Expert Systems With Applications 296 (2026) 129111

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Revolutionizing corruption dynamics: Integrating jury influence and fractional approaches with neural network and optimal control

Aamir Farooq [a], Kamil Shah [b,*], Mohamed Anass El Yamani [c], Usman khan [d], Jamal Shah [e], Wen-Xiu Ma [a,f,g,h,*]

[a] Department of Mathematics, Zhejiang Normal University, Jinhua 321004, China
[b] School of Infromation Engineering, Qujing Normal University, Qujing 655001 Yunnan, China
[c] Department of Mathematics and Computer Science, ERMIA Team, ENSA of Tangier Abdelmalek Essaâdi University, Morocco
[d] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001 Henan, China
[e] School of Mathematics and Statistics, Central South University, Changsha, Hunan 410083, China
[f] Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[g] Department of Mathematics and Statistics, University of South Florida, Tampa, FL 33620-5700, USA
[h] Material Science Innovation and Modelling, North-West University, Mafikeng Campus, Private Bag X2046, Mmabatho 2735, South Africa

ARTICLE INFO

Keywords:
Corruption mathematical model
Fractional derivative
Reproduction number
Neural network
Optimal control
Numerical simulation

ABSTRACT

Corruption, a global problem, has a harmful effect that includes the deprivation of citizen rights, degradation of community faith in government institutions, disturbance of peace and security, misallocation of resources, and termination of employment chances. Although there have been numerous and varied attempts to address corruption, its enduring presence continues to pose a significant challenge in multiple countries. This research paper studies a compartmental mathematical model, that aims to clarify the dynamics of corruption transmission. The classification of population segments into five compartments is based on their intrinsic features, which enables monitoring the spread of corruption inside and across these segments. The reproduction number is calculated using advanced methodology to measure the potential for transmission. The stability of the model is examined, exhibiting both local and global asymptotic stability at the equilibrium point without corruption, as well as at the equilibrium point during the endemic state, based on the primary reproduction number. This approach is enhanced by utilizing neural networks to model and verify the intricate relationships associated with corruption dynamics. Moreover, the outcomes are verified by neural networks and tested with those obtained for the Atangana-Baleanu fractional model. Additionally, graphical illustrations are provided to depict the influence of the embedded parameters. Furthermore, the model is extended to investigate optimal control ways. Numerical simulations verify theoretical studies carried out with and without optimal control.

1. Introduction

Modeling real-world phenomena or systems with mathematical equations, functions, and relationships is a systematic approach used in various scientific fields. Simulating a system allows researchers to make predictions, examine behavior, and obtain insights without conducting expensive or unfeasible experiments. By capturing the system's fundamental properties through assumptions and simplifications, researchers can make predictions, examine behavior, and obtain insights without conducting expensive or unfeasible experiments. Complexity real-world procedures within domains such as the sciences, engineering, biological sciences, finance, and ecological sciences can be better understood, predicted, and optimized with the help of mathematical simulations, which can take many forms. Quantitative models have long been the go-to method for acquiring theoretical knowledge for decision-making in real-world behavioral situations, even without actual data. Many scholars are interested in developing mathematical models on nepotism dynamics because of its profound effect on global society (Panovska-Griffiths et al., 2021).

The International Monetary Fund and World Bank define corruption as using public positions for personal benefit (Bhargava, 2005). In the past two decades, corruption has become a significant barrier to progress in a developing nation (Athithan et al., 2018). Deep-rooted corruption undermines the rule of law and social fairness in numerous

* Corresponding authors.
E-mail addresses: aamirf@zjnu.edu.cn (A. Farooq), kamilshah1280@gmail.com (K. Shah), mawx@cas.usf.edu (W.-X. Ma).

https://doi.org/10.1016/j.eswa.2025.129111
Received 24 June 2024; Received in revised form 7 July 2025; Accepted 19 July 2025
Available online 22 July 2025

countries worldwide (Aliyu and Elijah, 2011). Corruption, unlike diseases and aberrations, is a pervasive phenomenon that has been historically prevalent. It spreads through various contact channels among those who are corrupted and others who are susceptible to corruption (Lemecha and Feyissa, 2018). Corruption has perpetually existed in every country and is unlikely to be eradicated shortly. Corruption is a pervasive issue that necessitates worldwide remedies (Aliyu and Elijah, 2011). In spite of several endeavors, such as anti-corruption regulations or tactics aimed at mitigating corruption in society, corruption has persistently remained a pressing issue (Hathroubi and Trabelsi, 2014) and has evolved into an epidemic in certain nations. The World Bank Group actively supports anti-corruption initiatives in the public, corporate, and civil society sectors through collaborative efforts (Kanbur, 2018; Wanless, xxxx).

Research has demonstrated that policymakers and investigators can benefit from applying mathematical models to enhance our comprehension of the propagation and management of corruption. While there is a wealth of literature on corruption's political, social, and economic aspects, more efforts need to be made to measure its changing nature attempts (Aliyu and Elijah, 2011). As a prime instance, authors developed a SIR model to explain the dynamics of corruption in (Athithan et al., 2018). They claim that stricter anti-corruption policies, including those about the media and punishment, can help reduce society's corruption. Researchers discovered an endemic, corruption-free equilibrium point along with the number of fundamental reproductions. Lemecha (Lemecha and Feyissa, 2018) developed a mathematical model that considered the impact of anti-corruption awareness and counseling in prison on corruption. An epidemiological model of corruption was developed by Nathan and Jackob for Kenya, which identifies individuals who exploit positions of power (Nathan and Jackob, 2019). By utilizing endemic and corruption-free equilibrium points as indicators, we were able to calculate the basic reproduction number. An investigation of the corruption of the population was undertaken by Abdulrahman (Abdulrahman, 2014). All three equilibrium points have been calculated: the equilibrium point without corruption, the equilibrium point with endemicity, and the basic reproduction number (BRN). Based on numerical simulations, it has been determined that corruption can be reduced to a reasonable level, although it cannot be completely eradicated. Using the stability theory of differential equations, Alemneh (Athithan et al., 2018) has formulated a nonlinear deterministic model to study corruption dynamics. Understanding the dynamics of corruption, implementing effective media campaigns, cracking down on corrupt individuals, and enforcing strict punishments can significantly mitigate the impact of corruption.

Understanding real-world situations through non-integer-order derivatives is an advanced method in fractional calculus. Researchers are interested in it because of its practical applications in solving current real-world problems across different countries. Compared to integer-order models, fractional-order models excel in many areas, including precision, adaptability, rapid identification, memory retention, and the enhancement of healthcare techniques. In a recent study (Teklu, 2024), researchers delved into the most effective control mechanisms for managing corruption dynamics by utilizing fractional order derivatives. In this study, we developed and examined an optimum control strategy for the Caputo case fractional order corruption spreading model. An investigation was conducted to determine equilibrium points of the fractional order corruption model. The basic reproduction number of the model was calculated using the approach of the next-generation matrix. Picard-Lindelöf analysis was used to examine the existence and uniqueness of solutions to the proposed fractional order model. (Anjam et al., 2024) showed the results of stability evaluation of the changing patterns associated with corruption under fractional-order measures. Mansour et al. (Abdulwasaa et al., 2024) investigated an empirical and numerical evaluation of the corruption and poverty model utilizing Caputo-type fractional differential equations. To explore more uses of fractional derivatives, see the following (Akgül et al., 2024; Adamu et al., xxxx; Ali et al., xxxx; Awadalla et al., 2023) and the corresponding references.

Machine learning (ML) is a scientific discipline concerned with the design of algorithms and models that enable computers to improve their performance on a task through experience, typically by learning from data. Located at the intersection of data science, artificial intelligence, and computer science, this subject is of keen interest. Significant progress in machine learning has been driven by the development of new learning algorithms and theories in combination with the widespread availability of online data and affordable computing power. Various industries, such as medical care, production, teaching, modeling of finances, policing, and marketing, increasingly use data-intensive machine-learning approaches in science, technology, and business to make evidence-based judgments. ML, which stands for machine learning, is a field of artificial intelligence centered around living creatures' learning processes. The ML technique primarily focuses on developing algorithms that can learn from machine-readable data. The primary domains encompassed by machine learning are software applications, complex systems that are difficult to build, and the field of data mining. It is a collection of diverse algorithms capable of performing multidimensional, dynamic, nonparametric classification and regression tasks. Neural networks, support vector machines, self-organizing maps, decision trees, random forests, case-based reasoning, and genetic algorithms are some examples. The widespread application of ML-based methods in science and engineering can be credited to their exceptional modeling capabilities. The use of machine learning in mathematical modeling has significantly transformed multiple aspects of model building and analysis. ML enables the building of complex models that can effectively handle large and diverse datasets, substantially impacting data-driven model development. This is especially beneficial for real-world systems with intricate patterns those typical statistical techniques may fail to capture. The link between ML and mathematical modeling is profound, fundamentally changing how problems are solved and data are analyzed. Mathematical models serve as a theoretical framework for describing the links and patterns within systems. Nevertheless, conventional models frequently need help - dealing with nuanced structures in complicated real-world data. This is where ML comes into play, providing data-driven flexibility and the capacity to identify patterns in large datasets. Ali Husain et al. (Husnain et al., 1988) investigated the potential of artificial intelligence (AI) and Machine Learning in addressing the challenges presented by COVID-19. The researchers examined how these technologies may be applied to tackle several concerns arising from the pandemic. The study examines many applications, such as forecasting outbreaks and efficiently allocating healthcare resources, to showcase the substantial capacity of AI and machine learning in reducing the effects of COVID-19. Huang et al. (Huang et al., 2024) investigated the use of machine learning-based K-means clustering to improve the identification of financial fraud. Their method utilizes unsupervised learning to detect unusual transaction patterns, enhancing the precision and effectiveness of fraud detection systems. Sanz et al. (López Iturriaga et al., 2022) presented a neural network model that accurately forecasts instances of corruption in public procurement procedures. The model sought to improve transparency and accountability by identifying potential corrupt activities.

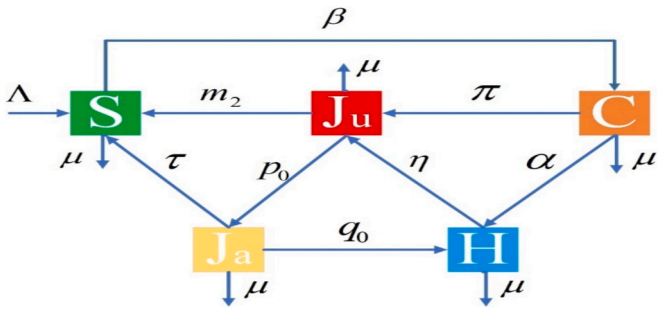

Fig. 1. Schamtic diagram for Corruption Mathematical Model.

For additional applications, please refer to the following papers and their corresponding references (El Yamani et al., 2023; Adedeji, 2023; Meskher et al., 2023; Akakuru et al., 2023).

This study addresses a critical gap in the literature concerning the complex interplay between judicial decisions and societal behavior, particularly focusing on how jury verdicts influence both corrupt individuals and those who choose to maintain their integrity. While prior models of corruption dynamics have provided foundational insights, they often lack the ability to account for long-term memory effects and behavioral inertia, which are essential in social and psychological systems. To overcome this limitation, we develop a novel dynamic model incorporating the Atangana–Baleanu fractional derivative in the Caputo sense, characterized by a non-singular Mittag-Leffler kernel. This fractional operator introduces a memory-dependent framework, enabling a more realistic depiction of how past events influence future decisions, especially in legal and moral contexts. By extending the classical corruption model into the fractional domain, we capture the persistence and gradual evolution of behavioral states, which are often oversimplified in integer-order systems. Our model is designed to simultaneously monitor the time-evolution of two distinct populations: the corrupt and the incorruptible. This dual-population perspective reveals the contrasting trajectories influenced by judicial scrutiny, especially the deterrent or reinforcing effects of jury outcomes. To solve and simulate the proposed model, we develop a new numerical scheme tailored for fractional differential equations, ensuring stability and accuracy in the presence of memory-dependent dynamics. Furthermore, we incorporate artificial neural networks (ANNs) as a computational tool to validate the behavior of our model and to approximate solutions in high-dimensional parameter spaces where traditional methods may falter. Beyond descriptive modeling, the study also tackles optimal control theory in the context of corruption mitigation. We formulate a set of time-dependent control strategies aimed at minimizing the number of corrupt individuals and maximizing societal integrity. The theoretical framework includes proof of existence and uniqueness of the optimal control, as well as detailed numerical simulations demonstrating the cost-effectiveness and dynamic impact of different policy interventions. Through this multifaceted approach spanning fractional calculus, numerical analysis, neural networks, and optimal control we offer a comprehensive and innovative framework to better understand the societal consequences of jury decisions and propose mathematically informed strategies for legal and policy reform.

