal management of aggregated distributed energy resources. *Computers & Operations Research*, *96*, 200-212.

[4] Beraldi, P., Conforti, D., Triki, C., & Violi, A. (2004). Constrained auction clearing in the Italian electricity market. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *2*(1), 35-51.

[5] Conejo, A. J., Garcia-Bertrand, R., Carrion, M., Caballero, Á., & de Andres, A. (2008). Optimal involvement in futures markets of a power producer. *IEEE Transactions on Power Systems*, *23*(2), 703-711.

[6] Hatami, A. R., Seifi, H., & Sheikh-El-Eslami, M. K. (2009). Optimal selling price and energy procurement strategies for a retailer in an electricity market. *Electric Power Systems Research*, *79*(1), 246-254.

[7] Conejo, A. J., Fernandez-Gonzalez, J. J., & Alguacil, N. (2005). Energy procurement for large consumers in electricity markets. *IEE Proceedings-Generation, Transmission and Distribution*, *152*(3), 357-364.

[8] Conejo, A. J., & Carrion, M. (2006). Risk-constrained electricity procurement for a large consumer. *IEE Proceedings-Generation, Transmission and Distribution*, *153*(4), 407-413.

[9] Carrion, M., Conejo, A. J., & Arroyo, J. M. (2007). Forward contracting and selling price determination for a retailer. *IEEE Transactions on Power Systems*, *22*(4), 2105-2114.

[10] Zare, K., Moghaddam, M. P., & El Eslami, M. K. S. (2010). Electricity procurement for large consumers based on information gap decision theory. *Energy Policy*, *38*(1), 234-242.

[11] Beraldi, P., Violi, A., Scordino, N., & Sorrentino, N. (2011). Short-term electricity procurement: A rolling horizon stochastic programming approach. *Applied Mathematical Modelling*, *35*(8), 3980-3990.

[12] Beraldi, P., Violi, A., Carrozzino, G., & Bruni, M. E. (2017, September). The optimal energy procurement problem: a stochastic programming approach. In *International Conference on Optimization and Decision Science* (pp. 357-365). Springer, Cham.

[13] Beraldi, P., Violi, A., Carrozzino, G., & Bruni, M. E. (2017). The optimal electric energy procurement problem under reliability constraints. *Energy Procedia*, *136*, 283-289.

[14] Beraldi, P., Violi, A., Bruni, M. E., & Carrozzino, G. (2017). A probabilistically constrained approach for the energy procurement problem. *Energies*, *10*(12), 2179.

[15] Leo, E., & Engell, S. (2018). Multi-stage integrated electricity procurement and production scheduling. In *Computer Aided Chemical Engineering* (Vol. 44, pp. 1291-1296). Elsevier.

[16] Carrión, M., Philpott, A. B., Conejo, A. J., & Arroyo, J. M. (2007). A stochastic programming approach to electric energy procurement for large consumers. *IEEE*

Transactions on Power Systems, *22*(2), 744-754.

[17] Triki, C., & Violi, A. (2009). Dynamic pricing of electricity in retail markets. *4or*, *7*(1), 21-36.

[18] Fotouhi Ghazvini, M. A., Soares, J., Morais, H., Castro, R., & Vale, Z. (2017). Dynamic pricing for demand response considering market price uncertainty. *Energies*, *10*(9), 1245.

[19] Nojavan, S., Qesmati, H., Zare, K., & Seyyedi, H. (2014). Large consumer electricity acquisition considering time-of-use rates demand response programs. *Arabian Journal for Science and Engineering*, *39*(12), 8913-8923.

[20] Fridgen, G., Kahlen, M., Ketter, W., Rieger, A., & Thimmel, M. (2018). One rate does not fit all: An empirical analysis of electricity tariffs for residential microgrids. *Applied energy*, *210*, 800-814.

[21] Kovács, A. (2019). Bilevel programming approach to demand response management with day-ahead tariff. *Journal of Modern Power Systems and Clean Energy*, *7*(6), 1632-1643.

[22] Violi, A., Beraldi, P., Ferrara, M., Carrozzino, G., & Bruni, M. E. (2018). The optimal tariff definition problem for a prosumers' aggregation. In *New Trends in Emerging Complex Real Life Problems* (pp. 483-492). Springer, Cham.

[23] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.