2. Mathematical Preliminaries

To explain fractional-order, this section contains some important ideas and prepositions.

Definition 2.1.1. (*In Atangana-Baleanu time fractional operators, an Atangana-Baleanu continuous function* $f(t) \in C^n[0,T]$ *with fractional parameter* ρ *is represented as follows:*)

$$ {}^{ABC}_{a}\wp^{\rho}_{t}\psi(t) = \frac{N(\rho)}{1-\rho}\int_a^t E_{\gamma}\left(\frac{-\rho(t-\omega)^{\rho}}{1-\rho}\right)\psi'(\omega)d\omega, \text{For } 0<\kappa<1. \tag{1}$$

The normalization function is $N(\rho)$, while the Mittag-Leffler function is $E_{\rho}(.)$.

Definition 2.1.2. (*A fractional derivative Atangana-Baleanu (ABC) has the following integral operator:*)

$$ {}^{ABC}_{a}\boldsymbol{I}^{\rho}_{t}f(t) = \frac{1-\rho}{N(\rho)}f(t) + \frac{\rho}{N(\rho)\Gamma(\rho)}\int_a^t (t-\zeta)^{\rho-1}f(\zeta)d\zeta \tag{2}$$

Definition 2.1.3. (*According to Toufik and Atangana, fractional order ODEs can be numerically solved using the following approach:*) *Let's consider a fractional ordinary differenctial equation* (*ODE*).

$$ {}^{ABC}\wp^{\rho}_{t}\varsigma(t) = \psi(t,\varsigma(t)) \, with \, \varsigma(0) = \varsigma_0 \tag{3}$$

The numerical structure of equation (3) is as follows:

$$\begin{aligned}\varsigma_{\dagger+1} &= \varsigma_0 + \frac{1-\rho}{N(\rho)}\psi(t_r,\varsigma(t_r)) \\ &+ \frac{\rho}{N(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{array}{l}\frac{h^{\rho}\psi(t_w,\varsigma(t_w))}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\} \\ \quad -\frac{h^{\rho}\psi(t_w,\varsigma(t_w))}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{array}\right],\end{aligned} \tag{4}$$

3. Mathematical modeling

Mathematical model of the corruption, the total population is

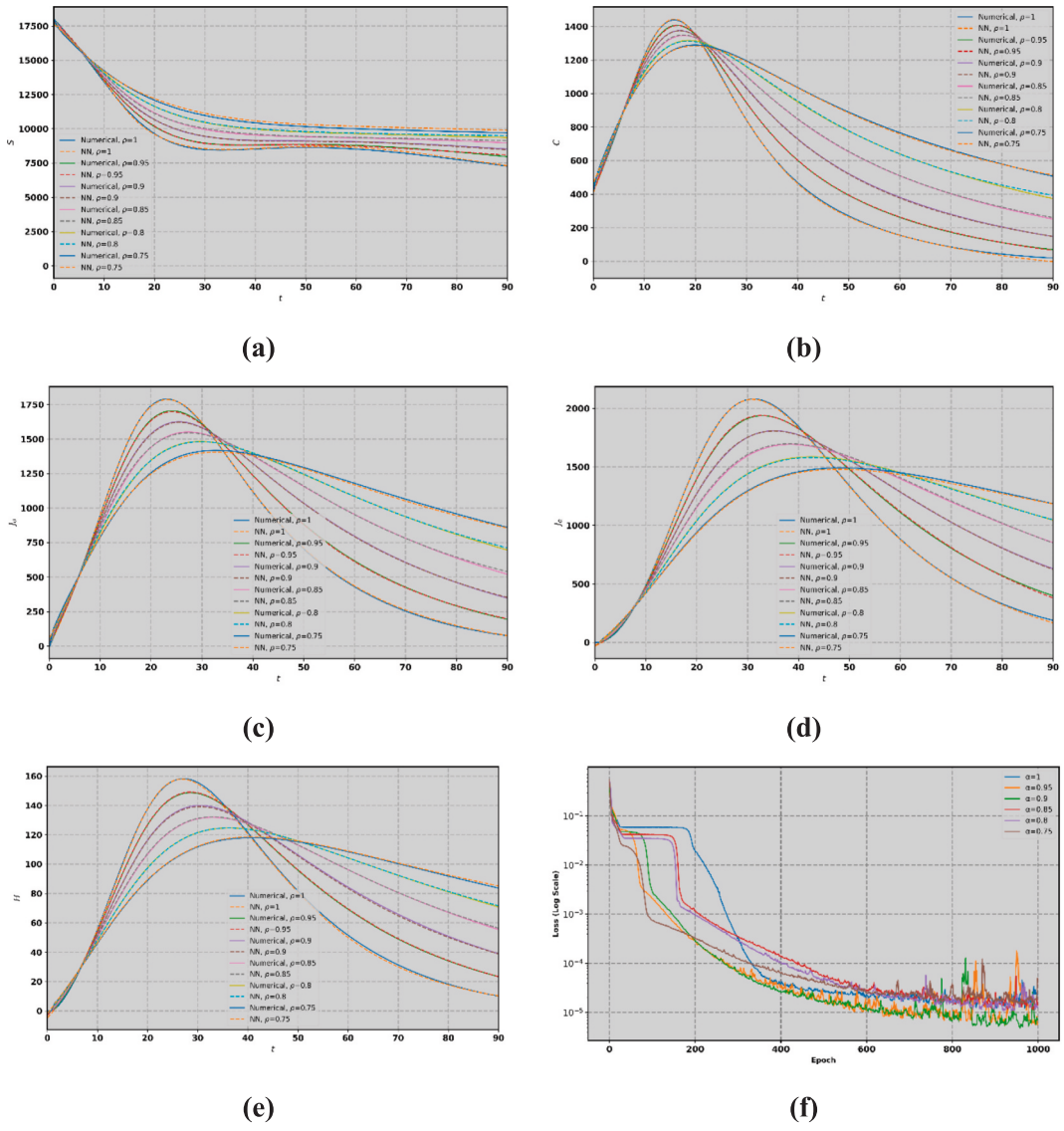

Fig. 2. Fig. 2 (a-f) illustrates how the fractional parameter ρinteracts with other parameters while keeping the fractional parameter constant. The fractional model offers a variety of solutions, unlike classical models, which only have one solution. To obtain the best fit between theoretical and real data, it is necessary to adjust the fractional parameter appropriately. Compared to classical models, fractional models offer a more generalized explanation of physical phenomena. In this case, the fractional differential operator of Atangana-Baleanu is more appropriate since it allows for better explanation of the problem related to corruption. Simulated scenarios are presented both for fractional-order $(0 < \rho < 1)$ and integer-order $(\rho = 1)$ scenarios allowing for a clear comparison. It is worth noting that when $\rho \to 1$, the model transitions towards the classical order. For training, testing and validation for varyingρ see Table 4.

Table 1
Discerption of model variables.

Variables	Discerption
$S(t)$	Susceptible class
$C(t)$	Corruption class
$J_u(t)$	Judiciary class
$J_a(t)$	Jail class
$H(t)$	Honest class

Table 2
Discerption of model Parameters.

Parameters	Discerption
Λ	Recruitment rate of susceptible individuals
β	Transmission rate of corruption
μ	Natural mortality rate of S, C, J_u, J_a, and H
m_2	Rate of individuals moving from the J_u to the S.
τ	Rate of individuals moving from the J_u to the J_a
π	Trial rate of corrupt individual
α	Rate of individuals that joins H from C
η	Rate of individuals entering the J_u from the H
p_o	Rate of individuals leaving the J_u to enter the J_a
q_o	Rate at which J_a individuals move to H class.

denoted by $N(t)$. It is further subdivided into 'susceptible' $S(t)$, 'corruption' $C(t)$ 'judiciary' $J_u(t)$ 'jail' $J_a(t)$ 'Honest' $H(t)$. The variables and parameters in the system are all non-negative. Recruitment rate of the corrupte individual is denoted by Λ. Contact rate of corruption is denoted by β. Natural mortality rate is denoted by μ. Rate of individuals

moving from the J_u to the susceptible class m_2 and rate of individuals moving from the J_u to the J_a is denoted by τ. Trial rate of corrupt individual π, and rate of individuals that joins H from C is denoted by α. Rate of individuals entering the J_u from the H is denoted by η, and rate of individuals leaving the J_u to enter the J_a is denoted by p_o. Rate at which J_a individuals move to H class is denoted by q_o. Flow chart of the corruption transmission model present in Fig. 1. (See Fig. 2).

This section will begin by modifying the traditional corruption transmission model, which can be defined as follows. For detailed explanations of the model's parameters and variables, please refer to Table 1 and Table 2.

with

$$\left.\begin{aligned}
\frac{dS}{dt} &= \Lambda - (\beta C + \mu)S + m_2 J_u + \tau J_a \\
\frac{dC}{dt} &= \beta SC - (\pi + \alpha + \mu)C \\
\frac{dJ_u}{dt} &= \pi C + \eta H - (p_o + m_2 + \mu)J_u \\
\frac{dJ_a}{dt} &= p_o J_u - (q_o + \tau + \mu)J_a \\
\frac{dH}{dt} &= q_o J_a + \alpha C - (\eta + \mu)H,
\end{aligned}\right\} \tag{5}$$

$$S(0) = s_o,\ \ C(0) = c_o,\ \ J_u(0) = (j_u)_o,\ \ J_a(0) = (j_a)_o,\ \ H(0) = h_o \tag{6}$$

In contrast to classical mathematical models, fractional order mathematical models are able to capture such phenomena due to memory effects of fractional order derivatives. Rather than the classical derivative, the Atangana-Baleanu (AB) operator of fractional order ${}^{ABC}\wp_t^{\kappa}$ is used instead. In order to explore the dynamics of the disease in the presence of memory effects, the fractional operator is included in the model. A fractional derivative of AB has an order parameter $\rho \in (0,1]$. A fractional order differential equation of ρ governs the problem.

$$\begin{aligned}
{}^{ABC}\wp_t^{\kappa}S(t) &= \Lambda - (\beta C + \mu)S + m_2 J_u + \tau J_a \\
{}^{ABC}\wp_t^{\kappa}C(t) &= \beta SC - (\pi + \alpha + \mu)C \\
{}^{ABC}\wp_t^{\kappa}J_u(t) &= \pi C + \eta H - (p_o + m_2 + \mu)J_u \\
{}^{ABC}\wp_t^{\kappa}J_a(t) &= p_o J_u - (q_o + \tau + \mu)J_a \\
{}^{ABC}\wp_t^{\kappa}H(t) &= q_o J_a + \alpha C - (\eta + \mu)H
\end{aligned} \tag{7}$$

4. Mathematical analysis

Within this section, we will thoroughly examine the mathematical analysis of system (7).

4.1. Non-Negativity and Boundedness of The fractional model

In this part, we illustrate that model (7) is non-negative. We start with the following theorem.

Theorem 4.1. Let $(S(0),\ C(0),\ J_u(0), J_a(0),\ H(0)) > 0$ then the solution set $S(t), C(t), J_u(t), J_a(t), H(t)$ of the system of the equation (7) is non-negative for all t > 0

$$\begin{gathered}
{}^{ABC}\wp_t^{\kappa}S(t)\big|_{s=0} = \Lambda + m_2 J_u + \tau J_a \geq 0,\ {}^{ABC}\wp_t^{\kappa}C(t)\big|_{C=o} = \beta SC \geq 0 \\
{}^{ABC}\wp_t^{\kappa}J_u(t)\big|_{J_{u=0}} = \pi C + \eta H \geq 0,\ {}^{ABC}\wp_t^{\kappa}J_a(t)\big|_{J_a=0} = p_o J_u \geq 0 \\
{}^{ABC}\wp_t^{\kappa}H(t)\big|_{H=0} = q_o J_a + \alpha C \geq 0
\end{gathered} \tag{8}$$

Hence the present model (7) is non-negative.

Lemma 4.1. (*The region* $\Pi \in R_+^5$ *is positively invariant for the model (7) with non-negative initial conditions in* R_+^5.) *Proof: It follows from summing up the equations of the model (7) that.*

$$\frac{dN(t)}{dt} = \Lambda - \mu N(t). \tag{9}$$

Hence,

$$\frac{dN(t)}{dt} \leq 0 \text{ if } N(0) \geq 0 \tag{10}$$

Thus

$$\frac{dN(t)}{dt} \leq N(0)e^{-\mu t} + \frac{\Lambda}{\mu}(1 - e^{-\mu t}). \tag{11}$$

Therefore, the region Π is bounded. Furthermore, if $N(0) \geq \frac{\Lambda}{\mu}$ then either the solution enters Π in finite time or $N(t)$ approaches $\frac{\Lambda}{\mu}$ asymptotically. Hence, the region Π attracts all solutions in R_+^5.

4.2. Model equilibria

Corruption dynamics can be understood through the equilibrium points of the Atangana-Baleanu fractional corruption model (7). For assessing the equilibria of this AB fractional model, we set

$$\left.\begin{gathered}
{}^{ABC}\wp_t^{\kappa}S(t) = 0,\quad {}^{ABC}\wp_t^{\kappa}C(t) = 0,\ {}^{ABC}\wp_t^{\kappa}J_u(t) = 0, \\
{}^{ABC}\wp_t^{\kappa}J_a(t) = 0,\quad {}^{ABC}\wp_t^{\kappa}H(t) = 0.
\end{gathered}\right\} \tag{12}$$

The following corruption-free equilibrium (CFE) of the above model can be obtained when solving the above system (12).

$$E_o = \left(\frac{\Lambda}{\mu}, 0, 0, 0, 0\right) \tag{13}$$

By utilizing well-known next-generation concepts, we can determine that the basic reproduction number is the following:

$$R_o = \frac{\Lambda\beta}{\mu(\pi + \alpha + \mu)} \tag{14}$$

Furthermore, whenever $R_o > 1$ is present, the problem has an endemic equilibrium (EE)

$$\left.\begin{gathered}
S^* = \frac{\chi_2}{\beta},\ C^* = \frac{(\beta\Lambda - \mu\chi_2)((\eta+\mu)(\mu+\tau)(\mu+p_o) + \mu(\eta+\mu+p_o)q_o + (\eta+\mu)m_2(\mu+\tau+q_o))}{\beta\mu\chi_1} \\
J_u^* = \frac{-((-\beta\Lambda + \mu\chi_2)(\alpha\eta + \pi(\eta+\mu))(\mu+\tau+q_o))}{\beta\mu\chi_1},\ J_a^* = \frac{-((-\beta\Lambda + \mu\chi_2)(\alpha\eta + \pi(\eta+\mu))p_o)}{\beta\mu\chi} \\
H^* = \frac{(\beta\Lambda - \mu\chi_2)(\alpha(\mu+\tau)(\mu+m_2+p_o) + (\alpha(\mu+m_2) + (\pi+\alpha)p_o)q_o)}{\beta\mu\chi_1}.
\end{gathered}\right\} \tag{15}$$

Where,

$$\chi_1 = (\chi_2(\eta+\mu)(\mu+\tau+q_o) + (\alpha+\eta+\mu)m_2(\mu+\tau+q_o) + p_o(\chi_2(\eta+\mu) + (\alpha+\eta+\mu)\tau + \chi_2 q_o)),$$

$$\chi_2 = (\pi+\alpha+\mu).$$

4.3. Local stability analysis for corruption free equilibrium

Theorem 4.3.1. A local asymptotically stable free equilibrium exists when $R_o < 1$, while an unstable equilibrium exists when $R_o > 1$.

$$J(DFE) = \begin{bmatrix} -\mu & -S^o\beta & m_2 & \tau & 0 \\ 0 & -\pi-\alpha+S^o\beta-\mu & 0 & 0 & 0 \\ 0 & \pi & -(\mu+m_2+p_o) & 0 & \eta \\ 0 & 0 & p_o & -(\mu+\tau+q_o) & 0 \\ 0 & \alpha & 0 & q_o & -(\eta+\mu) \end{bmatrix} \tag{16}$$

$$|J(DFE) - M| = 0 \tag{17}$$

$$\begin{vmatrix} -\mu-M & -S^o\beta & m_2 & \tau & 0 \\ 0 & -(\pi+\alpha-S^o\beta+\mu)-M & 0 & 0 & 0 \\ 0 & \pi & -(\mu+m_2+p_o)-M & 0 & \eta \\ 0 & 0 & p_o & -(\mu+\tau+q_o)-M & 0 \\ 0 & \alpha & 0 & q_o & -(\eta+\mu)-M \end{vmatrix} = 0 \tag{18}$$

$$(-\mu-M)\left(M^4 + a_1M^3 + a_2M^2 + a_3M + a_4\right) = 0 \tag{19}$$

Where,

$$\begin{aligned} a_1 &= 1 \\ a_2 &= \alpha\eta - S\beta\eta + 3\alpha\mu - 3S\beta\mu + 3\eta\mu + 6\mu^2 \\ &\quad + (\alpha - S\beta + \eta + 3\mu)\tau + \pi(\eta + 3\mu + \tau) \\ &\quad + (\pi + \alpha - S\beta + \eta + 3\mu + \tau)(m_2 + p_o) \\ &\quad + (\pi + \alpha - S\beta + \eta + 3\mu + m_2 + p_o)q_o, \\ a_3 &= \mu\left(2(\pi+\alpha-S\beta)\eta + 3(\pi+\alpha-S\beta+\eta)\mu + 4\mu^2\right) \\ &\quad + \left((\pi+\alpha-S\beta)\eta + 2(\pi+\alpha-S\beta+\eta)\mu + 3\mu^2\right)\tau \\ &\quad + \begin{pmatrix} \alpha\eta - S\beta\eta + 2\alpha\mu - 2S\beta\mu + 2\eta\mu + 3\mu^2 \\ +(\alpha - S\beta + \eta + 2\mu)\tau + \pi(\eta + 2\mu + \tau) \end{pmatrix}(m_2 + p_o) \\ &\quad + \begin{pmatrix} (\pi+\alpha-S\beta)\eta + 2(\pi+\alpha-S\beta+\eta)\mu + 3\mu^2 \\ +(\pi+\alpha-S\beta+\eta+2\mu)m_2 + (\pi+\alpha-S\beta+2\mu)p_o \end{pmatrix}q_o, \\ a_4 &= (\pi+\alpha+\mu)\begin{pmatrix} (\eta+\mu)(\mu+\tau)(\mu+p_o) + \mu(\eta+\mu+p_o)q_o \\ +(\eta+\mu)m_2(\mu+\tau+q_o) \end{pmatrix}(1-R_o). \end{aligned} \tag{20}$$

Clearly, if $R_o < 1$, then $a_i > 0$, for $i = 1, 2, 3, 4$, also it can be easily verified through some algebraic manipulations through that $a_1a_2a_3 > a_3^2 + a_1^2a_4$. Thus, implementing Routh-Hurwitz criteria the disease-free equilibrium (DFE) of the system (7) is the local stability about the equilibrium point.

4.4. Global stability analysis for corruption free equilibrium

Theorem 4.4.1. (*A global asymptotically stable free equilibrium exists when $R_o < 1$, while an unstable equilibrium exists when $R_o > 1$.*) *Proof: Consider the following lypanouv function:*

$$L = \gamma_1 C + \gamma_2 J_u + \gamma_3 J_a, \tag{21}$$

$$\begin{cases} \dfrac{dL}{dt} = (\gamma_1\beta S - \gamma_1\pi - \gamma_1\alpha - \gamma_1\mu + \gamma_2\pi)C - (\gamma_2 p_o + \gamma_2 m_2 + \gamma_2\mu - \gamma_3 p_o)J_u \\ \qquad\qquad - (\gamma_3 q_o + \gamma_3\tau + \gamma_3\mu)J_a - \gamma_2\eta H, \end{cases} \tag{22}$$

$$\frac{dL}{dt} \le (\gamma_1\beta S - \gamma_1\pi - \gamma_1\alpha - \gamma_1\mu + \gamma_2\pi)C, \tag{23}$$

$$\frac{dL}{dt} \le \left((\pi+\alpha+\mu)\left(\frac{\gamma_1\beta S}{(\pi+\alpha+\mu)} - \gamma_1\right) + \gamma_2\pi\right)C, \tag{24}$$

$$\frac{dL}{dt} \le (\pi+\alpha+\mu)(1-R_0)C, \tag{25}$$

where

$\gamma_1 = -1,\ \gamma_2 = 0,\ \gamma_3 = 0$. It implies that,

$$\frac{dL(t)}{dt} < 0 \text{ if } R_0 < 1 \tag{26}$$

$$\frac{dL(t)}{dt} = 0; \text{ if } C = 0 \text{ or } R_0 = 1. \tag{27}$$

According to the Lassale's invariance principle, disease-free equilibrium (DFE) is globally asymptotically stable whenever $R_0 < 1$.

4.5. Local stability analysis for corruption endemic equilibrium

Theorem 4.5.1. A local asymptotically stable endemic equilibrium exists when $R_o > 1$, otherwise an unstable equilibrium exists

Proof: The jacobian matrix of model (7) is given as follow:

Table 3
Parameter values of Corruption model with estimated and source.

S.no	Parameters	Estimated values	Source
1.	β	0.008234	Assumed
2.	m_2	0.0084	Assumed
3.	π	0.6	Assumed
4.	τ	0.1	Assumed
5.	α	0.03	(Alemneh, 2020)
6.	η	0.001009	Assumed
7.	p_o	0.1620	(Alemneh, 2020)
8.	μ	0.0160	(Alemneh, 2020)
9.	q_o	0.007	(Alemneh, 2020)

$$J(E^*) = \begin{pmatrix} -(\beta C+\mu) & -\beta S & m_2 & \tau & 0 \\ \beta C & \beta S-(\pi+\alpha+\mu) & 0 & 0 & 0 \\ 0 & \pi & -(p_o+m_2+\mu) & 0 & \eta \\ 0 & 0 & p_o & -(q_o+\tau+\mu) & 0 \\ 0 & \alpha & 0 & q_o & -(\eta+\mu) \end{pmatrix}. \tag{28}$$

$$|J(E^*) - M| = 0. \tag{29}$$

$$\begin{vmatrix} -(\beta C+\mu)-M & -\beta S & m_2 & \tau & 0 \\ \beta C & \beta S-(\pi+\alpha+\mu)-M & 0 & 0 & 0 \\ 0 & \pi & -(p_o+m_2+\mu)-M & 0 & \eta \\ 0 & 0 & p_o & -(q_o+\tau+\mu)-M & 0 \\ 0 & \alpha & 0 & q_o & -(\eta+\mu)-M \end{vmatrix} = 0. \tag{30}$$

$$M^5 + b_1M^4 + b_2M^3 + b_3M^2 + b_4M + b_5 = 0. \tag{31}$$

Clearly, if $b_i > 0$, for $i = 1,2,3,4,5$. Thus, by Routh-Hurwitz criteria of higher order polynomial the endemic equlibrium of the system (7) is local asymptotically stable (LAS) about the equilibrium point.