[24] Beraldi, P., De Simone, F., & Violi, A. (2010). Generating scenario trees: A parallel integrated simulation–optimization approach. *Journal of Computational and Applied Mathematics*, *233*(9), 2322-2331.

[25] Pichler, A. (2013). Evaluations of risk measures for different probability measures. *SIAM Journal on Optimization*, *23*(1), 530-551.

[26] Menniti, D., Scordino, N., Sorrentino, N., & Violi, A. (2010, June). Short-term forecasting of day-ahead electricity market price. In *2010 7th International conference on the European Energy Market* (pp. 1-5). IEEE.

[27] Ruszczyński, A., & Shapiro, A. (2003). Stochastic programming models. *Handbooks in operations research and management science*, *10*, 1-64).

References Chapter 6

[1] Markowitz, H. (1952). Portfolio selection. The journal of finance, 7(1), 77-91.

[2] Lejeune, M. A., & Shen, S. (2016). Multi-objective probabilistically constrained programs with variable risk: Models for multi-portfolio financial optimization. *European Journal of Operational Research*, *252*(2), 522-539.

[3] Nguyen, T. T., Gordon-Brown, L., Khosravi, A., Creighton, D., & Nahavandi, S. (2014). Fuzzy portfolio allocation models through a new risk measure and fuzzy sharpe ratio. *IEEE Transactions on Fuzzy Systems*, *23*(3), 656-676.

[4] Bonami, P., & Lejeune, M. A. (2009). An exact solution approach for portfolio

optimization problems under stochastic and integer constraints. *Operations research*, *57*(3), 650-670.

[5] Freeman, P., & Gear, A. E. (1971). A probabilistic objective function for R & D portfolio selection. *Journal of the Operational Research Society*, *22*(3), 253-265.

[6] Roy, A. D. (1952). Safety first and the holding of assets. *Econometrica: Journal of the econometric society*, 431-449.

[7] Sun, Y., Aw, G., Teo, K. L., & Zhou, G. (2015). Portfolio optimization using a new probabilistic risk measure. *Journal of Industrial & Management Optimization*, *11*(4), 1275.

[8] Uryasev, S., & Pardalos, P. M. (Eds.). (2013). *Stochastic optimization: algorithms and applications* (Vol. 54). Springer Science & Business Media.

[9] Ehrgott, M., & Gandibleux, X. (2002). Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, chap. Multiobjective Combinatorial Optimization. *International Series in Operations Research & Management Science. Kluwer Academic Publishers, Dordrecht*, 369-407.

[10] Miettinen, K. (2012). *Nonlinear multiobjective optimization* (Vol. 12). Springer Science & Business Media.

[11] Steuer, R. E. (1986). Multiple criteria optimization. *Theory, computation and applications*.

[12] Chow, G. (1995). Portfolio selection based on return, risk, and relative performance. *Financial Analysts Journal*, *51*(2), 54-60.

[13] Anagnostopoulos, K. P., & Mamanis, G. (2010). A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, *37*(7), 1285-1297.

[14] Meghwani, S. S., & Thakur, M. (2018). Multi-objective heuristic algorithms for practical portfolio optimization and rebalancing with transaction cost. *Applied Soft Computing*, *67*, 865-894.

[15] Qi, Y., Steuer, R. E., & Wimmer, M. (2017). An analytical derivation of the efficient surface in portfolio selection with three criteria. *Annals of Operations Research*, *251*(1), 161-177.

[16] Utz, S., Wimmer, M., Hirschberger, M., & Steuer, R. E. (2014). Tri-criterion inverse portfolio optimization with application to socially responsible mutual funds. *European Journal of Operational Research*, *234*(2), 491-498.

[17] Jin, X., Chen, N., & Yuan, Y. (2019). Multi-period and tri-objective uncertain portfolio selection model: A behavioral approach. *The North American journal of economics and finance*, *47*, 492-504.

[18] Parra, M. A., Terol, A. B., & Urıa, M. R. (2001). A fuzzy goal programming approach to portfolio selection. *European Journal of Operational Research*, 133(2), 287-297.

[19] Saborido, R., Ruiz, A. B., Bermudez, J. D., Vercher, E., & Luque, M. (2016).

Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection. *Applied soft computing*, *39*, 48-63.