4.6. Global stability analysis for corruption endemic equilibrium

Theorem 4.6.1. (*A global asymptotically stable endemic equilibrium exists when $R_o > 1$, otherwise an unstable equilibrium exists.*) *Proof: Consider the following lypanouv function:*

$$L(S,C,J_u,J_a,H) = \left(S - S^* - S^*\ln\frac{S}{S^*}\right) + \left(C - C^* - C^*\ln\frac{C}{C^*}\right) + \left(J_u - J_u^* - J_u^*\ln\frac{J_u}{J_u^*}\right) + \left(J_a - J_a^* - J_a^*\ln\frac{J_a}{J_a^*}\right) + \left(H - H^* - H^*\ln\frac{H}{H^*}\right). \tag{32}$$

$$\frac{dL}{dt} = \left(1-\frac{S^*}{S}\right)(\Lambda - (\beta C+\mu)S + m_2J_u + \tau J_a) + \left(1-\frac{C^*}{C}\right)(\beta SC - (\pi+\alpha+\mu)C) + \left(1-\frac{J_u^*}{J_u}\right)(\pi C + \eta H - (p_o+m_2+\mu)J_u) + \left(1-\frac{J_a^*}{J_a}\right)(p_oJ_u - (q_o+\tau+\mu)J_a) + \left(1-\frac{H^*}{H}\right)(q_oJ_a + \alpha C - (\eta+\mu)H). \tag{33}$$

$$\frac{dL}{dt} \le \zeta - \gamma. \tag{34}$$

$$\begin{cases} \zeta = \Lambda + m_2J_u + \tau J_a + \beta CSS^* + \mu S^* + \beta SC + \pi C^* + \alpha C^* \\ \quad + \mu C^* + \pi C + \eta H + p_oJ_u^* + m_2J_u^* + \mu J_u^* + p_oJ_u \\ \quad + q_oJ_a^* + \tau J_a^* + \mu J_a^* + q_oJ_a + \alpha C + \eta H^* + \mu H^*. \\ \gamma = \beta CS + \mu S + \Lambda\frac{S^*}{S} + m_2J_u\frac{S^*}{S} + \tau J_a\frac{S^*}{S} + \pi C + \alpha C + \mu C \\ \quad + \beta SC^* + p_oJ_u + m_2J_u + \mu J_u + \pi C\frac{J_u^*}{J_u} + \eta H\frac{J_u^*}{J_u} + q_oJ_a \\ \quad + \tau J_a + \mu J_a + p_oJ_u\frac{J_a^*}{J_a} + \eta H + \mu H + q_oJ_a\frac{H^*}{H} + \alpha C\frac{H^*}{H}. \end{cases} \tag{35}$$

If $\zeta < \gamma$ then $\frac{dL}{dt} < 0$.

And $\frac{dL}{dt} = 0$ if $S = S^*,\ C = C^*,\ J_u = J_u^*,\ J_a = J_a^*,\ H = H^*$. So by invariance principle endemic equilibrium point is globally asymptotically stable.

4.7. Numerical simulation

This part employs numerical simulations to validate the derived analytical conclusions. To generate an estimate with the fractional

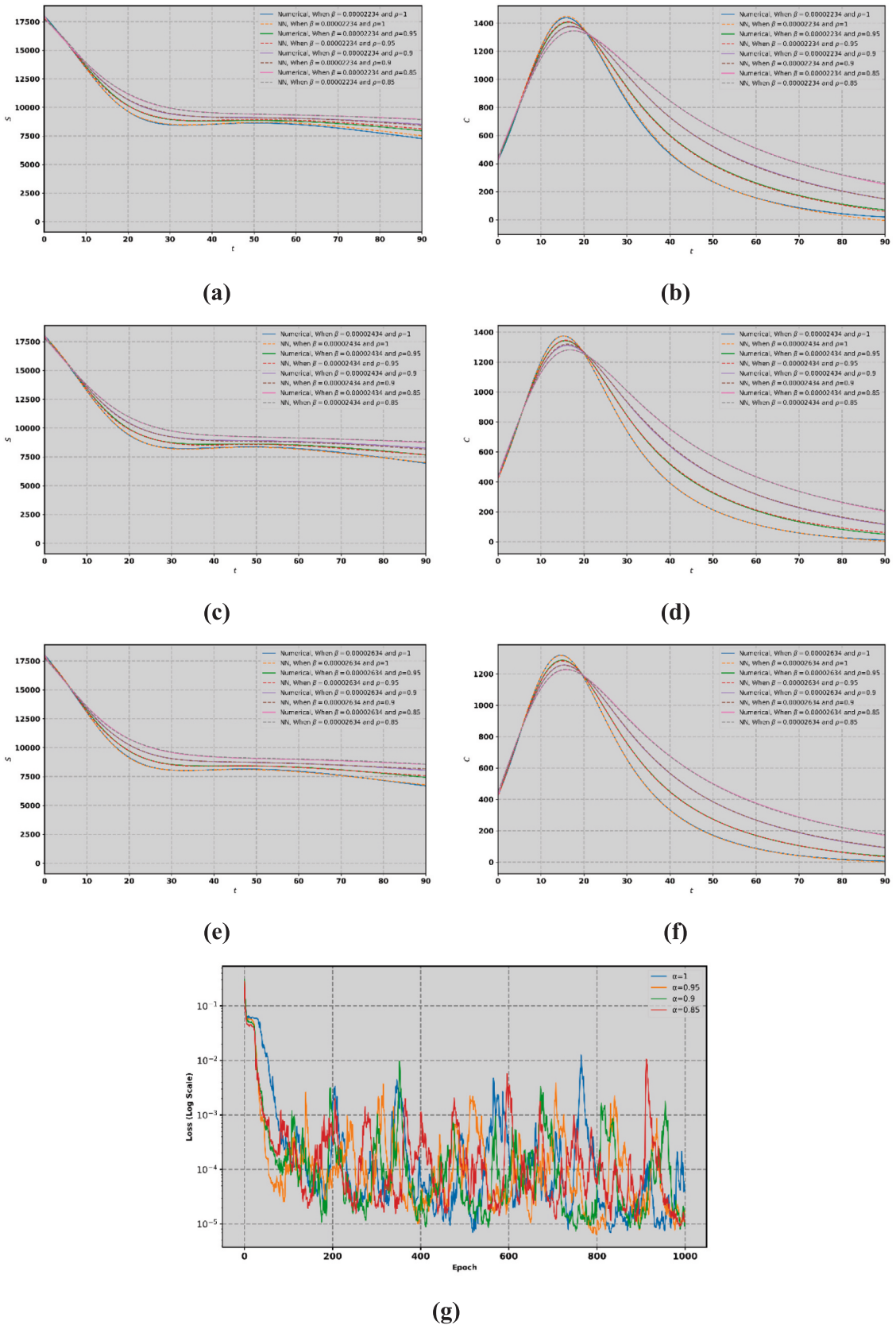

Fig. 3. Present the effect of contact rate β on susceptible and corrupted class for different values of fractional parameter$\alpha = 1$, $\alpha = 0.95$, $\alpha = 0.90$, $\alpha = 0.85$.

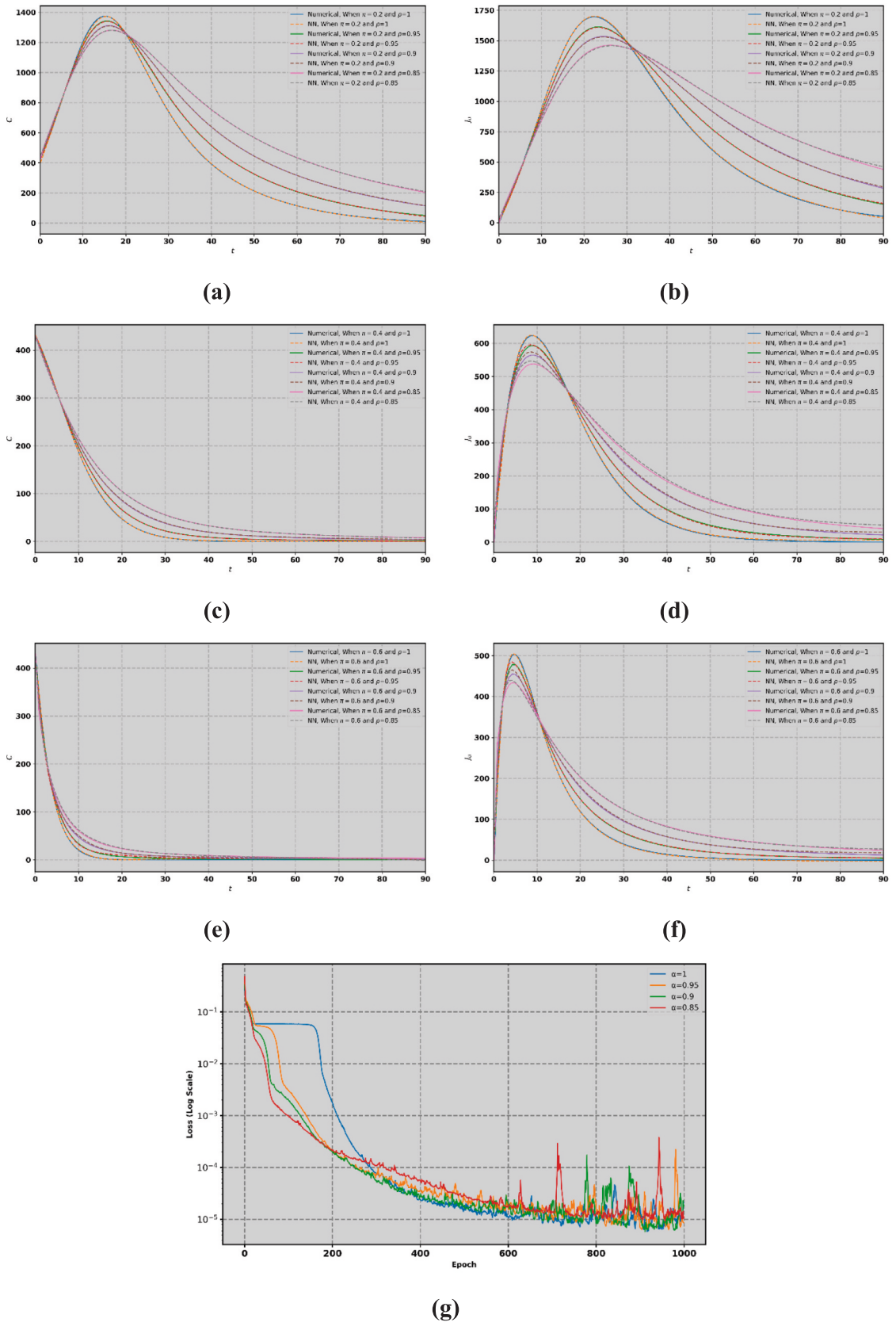

Fig. 4. Effect of π on corrupted and jury class for different values of fractional parameter$\rho = 1$, $\rho = 0.95$, $\rho = 0.90$, $\rho = 0.85$.

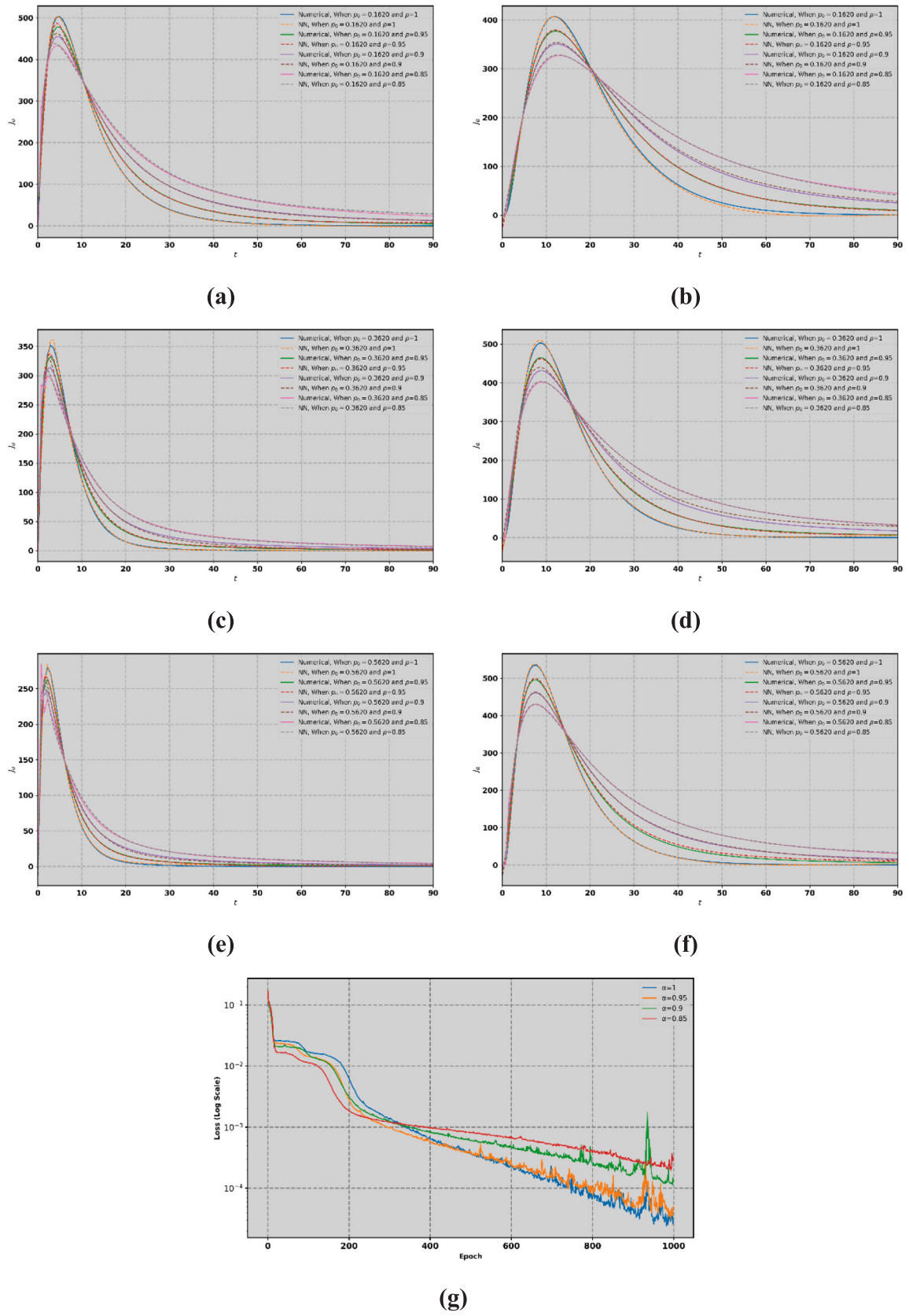

Fig. 5. Effect of P_o on jury and jail class for different values of fractional parameter $\rho = 1$, $\rho = 0.95$, $\rho = 0.90$, $\rho = 0.85$.

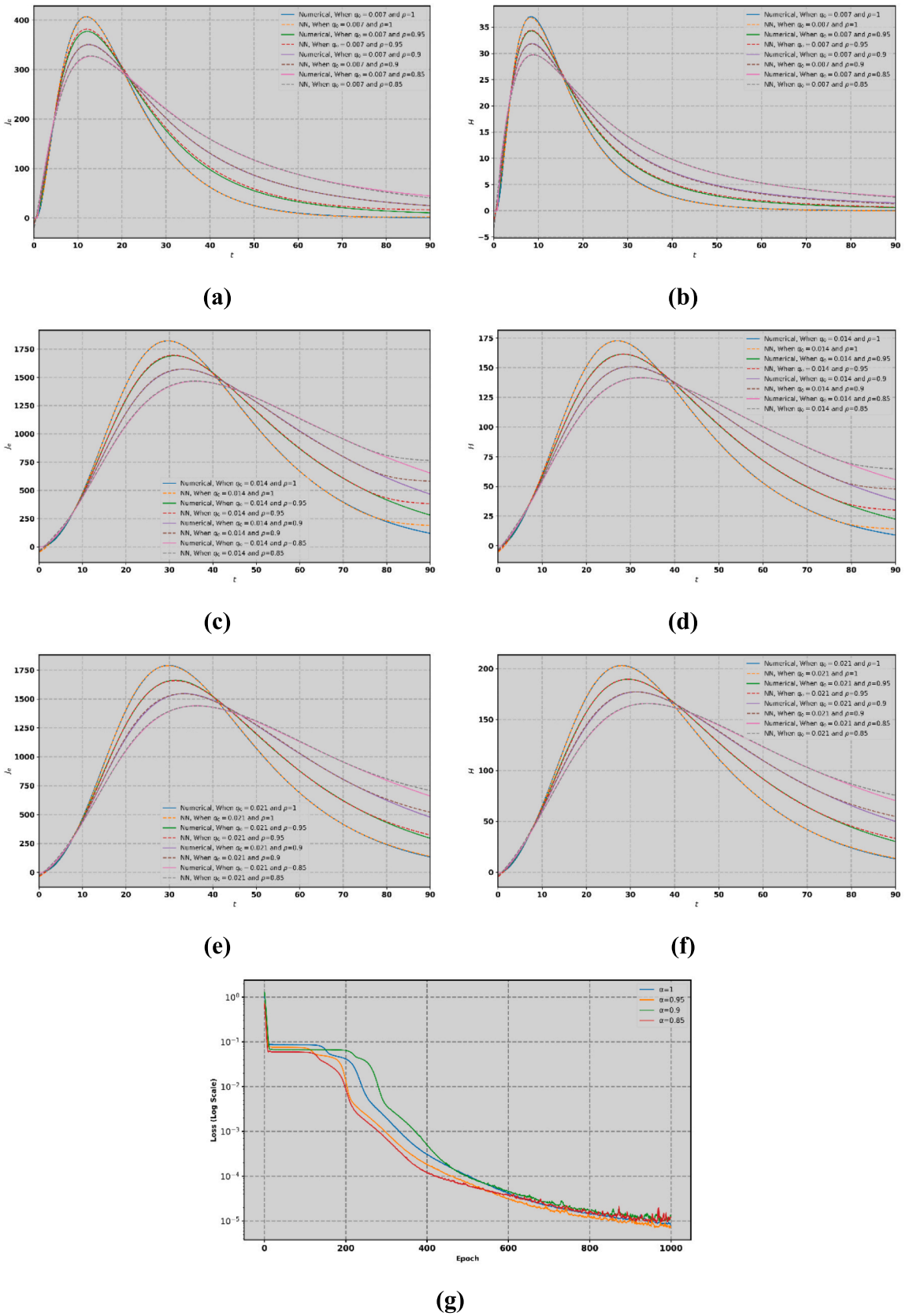

Fig. 6. Effect of q_o on jail and honest class for different values of fractional parameter $\rho = 1$, $\rho = 0.95$, $\rho = 0.90$, $\rho = 0.85$.

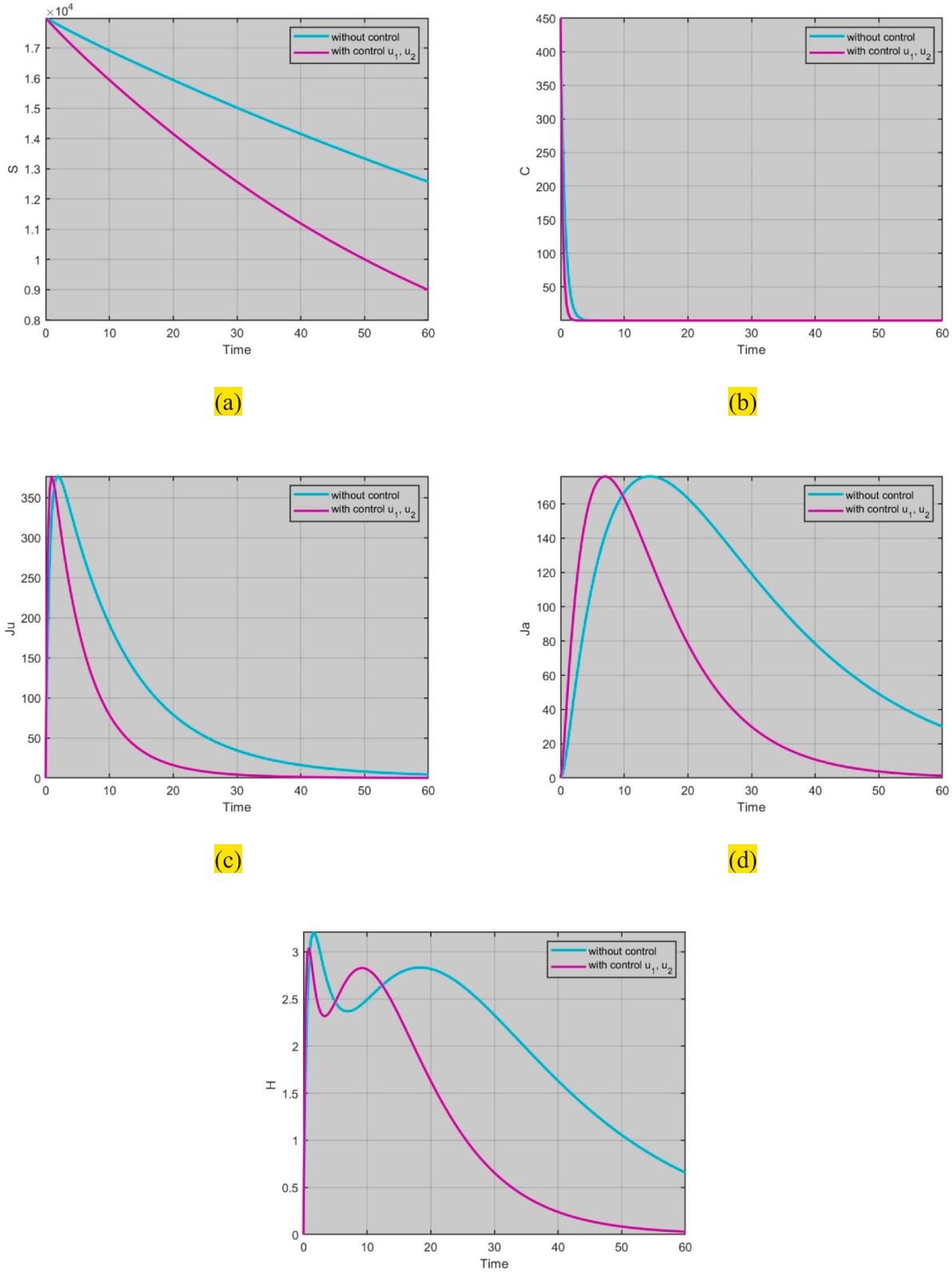

Fig. 7. Optimal control of corruption transmission mathematical model.

Table 4
Mean square error of parameter ρ.

Parameter ρ	Mean Square Error			Gradients	Epoch	Time (seconds)
	Training	Validation	Testing			
1	0.000943432159545584	0.0007448491444047045	0.0007448491444047045	0.6748254	500	88.06
0.95	0.0005296614037720976	0.0001893209037621992	0.0001531820355313538	0.7408396	500	83.84
0.90	0.0010274258824617152	0.0003681608282477693	0.00033809435648091265	0.641466	500	83.74
0.85	0.0020721940677255334	0.0009036478583356163	0.001055425740595258	−12.968877	500	50.25
0.80	0.0041479291787438494	0.0016889991068337975	0.0019164129823105683	2.231218	500	84.59
0.75	0.003772679954454488	0.0006044005542409352	0.0006158020806977647	−1.1609964	500	84.73

Table 5
Mean square error of parameter β.

Parameter β	Mean Square Error			Gradients	Epoch	Time (seconds)
	Training	Validation	Testing			
0.00002234	0.0010802350058999046	0.0007967403664745504	0.0007682544321693308	−0.7140956	500	90.13
0.00002434	0.0007296080504830539	0.0002273724741482036	0.0002358294628421942	18.364973	500	83.86
0.00002634	0.0011175552501075763	0.0007877555346312376	0.0007218840699250075	−0.4774780	500	83.99

Table 6
Mean square error of parameter π.

Parameter π	Mean Square Error			Gradients	Epoch	Time (seconds)
	Training	Validation	Testing			
0.2	0.0005863087903416389	0.0001478430876666182	0.0001481208662029224 7	0.02573235	500	48.06
0.4	0.0007117431023779926	0.0005098684128276882	0.0004409334201482469	3.4180465	500	42.69
0.6	0.00139379347689219	0.0010564461941053916	0.0010846186485675626	−7.2035594	500	83.64

Table 7
Mean square error of parameter p_0.