[20] Meghwani, S. S., & Thakur, M. (2017). Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and evolutionary computation*, *37*, 104-125.

[21] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.

[22] Goldfarb, D., & Iyengar, G. (2003). Robust portfolio selection problems. *Mathematics of operations research*, *28(1)*, 1-38.

[23] Tütüncü, R. H., & Koenig, M. (2004). Robust asset allocation. *Annals of Operations Research*, *132(1- 4)*, 157-187.

[24] Fabozzi, F. J., Focardi, S. M., Kolm, P. N., & Pachamanova, D. A. (2007). *Robust portfolio optimization and management*. John Wiley & Sons.

[25] Fabozzi, F. J., Huang, D., & Zhou, G. (2010). Robust portfolios: contributions from operations research and finance. *Annals of operations research*, *176*(1), 191-220.

[26] Gülpınar, N., & Çanakoğlu, E. (2017). Robust portfolio selection problem under temperature uncertainty. *European Journal of Operational Research*, *256*(2), 500-523.

[27] Fernandes, B., Street, A., Valladão, D., & Fernandes, C. (2016). An adaptive robust portfolio optimization model with loss constraints based on data-driven polyhedral uncertainty sets. *European Journal of Operational Research*, *255*(3), 961-970.

[28] Li, J. Y., & Kwon, R. H. (2013). Portfolio selection under model uncertainty: a penalized moment-based optimization approach. *Journal of Global Optimization*, *56*(1), 131-164.

[29] Cont, R. (2006). Model uncertainty and its impact on the pricing of derivative instruments. *Mathematical finance*, *16*(3), 519-547.

[30] Consigli, G., Kuhn, D., & Brandimarte, P. (2017). Optimal financial decision making under uncertainty. In *Optimal Financial Decision Making under Uncertainty* (pp. 255-290). Springer, Cham.

[31] Qin, Z., Kar, S., & Zheng, H. (2016). Uncertain portfolio adjusting model using semiabsolute deviation. *Soft Computing*, *20*(2), 717-725.

[32] Ning, Y., Yan, L., & Xie, Y. (2012). Mean-TVaR model for portfolio selection with uncertain returns. *Information*, *15*, 129-137.

[33] Kaucic, M. (2019). Equity portfolio management with cardinality constraints and risk parity control using multi-objective particle swarm optimization. *Computers & Operations Research*, *109*, 300-316.

[34] Liu, Y. J., & Zhang, W. G. (2013). Fuzzy portfolio optimization model under real constraints. Insurance: *Mathematics and Economics*, *53*(3), 704-711.

[35] Zhang, P., & Zhang, W. G. (2014). Multiperiod mean absolute deviation fuzzy portfolio selection model with risk control and cardinality constraints. *Fuzzy Sets and Systems*, *255*,

74-91.

[36] Ferreira, L., Borenstein, D., Righi, M. B., & de Almeida Filho, A. T. (2018). A fuzzy hybrid integrated framework for portfolio optimization in private banking. *Expert Systems with Applications*, *92*, 350- 362.

[37] Gupta, P., Mehlawat, M. K., Yadav, S., & Kumar, A. (2019). A polynomial goal programming approach for intuitionistic fuzzy portfolio optimization using entropy and higher moments. *Applied Soft Computing*, *85*, 105781.

[38] Moral-Escudero, R., Ruiz-Torrubiano, R., & Suárez, A. (2006, July). Selection of optimal investment portfolios with cardinality constraints. In *2006 IEEE International Conference on Evolutionary Computation* (pp. 2382-2388). IEEE.

[39] Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, *27*(13), 1271-1302.

[40] Woodside-Oriakhi, M., Lucas, C., & Beasley, J. E. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*, *213*(3), 538-550.

[41] Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. Nonlinear analysis: *Real world applications*, *10*(4), 2396-2406.

[42] Ni, Q., Yin, X., Tian, K., & Zhai, Y. (2017). Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Natural Computing*, *16*(1), 31-44.

[43] Anagnostopoulos, K. P., & Mamanis, G. (2011). The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, *38*(11), 14208-14217.

[44] Krink, T., & Paterlini, S. (2011). Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science*, *8*(1-2), 157-179.

[45] Macedo, L. L., Godinho, P., & Alves, M. J. (2017). Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Systems with Applications*, *79*, 33-43.