Parameter p_0	Mean Square Error			Gradients	Epoch	Time (seconds)
	Training	Validation	Testing			
0.1620	0.0013918580323475462	0.0009325762518555004	0.0008366848464818938	6.8000407	500	47.47
0.3620	0.0014259593342586623	0.0012763894096096905	0.0012190671938689238	−2.2520385	500	42.71
0.5620	0.0004853101681840166	0.0002431725558202924	0.0002224312855156916	−0.8025021	500	83.66

Table 8
Mean square error of parameter q_0.

Parameter q_0	Mean Square Error			Gradients	Epoch	Time (seconds)
	Training	Validation	Testing			
0.014	0.001094270929623875	0.0007869453628662185	0.0007210533219478958	1.2430755	500	33.78
0.014	0.0011478442434164592	0.0008774561031368146	0.0010015972335303178	−0.4870227	500	43.04
0.021	0.0004650403991820 5417	0.0001736478194307 6868	0.0001615336734545 7067	−0.2307364	500	44.53

Table 9
S, C, and J_u use different architectures with Tanh activation functions, and compare MSE and CPU time using neural networks against Odeint functions.

Epoch	HL	MSE S	MSE C	MSE J_u
1000	(50, 50, 50, 50, 50, 50)	0.00011834826870576394	6.735903455922328e-05	0.00011347152723109797
1000	(100, 100, 100, 100,100, 100)	0.00010942539496918528	4.3290713612521e-05	0.00016953481055190114
1000	(150, 150, 150, 150, 150, 150)	0.00021249947743635838	6.215586386144093e-05	0.0001256579064716381
1000	(50, 50, 50, 50, 50, 50, 50, 50)	0.00047094430098290163	0.000163683378789676	3.881227428842706e-05
1000	(100, 100, 100, 100,100, 100, 100, 100)	0.0003982900499426048	0.00016783945713286268	6.568833250953327e-05
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.001870307484133571	0.0002107030224946311	0.00030915095312768036

Table 10
The comparison between MSE and CPU time for J_α and H using neural network models versus Odeint function over 144 subdivisions within the observation interval is performed with different architectures with Tanh activation function.

Epoch	Nb. Of nodes in each HL	MSE J_α	MSE H	Time (sec)
1000	(50, 50, 50, 50, 50, 50)	0.00012996505500372538	2.173245143424839e-05	81
1000	(100, 100, 100, 100, 100, 100)	4.375388568377944e-05	5.764949958970097e-05	86
1000	(150, 150, 150, 150, 150, 150)	0.00043087716451023473	0.0001295844769123584	88
1000	(50, 50, 50, 50, 50, 50, 50, 50)	1.4055346378540256e-05	0.0001296466359460163	144
1000	(100, 100, 100, 100, 100, 100,100, 100)	0.00033130574918995785	0.0001349608222382391	147
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.00011183580833351509	0.0002945990683589153	149

Table 11
A comparison of MSE and CPU time of neural network models versus the Odeint function over 144 subdivisions within the observation interval for S, C, and J_u using various architectures with sigmoid activation functions has been conducted using various architectures with sigmoid activation functions.

Epoch	HL	MSE S	MSE C	MSE J_u
1000	(50, 50, 50, 50, 50, 50)	6.33371231905628e-05	3.517249456973445e-05	7.694272849410843e-05
1000	(100, 100, 100, 100,100, 100)	0.00012025056236908722	5.010094426059254e-06	6.396973247192084e-05
1000	(150, 150, 150, 150, 150, 150)	0.0001189382193607173	1.0222564137435596e-05	4.3035362047654745e-05
1000	(50, 50, 50, 50, 50, 50, 50, 50)	9.98327593997788e-05	2.97779733349457 3e-06	2.9324136554424427e-05
1000	(100, 100, 100, 100,100, 100, 100, 100)	0.0023225102208376536	0.00017827369178426207	0.0007391143308123905
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.0008059106873161046	3.615142378772123e-05	0.00021745887734848595

Table 12
Comparing MSE and CPU time using neural network models and Odeint function over 144 subdivisions within the observation interval for J_α and H using various architectures with sigmoid activation functions.

Epoch	Nb. Of nodes in each HL	MSE J_α	MSE H	Time (sec)
1000	(50, 50, 50, 50, 50, 50)	0.0001238505667502692	0.0001041099151634035	87
1000	(100, 100, 100, 100, 100, 100)	6.247398295421604e-05	8.087175245778042e-05	89
1000	(150, 150, 150, 150, 150, 150)	2.0287230275063084e-05	4.510181334712294e-05	83
1000	(50, 50, 50, 50, 50, 50, 50, 50)	4.7783802987130056e-05	3.946055746080464e-05	126
1000	(100, 100, 100, 100, 100, 100,100, 100)	0.000444974061088305	0.0005463260533184991	133
1000	(150, 150, 150, 150, 150, 150, 150, 150)	7.638536340140122e-05	0.0002697115605297377	148

Table 13
Comparison of MSE and CPU time using neural network models and the Odeint function for S, C, and J_u over 1999 subdivisions within the observation period using various architectures with Tanh activation function.

Epoch	HL	MSE S	MSE C	MSE J_u
1000	(50, 50, 50, 50, 50, 50)	1.3747911355204665e-05	9.295437024345911e-06	6.578791796874257e-07
1000	(100, 100, 100, 100,100, 100)	9.638255448543593e-06	8.926690736875838e-06	2.206808050423099e-05
1000	(150, 150, 150, 150, 150, 150)	7.583536127612143e-05	2.9004713332015058e-05	5.084406688046023e-05
1000	(50, 50, 50, 50, 50, 50, 50, 50)	2.6815156897660486e-05	2.489967477298457e-05	5.1623016681491776e-05
1000	(100, 100, 100, 100,100, 100, 100, 100)	1.0282521793289504e-06	2.9103209072277117e-06	1.971509497373305e-05
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.00033457517435414114	0.00023548044618867 8	4.777088649124855e-05

Table 14
Comparing MSE and CPU time using neural network models versus the Odeint function over 1999 subdivisions in the observation interval for J_α and H using various architectures with Tanh activation functions.

Epoch	Nb. Of nodes in each HL	MSE J_α	MSE H	Time
1000	(50, 50, 50, 50, 50, 50)	3.800589780228712e-06	1.860283720019847e-06	8 m 24 s
1000	(100, 100, 100, 100, 100, 100)	5.320197295486083e-06	3.823664942878311e-06	8 m 12 s
1000	(150, 150, 150, 150, 150, 150)	3.5104141731944575e-05	4.882989129869976e-05	9 m 34 s
1000	(50, 50, 50, 50, 50, 50, 50, 50)	0.00010995783012417793	3.75259171659801e-05	9 m 54
1000	(100, 100, 100, 100, 100, 100,100, 100)	4.8140505461706705e-06	6.428443431773065e-06	10 m 47 s
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.00023531921850474528	0.0004034219380610336	10 m 59 s

model, the first step is to discretize model (7). To begin, we set up the model (7) as follows:

$$\left.\begin{aligned} {}^{ABC}\mathcal{D}_{\tau}^{\rho}S(\tau) &= \lambda_1(t,S,C,J_u,J_a,H)\\ {}^{ABC}\mathcal{D}_{\tau}^{\rho}C(\tau) &= \lambda_2(t,S,C,J_u,J_a,H)\\ {}^{ABC}\mathcal{D}_{\tau}^{\rho}J_u(\tau) &= \lambda_3(t,S,C,J_u,J_a,H)\\ {}^{ABC}\mathcal{D}_{\tau}^{\rho}J_a(\tau) &= \lambda_4(t,S,C,J_u,J_a,H)\\ {}^{ABC}\mathcal{D}_{\tau}^{\rho}H(\tau) &= \lambda_5(t,S,C,J_u,J_a,H) \end{aligned}\right\}. \tag{36}$$

We get the following iterative form

learning capability, neural network modeling has emerged as a powerful alternative to traditional simulation techniques for solving differential equations. In our study, the neural network was trained on the reference data generated by the Atangana–Toufik method to approximate the system's dynamic behavior. The resulting computational model is represented in Equation (7). An input to the training dataset is a set of nodes that form a mesh inside the interval that stands for the system's answer. The numerical solution to model (7) is provided in this dataset based on the numerical scheme defined by Eqs. (37)–(41). To improve the neural network's performance, the training data might be normalized to a range of zero to one. The revised formulation should be modified as needed to change the component densities. Our function-based model produces the dependent variable's derivatives. The next step involves

$$S_{\dagger+1} = S + \frac{1-\rho}{\Im(\rho)}\lambda_1(t,S,C,J_u,J_a,H) + \frac{\rho}{\Im(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{aligned} &\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\}\\ &-\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{aligned}\right], \tag{37}$$

$$C_{\dagger+1} = C + \frac{1-\rho}{\Im(\rho)}\lambda_2(t,S,C,J_u,J_a,H) + \frac{\rho}{\Im(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{aligned} &\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\}\\ &-\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{aligned}\right], \tag{38}$$

$$(J_u)_{\dagger+1} = J_u + \frac{1-\rho}{\Im(\rho)}\lambda_3(t,S,C,J_u,J_a,H) + \frac{\rho}{\Im(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{aligned} &\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\}\\ &-\frac{l^{\kappa}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{aligned}\right], \tag{39}$$

$$(J_a)_{\dagger+1} = J_a + \frac{1-\rho}{\Im(\rho)}\lambda_4(t,S,C,J_u,J_a,H) + \frac{\rho}{\Im(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{aligned} &\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\}\\ &-\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{aligned}\right], \tag{40}$$

$$H_{\dagger+1} = H + \frac{1-\rho}{\Im(\rho)}\lambda_5(t,S,C,J_u,J_a,H) + \frac{\rho}{\Im(\rho)}\sum_{\ell=0}^{\dagger}\left[\begin{aligned} &\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\{(\dagger+1-\ell)^{\rho}(\dagger+2-\ell+\rho)-(\dagger-\ell)^{\rho}(\dagger+2-\ell+2\rho)\}\\ &-\frac{l^{\rho}\lambda_1(t,S,C,J_u,J_a,H)}{\Gamma(\rho+2)}\left\{(\dagger+1-\ell)^{\rho+1}-(\dagger-\ell)^{\rho}(\dagger+1-\ell+\rho)\right\}\end{aligned}\right], \tag{41}$$

The results were derived using a numerical technique based on the Atangana–Toufik method, which served as the reference solution for our study. This method was implemented via the Google Colab platform for efficient computation. The model parameters and initial conditions used in the simulations are summarized in Table 3. Given its flexibility and

building an array of the integration intervals (t_initial, t_final, subdivisions) for our model using the *linspace* function in NumPy. To augment our training set, X_train, with additional independent variables, we raise the value of subdivisions. Choosing the correct architecture is critical to the success of neural network simulations of corrupted models

Table 15
Comparison of MSE and CPU time for J_α and H using neural network models versus the Odeint function over 1999 subdivisions within the observation interval using various architectures with sigmoid activation functions.

Epoch	HL	MSE S	MSE C	MSE J_u
1000	(50, 50, 50, 50, 50, 50)	1.898447650035918e-05	5.311400205636435e-06	2.323755414259005e-06
1000	(100, 100, 100, 100,100, 100)	1.912596132533878e-05	2.0023283645725164e-05	2.0539912400809698e-05
1000	(150, 150, 150, 150, 150, 150)	3.187485339995028e-06	1.818946488203046e-05	1.1659273688805027e-05
1000	(50, 50, 50, 50, 50, 50, 50, 50)	2.9122098478272828e-06	1.87703505275504e-06	5.833551203013588e-06
1000	(100, 100, 100, 100,100, 100, 100, 100)	5.331985869427844e-05	2.0785074139480056e-05	3.137831973074243e-05
1000	(150, 150, 150, 150, 150, 150, 150, 150)	1.6311480304156958e-05	4.144056268066531e-06	2.176914738919118e-05

Table 16
Comparing the MSE and CPU time of neural network models with Odeint functions across 1999 subdivisions within the observation interval for J_α and H using various architectures with sigmoid activation functions.