[46] Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, *24*, 757-772.

[47] Solares, E., Coello, C. A. C., Fernandez, E., & Navarro, J. (2019). Handling uncertainty through confidence intervals in portfolio optimization. *Swarm and evolutionary computation*, *44*, 774-787.

[48] Liagkouras, K., & Metaxiotis, K. (2018). A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem. *Annals of Operations Research*, *267*(1-2), 281-319.

[49] Lwin, K. T., Qu, R., & MacCarthy, B. L. (2017). Mean-VaR portfolio optimization: A

nonparametric approach. *European Journal of Operational Research, 260*(2), 751-766.

[50] Chen, B., Lin, Y., Zeng, W., Xu, H., & Zhang, D. (2017). The mean-variance cardinality constrained portfolio optimization problem using a local search-based multi-objective evolutionary algorithm. *Applied Intelligence, 47*(2), 505-525.

[51] Silva, Y. L. T., Herthel, A. B., & Subramanian, A. (2019). A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems. *Expert Systems with Applications, 133*, 225-241.

[52] Babazadeh, H., & Esfahanipour, A. (2019). A novel multi period mean-VaR portfolio optimization model considering practical constraints and transaction cost. *Journal of Computational and Applied Mathematics, 361*, 313-342.

[53] Fernandez, E., Navarro, J., Solares, E., & Coello, C. C. (2019). A novel approach to select the best portfolio considering the preferences of the decision maker. *Swarm and Evolutionary Computation, 46*, 140-153.

[54] Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation, 1*(1), 32-49.

[55] Doering, J., Kizys, R., Juan, A. A., Fitó, À., & Polat, O. (2019). Metaheuristics for rich portfolio optimisation and risk management: current state and future trends. *Operations Research Perspectives*, 100121.

[56] Kalayci, C. B., Ertenlice, O., & Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications.*

[57] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation, 6*(2), 182-197.

[58] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report, 103*.

[59] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation, 11*(6), 712-731.

[60] Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001, July). PESA-II: Region-based selection in evolutionary multiobjective optimization. *In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (pp. 283-290).

[61] Baliarsingh, S. K., Vipsita, S., Muhammad, K., & Bakshi, S. (2019). Analysis of high-dimensional biomedical data using an evolutionary multi-objective emperor penguin optimizer. *Swarm and Evolutionary Computation, 48*, 262-273.

[62] Konstantinidis, A., & Yang, K. (2011). Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing, 11*(6), 4117- 4134.

[63] Guzek, M., Pecero, J. E., Dorronsoro, B., & Bouvry, P. (2014). Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems.

Applied Soft Computing, 24, 432-446.

[64] Deep, K., & Thakur, M. (2007). A new crossover operator for real coded genetic algorithms. *Applied mathematics and computation, 188*(1), 895-911.

[65] Deep, K., & Thakur, M. (2007). A new mutation operator for real coded genetic algorithms. *Applied mathematics and Computation, 193*(1), 211-230.

[66] Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering, 191*(11-12), 1245-1287.

[67] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In MHS'95. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39-43). *Ieee.*

[68] Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation, 8*(3), 256-279.

[69] Wu, D., Ogawa, M., Suzuki, Y., Ogai, H., & Kusaka, J. (2010). Modified multi-objective particle swarm optimization: Application to Optimization of Diesel Engine Control Parameter. *SICE Journal of Control, Measurement, and System Integration, 3*(5), 315-323.

[70] Wang, X., & Tang, L. (2011, June). A PSO-based hybrid multi-objective algorithm for multi-objective optimization problems. *In International Conference in Swarm Intelligence* (pp. 26-33). Springer, Berlin, Heidelberg.

[71] Gao, Y. L., & Qu, M. (2012, July). Constrained multi-objective particle swarm optimization algorithm. *In International Conference on Intelligent Computing* (pp. 47-55). Springer, Berlin, Heidelberg.

[72] Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of the operational research society, 41*(11), 1069-1072.

[73] Becker, W., Saisana, M., Paruolo, P., & Vandecasteele, I. (2017). Weights and importance in composite indicators: Closing the gap. *Ecological indicators, 80*, 12-22.

[74] Seth, S., & McGillivray, M. (2018). Composite indices, alternative weights, and comparison robustness. *Social Choice and Welfare, 51*(4), 657-679.