Epoch	Nb. Of nodes in each HL	MSE J_α	MSE H	Time (min)
1000	(50, 50, 50, 50, 50, 50)	7.478707959392191e-06	3.5611877756629283e-06	8 m 2 s
1000	(100, 100, 100, 100, 100, 100)	3.6612966347246304e-05	2.471677146562837e-05	8 m 4 s
1000	(150, 150, 150, 150, 150, 150)	4.152372051525942e-06	6.691317843088231e-06	9 m 1 s
1000	(50, 50, 50, 50, 50, 50, 50, 50)	2.1612268944449704e-05	8.82213601273546e-06	8 m 13 s
1000	(100, 100, 100, 100, 100, 100,100, 100)	4.0056949337182e-05	1.77788189694214e-05	9 m 19 s
1000	(150, 150, 150, 150, 150, 150, 150, 150)	0.00011942757548675624	2.378926280523484e-05	10 m 19 s

because it determines the model's ability to learn from data and make predictions. The architecture of a neural network encompasses all aspects of its organization, including the quantity and type of layers, the number of neurons in each layer, and the connections between them. When building the architecture, it is vital to consider computing resources such as processing power and time and avoid unreasonable demands. As a last need, the model must be flexible and scalable to handle varying amounts of data and evolving disease patterns. The model's ability to faithfully reproduce real-world problem scenarios depends on our ability to preserve this delicate balance. Using a neural network architecture with one input layer, eight hidden layers consisting of 150 nodes each, and a five-node output layer, the study examined the effects of brain activity on the brain. The hyperbolic tangent (tanh) was used as the activation function, also assumed values in Table 3 ware chosen based on model stability, parameter sensitivity, and alignment with prior literature where exact data was unavailable. Additionally, we included a graph to illustrate how these values influence model behavior.

A stochastic gradient descent algorithm called Adaptive Moment Estimation (Adam) is used in our neural network to incorporate stochastic moments. Additionally, the script that constructs the neural network using Keras is utilized. The model's loss decay over 500 training epochs is shown in Fig. 3, plotted on a logarithmic scale. A quick decline in loss early on indicates that the model is picking up new information and getting better. With each passing epoch, the model is approaching its optimal performance as both the rate of advancement and the loss curve level out. As training nears completion, subtle shifts in behavior suggest the model is fine-tuning its parameters. During the training process, the model's predictive power increased steadily as the loss decreased. We split the dataset into three parts: training (64 %), validation (16 %), and testing (20 %). Initially, training and validation took up 80 %, while testing took up 20 %. This allocation was then finalized. After that, the training set was divided again, with 80 % going into training and 20 % into validation. The percentages are calculated in order to thoroughly assess the model's performance on unmeasured data, and to ensure the best validation and training of the model. Our proposed approach to solving the corruption model with a neural network is illustrated by the following comprehensive code sample. This code provides an overview of the neural network model's architecture and then details the main stages for initializing, compiling, and training the model. For detail implementation see Algorithm 1.

In Tables 4-9 the mean square error (MSE) across training, validation, and testing datasets for four parameters ρ, β, π, P_0, and q_0, alongside the computational metrics of gradients, epochs, and processing time. Generally, as the values of ρ, π, P_0, and q_0 increase, the MSE also tends to rise, suggesting a degradation in model performance. For β, the MSE remains relatively stable across its different values, though variations in gradients are observed. The processing time remains fairly consistent across all parameter adjustments, except for the parameter P_0 which shows a significant decrease in time with higher values. These results underscore the delicate balance required in parameter tuning to optimize model performance while managing computational efficiency. The Tables 9-16 offer a comprehensive comparison of mean square error (MSE) and centeral processing unit (CPU) time for various neural network architectures using different activation functions (Tanh and Sigmoid), numbers of epochs, nodes in hidden layers, and subdivisions within the observation interval. Table 9 and Table 13 examine the performance of models with Tanh activation functions across different neural network architectures and subdivision points, showing MSE for parameters (S, C, J_u). Table 10 and Table 14 extend this comparison to parameters (J_α, H). Similarly, Tables 11 and 15 present results using Sigmoid activation functions for (S, C, J_u), while Tables 12 and 16 focus on (J_α, H). These experiments reveal the impact of different configurations on model accuracy and computational efficiency, highlighting variations in MSE and CPU time across the tested scenarios.

Algorithm 1: Numerical Solution and Neural Network Modeling

Input: Time step h, Final time t_{final}, and varying parametersOutput: Time series solutions and neural network predictions for variables (S, C, J_u, J_α, H)
ProcedureDifferential Equations Definition:

- Eq. (34) returns the rate of change for variable (S, C, J_u, J_α, H), based on given parameters and states.

Numerical Solution:

- Initialize arrays (S, C, J_u, J_α, H) with initial conditions.
- For each parameter in parameter values:
- Compute time steps t from 0 to t_{final}, at intervals of h.
- For each time step n, calculate:
- Summations: Use Eqs. (35)–(39) approximations to compute contributions of each differential equation over past time steps.
- Updates: Calculate next values of (S, C, J_u, J_α, H) using initial conditions and summations.

Neural Network Model Setup:

(*continued on next page*)

(*continued*)

Algorithm 1: Numerical Solution and Neural Network Modeling

- normalize_data (X_{train}, Y_{train}): Scales the time and output data to the range [0, 1].
- train_keras_model (X_{train}, Y_{train}): Constructs and trains a neural network model:
- Architecture: Sequential with multiple dense layers using tanh activation and a final dense output layer.
- Training: Uses Adam optimizer and mean squared error loss function over 500 epochs with a validation split of 16 %.

Execution Loop:

- For each parameter in parameter values:
- Execute computeValues to get numerical solutions.
- Train the neural network model on the numerical data.
- Use the trained model to predict future values.
- Store predictions and numerical results for subsequent analysis.

Data Visualization:

- For each variable (S, C, J_u, J_a, H)
- Plot both numerical solutions and neural network predictions.
- Configure plots with appropriate labels, legends, and titles.

End Procedure

Fig. 3 (a-g) Analyzes the effect of contact rate (β) on the dynamics of susceptible and corrupted populations while considering fractional parameter α. Notably, as the contact rate increases, the number of susceptible individuals decreases, indicating a heightened risk of infection. Simultaneously, the count of corrupted individuals rises with greater contact rates, reflecting the increased transmission of corruption. It is evident that when the corrupted individuals meet with the susceptible one probably the susceptible person expires from its lifestyle that the corrupt person has built through corruption. The number of susceptible individuals decreases when the value of alpha increases, whereas the number of corrupted individuals increases when the value of alpha increases. For training, testing and validation for varying β see Table 5.

Fig. 4(a-g) illustrate the influence of the trial (π) on the dynamics of corrupted and jury classes. Notably, the figure reveals that as the π value increases (means the number of trail of corrupted persons), there is a notable reduction in the population of corrupted individuals and the number of trialed individuals increases. Obviously, when the trial of a corrupted individual starts because of the consequences of the trial on the corrupted individual, those who intend to do corruption will restrain themselves by considering its result. The trial (π) plays an important role in reducing corruption in the community. For training, testing and validation for varying π see Table 6.

Fig. 5 (a-g) show the effect of P_o on jury and jail. The number of trialed individuals decreases and the number of punished individuals increases when the value of (result of corrupted trials) increases. When corrupted individuals are found to be gullible and punished by sending them to jail, the number of corrupted individuals under trial in the jury decreases and the number of individuals in jail increases. For training, testing and validation for varying P_0 see Table 7.

Fig. 6 (a-h) show the effect of q_o on jail and honest classes. As it can be seen that when the value of q_o (means an individuals who have completed his punishment and have become honest individuals) increase the number of individuals in jailed class decreases and the number of honest individuals increases. Due to the fact that the guilty person completed his punishment and was released from prison, it is likely that he left the corrupt system and served the community as an honest citizen. The fractional parameter ρ introduces a memory effect into the system, where higher values of alpha emphasize the influence of past corruption and interactions on the current state of the population, amplifying the significance of historical data in shaping these dynamics. For training, testing and validation for varying q_0 see Table 8.

5. Optimal control model of corruption

This section addresses both minimizing corruption's spread and minimizing its cost. The optimal control model (40) reduces corruption numbers by revisiting the corrupting model (7). The spread of corruption is reduced by using two control variables, u_1 and u_2. There is a strong and efficient anti-corruption policy in control u_1; efforts are being made using social media, media, and civil society (religious organizations). Corrupt people need to be punished, so control u_2 represents that effort. This nonlinear system of ordinary differential equations describes the model with control.

$$\left.\begin{aligned}
\frac{dS}{dt} &= \Lambda - (1-u_1)\beta CS + m_2 J_u + \tau J_a - \mu S \\
\frac{dC}{dt} &= (1-u_1)\beta CS - \pi C - \alpha C - u_2 C - \mu C \\
\frac{dJ_u}{dt} &= \pi C + \eta H - (p_o + m_2 + \mu) J_u + u_2 C \\
\frac{dJ_a}{dt} &= p_o J_u - (q_o + \tau + \mu) J_a \\
\frac{dH}{dt} &= q_o J_a + \alpha C - (\eta + \mu) H
\end{aligned}\right\}. \tag{42}$$

Taking into account initial condition (7)

The most effective intervention strategy is determined by an integrated model to mitigate disease transmission. Spread-controlling and spread-implementation costs must both be assessed. In order to minimize the control variables u_1 and u_2, a differential equation (40) is used:. A differential equation (42) is used to minimize the control variables u_1 and u_2, where the objective function is given as follows:

$$J(u_1, u_2) = \int_0^{t_f} \left[G_1 C + \frac{1}{2}\left(G_2 u_1^2 + G_3 u_2^2\right) \right] dt \tag{43}$$

In accordance with (31). Weight constants G_1, G_2 and G_3 represent an infected population in the objective functional. A cost associated with relvant intervention is represented by the term $\frac{G_1 u_1^2}{2}, \frac{G_2 u_2^2}{2}$ and $\frac{G_3 u_3^2}{2}$. In addition, the proportionality of the control function to the square of the costs is assumed. We must determine the optimal control functions u_1^* and u_2^* in order to solve the optimal control problem such that

$$J(u_1^*, u_2^*) = \min\{J(u_1, u_2), (u_1, u_2) \in U\}, \tag{44}$$

The control set is defined by the system (30) as follows:

$$U = \{(u_1, u_2) \,|u_i(t) \text{ is Lebesgue measure on } [0,1], i = 1, 2, 3\}. \tag{45}$$

A maximum principle of Pontryagin allows the objective function and state variables u in relation to a Hamiltonian H to be minimized or maximized using Pontryagin's maximum principle. A Pontryagin maximum principle is used to derive this optimal control problem. In order to determine the Hamiltonian and Lagrangian of optimal control problems (42) to (44), we must first define the Lagrangian. The lagrangian for optimal problems is expressed as follows:

$$L = G_1 C + \frac{1}{2}\left(G_2 u_1^2 + G_3 u_2^2\right) \tag{46}$$

In the control problem, the Hamiltonian is given by a minimal Lagrangian value:

$$H = L(C, u_1, u_2) + \lambda_1 \frac{dS}{dt} + \lambda_2 \frac{dC}{dt} + \lambda_3 \frac{dJ_u}{dt} + \lambda_4 \frac{dJ_a}{dt} + \lambda_5 \frac{dH}{dt} \tag{47}$$

where λ_i, $i = 1, 2, \ldots 5$ are the adjoint variables. In order to derive the optimality system for system (40), we need to apply Lukes' (Lukes, 1982) results and prove the existence of an optimal control scheme. In fact, for the existence of this optimal control, we use the results in Lukes (Lukes, 1982). State variables and control variables have nonnegative values. This minimizing problem allows the objective functional L to satisfy the necessary convexity. The set of all the control variables $(u_1, u_2) \in U$ is also convex and closed by definition. Moreover, the integrand

in equation (41), $G_1C + \frac{1}{2}\left(G_2u_1^2 + G_3u_2^2\right)$, is convex in the control set U because the optimal system is bounded; therefore, an optimal control exist. Also, we can easily see that there exists a constant $\rho > 0$ and positive numbers Ξ_1, Ξ_2 such that

$$J(u_1, u_2) \geq \Xi_1\left(|u_1|^2 + |u_2|^2\right)^{\frac{l}{2}} - \Xi_2, \tag{48}$$

As the state variables are bounded, we can conclude that an optimal control exists.

Theorem 5.1. (*There exists an optimal control* $u^* = \left(u_1^*, u_2^*\right) \in U$ *such that*)

$$J\left(u_1^*, u_2^*\right) = \min_{(u_1,u_2)\in U} J(u_1, u_2) \tag{49}$$

Depending on initial conditions.

The Hamiltonian (47) is subject to Pontryagin's maximum principle (Pontryagin, xxxx), which implies that there is a nontrivial vector function $\lambda = (\lambda_1, \lambda_2, \ldots\ldots, \lambda_n)$ that meets the following conditions if (x, u) is an optimal response to the optimal control problem:

$$x' = \frac{\partial H(t,x,u,\lambda)}{\partial \lambda},\ 0 = \frac{\partial H(t,x,u,\lambda)}{\partial u},\ \lambda' = \frac{\partial H(t,x,u,\lambda)}{\partial x} \Bigg\} \tag{50}$$

The Hamiltonian H is then treated as necessary in (47)

Theorem 5.2. (*Let* S^*, C^*, J_u^*, J_a^* *and* H^* *be optimal state solutions with associated optimal control variables* u_1^*, u_2^* *for the optimal control problems (30) and (31). Then there exist adjoint variables* λ_i *for* $i = 1, 2, 3....5$ *satisfying*)

$$\left.\begin{aligned} \lambda_1 &= -\beta SC - \Lambda - \mu S + \tau J_a + J_u m_2 + \beta SCu_1, \\ \lambda_2 &= -C(\pi + \alpha) + \beta SC - \mu C - \beta SCu_1 - u_2, \\ \lambda_3 &= C\pi + \eta H - J_u(\mu + m_2 + p_o) + Cu_2, \\ \lambda_4 &= J_u p_o - J_a(\mu + \tau + q_o), \\ \lambda_5 &= \alpha C - H(\eta + \mu) + J_a q_o. \end{aligned}\right\} \tag{51}$$

A boundary condition or a transversality condition

$$\lambda_i(T) = 0,\ \ i = 1, 2, 3....5$$

Furthermore, the control functions u_1^* and u_2^* are given by

$$\begin{aligned} u_1^* &= max\left\{\min\left(\frac{-\beta SC\lambda_1 + \beta SC\lambda_2}{G_2}, 1\right), 0\right\},\ \ u_2^* \\ &= max\left\{\min\left(\frac{\lambda_2 - C\lambda_3}{G_3}, 1\right), 0\right\}\Bigg\} \end{aligned} \tag{52}$$

Hence, the optimal control pair is characterized.

5.1. Numerical Discussion of optimal control

We used parameters given in Table 1 and chose weight constants $A_1 = 0.06,\ A_2 = 0.8,\ G_1 = 100,\ G_2 = 30,\ G_3 = 50$ in our numerical simulations in order to balance the population and control functions in the objective functional. It shows in the Fig. 7 that the infected individuals without any control functions in Cyan, while the magenta curve shows the infected individuals with optimal control functions u_1^* and u_2^*. In conclusion, the outbreak has been significantly reduced with optimal control strategies, and consequently, the number of corrupted individuals has been significantly reduced as well.

6. Conclusion

In this study, we developed a corruption dynamics model incorporating both deterministic and fractional approaches. The model successfully identified the deterministic reproduction number as a critical threshold for corruption spread. It also demonstrated that if this threshold is controlled effectively (i.e.$R_o < 1$), corruption can be eliminated. Conversely, $R_o > 1$ indicates persistence of corruption at a steady level. The analysis revealed that strong anti-corruption measures, supported by media and civil society, significantly influence this threshold and can reduce corruption levels.

The fractional model, solved using the Atangana–Toufik method, offered deeper insight into the memory effects and complex behavior of corruption systems. Furthermore, the integration of neural networks presented a novel methodological advancement, enabling more flexible and accurate predictions beyond standard numerical methods. Unlike mesh-based approaches, neural networks can estimate outcomes at arbitrary points, enhancing model applicability and computational efficiency. This work is novel in combining fractional modeling with neural networks to simulate and predict corruption dynamics more effectively.

As part of our future research, we plan to extend the current model by incorporating time-variable parameters, allowing for more realistic representation of evolving corruption dynamics. We also aim to apply physics-informed neural networks (PINNs) for parameter estimation and system identification from synthetic or real data. In parallel, bio-inspired optimization algorithms (e.g., genetic algorithms, particle swarm optimization) will be explored to improve solution convergence and model training efficiency. Furthermore, we intend to investigate the inverse problem for key parameter estimation and assess estimation uncertainty using Monte Carlo simulations and comparisons with theoretical bounds such as the Cramér–Rao lower bound (CRLB). While the current work provides a strong theoretical foundation, future efforts will focus on validating the model with real-world corruption datasets and developing policy-driven control strategies to assess its applicability in practical governance scenarios.

Author contributions

Each author played an equal role in finalizing and approving the manuscript's final draft.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Aamir Farooq reports administrative support, statistical analysis, and travel were provided by Zhejiang Normal University. Aamir Farooq reports a relationship with Zhejiang Normal University that includes: non-financial support and travel reimbursement. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Aamir Farooq acknowledges the postdoctoral fellowship supported by Zhejiang Normal University, China, under Grant No. YS304024913. Kamil Shah, gratefully acknowledges the financial support provided by the Yunnan Provincial Joint Special Project for Basic Research of Local Undergraduate Colleges and Universities under Grant No. 202401BA070001-132. Wen Xiu Ma acknowledges support from the Ministry of Science and Technology of China (G2021016032L and G2023016011L), the National Natural Science Foundation of China (NSFC) under grants 12271488 and 11975145.

Appendix A

Data availability

No data was used for the research described in the article.

References

Abdulrahman, S. (2014). Stability Analysis of the Transmission Dynamics and Control of Corruption. *Pacific J. Sci. Technol., 15*, 99–113.

Abdulwasaa, M. A., Kawale, S. V., Abdo, M. S., Albalwi, M. D., Shah, K., Abdalla, B., & Abdeljawad, T. (2024). Statistical and Computational Analysis for Corruption and Poverty Model Using Caputo-Type Fractional Differential Equations. *Heliyon, 10*. https://doi.org/10.1016/j.heliyon.2024.e25440

Adamu G., Mohammed J., Ahmed K. D., Kevin J. A.; Deterministic Model of Corruption Dynamics in Nigeria VIA Homotopy Perturbation Method. ANSNE_Volume 8_Issue 1_ Pages 35-52.

Adedeji, B. P. (2023). Electric Vehicles Survey and a Multifunctional Artificial Neural Network for Predicting Energy Consumption in All-Electric Vehicles. *Results in Engineering, 19*. https://doi.org/10.1016/j.rineng.2023.101283

Akakuru, O. C., Adakwa, C. B., Ikoro, D. O., Eyankware, M. O., Opara, A. I., Njoku, A. O., Iheme, K. O., & Usman, A. (2023). Application of Artificial Neural Network and Multi-Linear Regression Techniques in Groundwater Quality and Health Risk Assessment around Egbema, Southeastern Nigeria. *Environment and Earth Science, 82*. https://doi.org/10.1007/s12665-023-10753-1

Akgül, A., Farman, M., Sutan, M., Ahmad, A., Ahmad, S., Munir, A., & Hassani, M. K. (2024). Computational Analysis of Corruption Dynamics Insight into Fractional Structures. *Applied Mathematics in Science and Engineering, 32*. https://doi.org/10.1080/27690911.2024.2303437

Alemneh, H. T. (2020). Mathematical Modeling, Analysis, and Optimal Control of Corruption Dynamics. *Journal of Applied Mathematics, 2020*. https://doi.org/10.1155/2020/5109841

R. Ali, S. Gulshan, K. T. Kubra; Modeling and Analysis of Corruption Dynamics in Society under Fractal-Fractional Derivative in Caputo Sense with Power-Law. Ptolemy Scientific Reasearch Press. https://pisrt.org/.

Aliyu, S. U. R., & Elijah, A. O. (2011). How Corruption Affects Economic Growth in Nigeria. *J. Glob. Bus. Adv., 4*, 143–154. https://doi.org/10.1504/JGBA.2011.041498

Anjam, Y. N., Aslam, M. I., Cheema, S. A., Munawar, S., Saleem, N., & Rahman, M. U. (2024). Stability Analysis of the Corruption Dynamics under Fractional-Order Interventions. *Nonlinear Eng., 13*. https://doi.org/10.1515/nleng-2022-0363

Athithan, S., Ghosh, M., & Li, X. Z. (2018). Mathematical Modeling and Optimal Control of Corruption Dynamics. *Asian-European J. Math., 11*. https://doi.org/10.1142/S1793557118500900

Awadalla, M., Rahman, M., & ur; Al-Duais, F.S., Al-Bossly, A., Abuasbeh, K., Arab, M.. (2023). Exploring the Role of Fractal-Fractional Operators in Mathematical Modelling of Corruption. *Applied Mathematics in Science and Engineering, 31*. https://doi.org/10.1080/27690911.2023.2233678

V. Bhargava;World bank global issues seminar series; 2005, undefined The 'Cancer of Corruption.' Between Moral. Law 2020, 143–162, doi:10.4324/9781315261683-14.

M.A. El Yamani, J. El Karkri, S. Lazaar, R. Aboulaich; A two-group epidemiological model: Stability analysis and numerical simulation using neural networks, 2023, International Journal of Modeling, Simulation, and Scientific Computing, https://www.worldscientific.com/doi/abs/10.1142/S1793962323500290.

Hathroubi, S., & Trabelsi, H. (2014). Epidemic Corruption:A Bio-Economic Homology. *European Scientific Journal, 10*, No.10.

Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of Machine Learning-Based K-Means Clustering for Financial Fraud Detection. *Acad. J. Sci. Technol., 10*, 33–39. https://doi.org/10.54097/74414c90

A; Husnain, H. K. Hussain, H. M. Shahroz, M. Ali, A. Gill, S. Rasool; Exploring AI and Machine Learning Applications in Tackling COVID-19 Challenges. Revista Española de Documentación Científica eISSN: 1988-4621 pISSN: 0210-0614.

Kanbur, R. The Role of the World Bank in Middle-Income Countries. 2018, 167–180, doi: 10.1007/978-981-10-7950-4_8.

Lemecha, L., & Feyissa, S. (2018). Modelling Corruption Dynamics and Its Analysis. *Ethiop. J. Sci. Sustain. Dev., 5*, 13–27. https://doi.org/10.20372/EJSSDASTU:V5.I2.2018.34

López Iturriaga, F. J., Pastor Sanz, I., & Blanco Alcántara, D. (2022). A Neural Network Approach for Predicting Corruption in Public Procurement. *European Journal of International Management, 1*. https://doi.org/10.1504/ejim.2022.10057105

Lukes, D.L. Differential Equations : Classical to Controlled. 1982, 337.

Meskher, H., Belhaouari, S. B., Deshmukh, K., Hussain, C. M., & Sharifianjazi, F. (2023). A Magnetite Composite of Molecularly Imprinted Polymer and Reduced Graphene Oxide for Sensitive and Selective Electrochemical Detection of Catechol in Water and Milk Samples: An Artificial Neural Network (ANN) Application. *Journal of the Electrochemical Society, 170*, Article 047502. https://doi.org/10.1149/1945-7111/acc97c

Nathan, O. M., & Jackob, K. O. (2019). Stability Analysis in a Mathematical Model of Corruption in Kenya. *Asian Res. J. Math., 1–15*. https://doi.org/10.9734/ARJOM/2019/V15I430164

J. Panovska-Griffiths, C.C. Kerr, W. Waites, R.M. Stuart; Chapter 10 - Mathematical modeling as a tool for policy decision making: Applications to the COVID-19 pandemic, Handbook of Statistics, Elsevier, 44, 2021,P 291-326, https://doi.org/10.1016/bs.host.2020.12.001.

Pontryagin, L.S. Mathematical Theory of Optimal Processes; ISBN 9782881240775.

Teklu, S. W. (2024). Insight into the Optimal Control Strategies on Corruption Dynamics Using Fractional Order Derivatives. *Sci. African, 23*. https://doi.org/10.1016/j.sciaf.2024.e02069

M. Wanless; The World Bank's Fight against Corruption: "See Nothing, Hear Nothing, Say Nothing." Hydra.