[75] dos Santos, S. F., & Brandi, H. S. (2017). Selecting portfolios for composite indexes: application of Modern Portfolio Theory to competitiveness. *Clean Technologies and Environmental Policy, 19*(10), 2443-2453.

[76] Joint Research Centre-European Commission. (2008). Handbook on constructing composite indicators: methodology and user guide. *OECD publishing*.

[77] CHAKRABARTTY, S. N. (2017). Composite index: methods and properties. *Journal of Applied Quantitative Methods,* 25-33.

[78] Fernandez, E., Navarro, J., Solares, E., & Coello, C. C. (2020). Using evolutionary computation to infer the decision maker's preference model in presence of imperfect knowledge: A case study in portfolio optimization. *Swarm and Evolutionary*

Computation, 54, 100648.

[79] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation, 3*(4), 257-271.

[80] Van Veldhuizen, D. A., & Lamont, G. B. (1998, July). Evolutionary computation and convergence to a pareto front. *In Late breaking papers at the genetic programming 1998 conference* (pp. 221-228).

[81] Sierra, M. R., & Coello, C. A. C. (2005, March). Improving PSO-based multi-objective optimization using crowding, mutation and∈-dominance. *In International conference on evolutionary multi-criterion optimization* (pp. 505-519). Springer, Berlin, Heidelberg.

[82] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation, 7*(2), 117-132.

[83] Tu, B., & He, C. (2020). Investor's portfolio decision: perspective of parameter uncertainty. *Systems Science & Control Engineering, 8*(1), 48-56.

[84] Garlappi, L., Uppal, R., & Wang, T. (2007). Portfolio selection with parameter and model uncertainty: A multi-prior approach. *The Review of Financial Studies, 20*(1), 41-81.

[85] Asadujjaman, M., & Zaman, K. (2019). Robustness-based portfolio optimization under epistemic uncertainty. *Journal of Industrial Engineering International, 15*(2), 207-219.

[86] Konno, H. (1990). Piecewise linear risk function and portfolio optimization. *Journal of the Operations Research Society of Japan, 33*(2), 139-156.

[87] Konno, H., & Suzuki, K. I. (1995). A mean-variance-skewness portfolio optimization model. *Journal of the Operations Research Society of Japan, 38*(2), 173-187.

[88] Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management science, 37*(5), 519-531.

[89] Sharpe, A., & Andrews, B. (2012). An assessment of weighting methodologies for composite indicators: The case of the index of economic well-being. *Centre for the Study of Living Standards (CLS) research report,* (2012-10).

[90] Hagerty, M. R., & Land, K. C. (2007). Constructing summary indices of quality of life: A model for the effect of heterogeneous importance weights. *Sociological Methods & Research, 35*(4), 455-496.

References Chapter 7

[1] Li, S. (2020). Explore covid-19 infodemic. URL: https://towardsdatascience.com/explore-covid-19-infodemic-2d1ceaae2306. accessed: 2020-07-10.

References Chapter 8

[1] Aboutalebi, H., Pavlova, M., Shafiee, M. J., Sabri, A., Alaref, A., & Wong, A. (2021). Covid-net cxr-s: Deep convolutional neural network for severity assessment of covid-19 cases from chest x-ray images. *Diagnostics*, *12*(1), 25.

[2] https://github.com/GeneralBlockchain/covid-19-chest-xray-lung-bounding-boxes-dataset.

[3] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020). Covid-19 image data collection: Prospective predictions are the future. *arXiv preprint arXiv:2006.11988*.

[4] COVID, C. A. A. (19). chest x-ray data initiative; 2020. *URL https://github. com/agchung/Actualmed-COVID-chestxray-dataset*.

[5] https://github.com/agchung/Actualmed-COVID-chestxray-dataset (2020).

[6] Aboutalebi, H., Pavlova, M., Shafiee, M. J., Sabri, A., Alaref, A., & Wong, A. (2021). Covid-net cxr-s: Deep convolutional neural network for severity assessment of covid-19 cases from chest x-ray images. *Diagnostics*, *12*(1), 25.

[7] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[8] https://www.kaggle.com/c/dlai3/data

[9] https://github.com/ieee8023/covid-chestxray-dataset

[10] https://github.com/agchung/Actualmed-COVID-chestxray-dataset (2020).

[11] Shuai, J., Shen, C., & Zhu, Z. (2017). Adaptive morphological feature extraction and support vector regressive classification for bearing fault diagnosis. *International Journal of Rotating Machinery*, *2017*.

[12] Khalvati, F., Wong, A., & Haider, M. A. (2015). Automated prostate cancer detection via comprehensive multi-parametric magnetic resonance imaging texture feature models. *BMC medical imaging*, *15*(1), 1-14.

[13] Haider, M. A., Vosough, A., Khalvati, F., Kiss, A., Ganeshan, B., & Bjarnason, G. A. (2017). CT texture analysis: a potential tool for prediction of survival in patients with metastatic clear cell carcinoma treated with sunitinib. *Cancer Imaging*, *17*(1), 1-9.

[14] Guru, D. S., Sharath, Y. H., & Manjunath, S. (2010). Texture features and KNN in classification of flower images. *IJCA, Special Issue on RTIPPR (1)*, 21-29.

[15] Yu, H., Scalera, J., Khalid, M., Touret, A. S., Bloch, N., Li, B., ... & Anderson, S. W. (2017). Texture analysis as a radiomic marker for differentiating renal tumors. *Abdominal Radiology*, *42*(10), 2470-2478.

[16] Castellano, G., Bonilha, L., Li, L. M., & Cendes, F. (2004). Texture analysis of medical images. *Clinical radiology*, *59*(12), 1061-1069.

[17] Khuzi, A. M., Besar, R., & Zaki, W. M. D. (2008). Texture features selection for masses

detection in digital mammogram. In *4th Kuala Lumpur international conference on biomedical engineering 2008* (pp. 629-632). Springer, Berlin, Heidelberg.

[18] Esgiar, A. N., Naguib, R. N., Sharif, B. S., Bennett, M. K., & Murray, A. (2002). Fractal analysis in the detection of colonic cancer images. *IEEE transactions on information technology in biomedicine*, *6*(1), 54-58.

[19] Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems*, *91*(2), 179-189.

[20] Kundu, R., Singh, P. K., Ferrara, M., Ahmadian, A., & Sarkar, R. (2021). ET-NET: an ensemble of transfer learning models for prediction of COVID-19 infection through chest CT-scan images. *Multimedia Tools and Applications*, 1-20.

[21] Fouladi, S., Ebadi, M. J., Safaei, A. A., Bajuri, M. Y., & Ahmadian, A. (2021). Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio. *Computer Communications*.

[22] Mohan, S., Abugabah, A., Kumar Singh, S., kashif Bashir, A., & Sanzogni, L. (2021). An approach to forecast impact of Covid-19 using supervised machine learning model. *Software: Practice and Experience.*

[23] Gupta, V., Jain, N., Katariya, P., Kumar, A., Mohan, S., Ahmadian, A., & Ferrara, M. (2021). An emotion care model using multimodal textual analysis on COVID-19. *Chaos, Solitons & Fractals*, 144, 110708.

[24] Luz, E., Silva, P., Silva, R., Silva, L., Guimarães, J., Miozzo, G., ... & Menotti, D. (2021). Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. *Research on Biomedical Engineering*, 1-14.

[25] Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., & Singh, V. (2020). Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet. *Chaos, Solitons & Fractals*, *138*, 109944.

[26] Sitaula, C., & Hossain, M. B. (2021). Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Applied Intelligence*, 51(5), 2850-2863.

[27] Ahsan, M., Based, M., Haider, J., & Kowalski, M. (2021). COVID-19 detection from chest X-ray images using feature fusion and deep learning. *Sensors*, *21*(4), 1480.

[28] Hirra, I., Ahmad, M., Hussain, A., Ashraf, M. U., Saeed, I. A., Qadri, S. F., ... & Alfakeeh, A. S. (2021). Breast Cancer Classification from Histopathological Images Using Patch-Based Deep Learning Modeling. *IEEE Access*, *9*, 24273-24287.

[29] Mengistu, A. D. (2018). The effects of segmentation techniques in digital image based identification of Ethiopian coffee variety. *Telkomnika*, *16*(2), 713-717.

[30] Lyu, Y., Chen, J., & Song, Z. (2018). Image-based process monitoring using deep belief networks. *IFAC-PapersOnLine*, *51*(18), 115-120.

References Chapter 9

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

[2] Giudici, G., Milne, A., & Vinogradov, D. (2020). Cryptocurrencies: market analysis and perspectives. *Journal of Industrial and Business Economics*, *47*(1), 1-18.

[3] Arias-Oliva, M., Pelegrín-Borondo, J., & Matías-Clavero, G. (2019). Variables influencing cryptocurrency use: a technology acceptance model in Spain. *Frontiers in Psychology*, *10*, 475.

[4] Jaquart, P., Dann, D., & Weinhardt, C. (2021a). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science, 7,* 45–66.

[5] Golbabai, A., & Ezazipour, S. (2020). A projection-based recurrent neural network and its application in solving convex quadratic bilevel optimization problems. *Neural Computing and Applications*, *32*(8), 3887-3900.

[6] Jaquart, P., Dann, D., & Weinhardt, C. (2021b). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science*.

[7] Ezazipour, S., & Golbabai, A. (2020). A globally convergent neurodynamics optimization model for mathematical programming with equilibrium constraints. *Kybernetika*, *56*(3), 383-409.

[8] Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, *7*(1), 1-30.

[9] Hitam, N. A., & Ismail, A. R. (2018). Comparative performance of machine learning algorithms for cryptocurrency forecasting. *Ind. J. Electr. Eng. Comput. Sci*, *11*(3), 1121-1128.

[10] Fauzi, M. A., Paiman, N., & Othman, Z. (2020). Bitcoin and cryptocurrency: Challenges, opportunities and future works. *The Journal of Asian Finance, Economics, and Business*, *7*(8), 695-704.

[11] Jamali, N., Sadegheih, A., Lotfi, M. M., Wood, L. C., & Ebadi, M. J. (2021). Estimatin the depth of anesthesia during the induction by a novel adaptive neuro-fuzzy inference system: a case study. *Neural Processing Letters*, *53*(1), 131-175.

[12] Ebadi, M. J., Hosseini, A., & Hosseini, M. M. (2017). A projection type steepest descent neural network for solving a class of nonsmooth optimization problems. *Neurocomputing*, *235*, 164-181.

[13] Tirandazi, P., Rahiminasab, A., & Ebadi, M. J. (2022). An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 1-23.

[14] Farhath, Z. A., Arputhamary, B., & Arockiam, L. (2016). A survey on ARIMA forecasting using time series model. *Int. J. Comput. Sci. Mobile Comput*, *5*, 104-109.

[15] Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. Future Computing and Informatics Journal, 3(2), 334–340.

[16] Wang, L., Zou, H., Su, J., Li, L., & Chaudhry, S. (2013). An ARIMA-ANN hybrid model for time series forecasting. *Systems Research and Behavioral Science, 30*(3), 244-259.

[17] Khandelwal, I., Adhikari, R., & Verma, G. (2019). Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Procedia Computer Science, 48,* 173-179.

[18] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50,* 159-175.

[19] Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*, 112395.

Ringraziamenti

Giunta alla fine della stesura di questa tesi dottorale e guardando indietro con occhi diversi - più maturi e più ricchi sia in termini di conoscenza che umani - non potevo non ringraziare le persone che hanno contribuito alla realizzazione di questo importantissimo traguardo.

Un doveroso ma soprattutto sentito ringraziamento va al mio Maestro Prof. Massimiliano Ferrara che prima ancora di avermi fornito un'impeccabile guida scientifica è stato per me un esempio umano e morale che non dimenticherò, un mentore, una guida pronta a fornirmi strumenti, migliorie, affinché il presente lavoro di ricerca potesse avere il giusto riconoscimento scientifico.

Un pensiero di grande affetto alla Prof.ssa Maria Barbati per i preziosi consigli dati durante il mio periodo di soggiorno estero presso l'Università di Portsmouth.

Un ringraziamento speciale va alla mia famiglia per la vicinanza e il supporto fornitomi durante questi anni di formazione e crescita professionale.

Infine, ma non meno importante, un ringraziamento particolare va ai miei colleghi e amici Valentina, Olga e Pasquale veri e propri sostenitori durante questo percorso. La vostra vicinanza nei momenti di gioia, ma anche di sconforto è stata fondamentale per ottenere questo risultato.

Reggio Calabria, Ottobre 2022

Tiziana Ciano