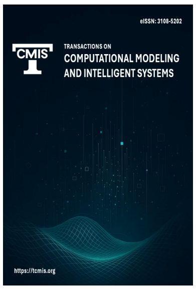

Transactions on
Computational Modeling
and Intelligent Systems

Volume 4, July 2026, Article 10023
https://doi.org/10.65112.tcmis.10023

eISSN: 3108-5202
www.tcmis.org

RESEARCH ARTICLE

Application of Upadhyaya transforms with machine learning for predictive and analytical solutions in complex systems

Yamini Parthiban[1,3], Prabakaran Raghavendran[2,3,*], Dinesh Thakur[4] and Srinivasan Madhumitha[5]

[1] *Department of Mathematics, SRM Institute of Science and Technology, Ramapuram, Chennai-600089, Tamil Nadu, India.*
[2] *Department of Mathematics, Easwari Engineering College, 18 Bharathi Salai, Ramapuram Chennai-600089, Tamil Nadu, India.*
[3] *Department of Mathematics and Science Education, Faculty of Education, Harran University, Sanliurfa 63150, Turkey.*
[4] *Department of Mathematics, Bahra University, Waknaghat, District Solan, Himachal Pradesh, India.*
[5] *Department of Mathematics, Sri Muthukumaran Institute of Technology, Chennai, Tamil Nadu, India.*
[*] *Corresponding Author. Email: rockypraba55@gmail.com (P. Raghavendran)*

Article Information

Received: 6 December 2025
Accepted: 2 March 2026
Published 11 July 2026

AMS Classification:
44A15, 45D05

Abstract

The study uses the Upadhyaya transform to find exact solutions of three physical models which include the Atwood machine the Beer–Lambert law and electron motion in electromagnetic fields. The proposed framework demonstrates how exact analytical solutions derived via the Upadhyaya transform can be utilized as physically consistent priors, constraints, and validation references within learning-based models, thereby enhancing interpretability, numerical stability, and reproducibility while reducing dependence on large empirical datasets. The research presents an application-oriented method which connects integral transform theory with contemporary machine learning techniques to simulate and study complex physical and chemical systems.

Keywords: Upadhyaya transforms, inverse Upadhyaya transform, machine learning, chemical sciences, physical sciences

1. Introduction

Scientists use Integral Transforms together with Machine Learning as their fundamental method to solve problems in Chemical and Physical Sciences which allows them to make substantial progress in their scientific work through their ability to analyze data and create models and make predictions. The mathematical field of integral transforms which includes Fourier transforms and Laplace transforms and Wavelet transforms enables scientists to study complex signals and solve differential equations and discover important characteristics in high-dimensional datasets. Scientists use these tools to study chemical and physical phenomena which operate through continuous dynamic systems that scientists represent with time and frequency domain data and spectroscopic signals and quantum mechanical wave functions and molecular dynamics. Machine learning can recognize patterns and make predictions and create model optimizations when handling large datasets which contain noise and incomplete information. The research team uses integral transforms together with ML to convert scientific data into new formats which improve ML model performance and result in more precise predictions and better data compression and understanding of physical and chemical processes.

Mousa [1] used the Upadhyaya Transform to find solutions for first-kind Volterra integral equations. Patil [2] studied integral transforms which include Laplace and Shehu transforms to solve Chemical Sciences problems. Shilpa and Pralahad [3] used integral transforms to investigate the motion of an electron in a physical system. Dinesh and Emad [4] used the Upadhyaya Transform to create precise solutions for cardiovascular models. Dinesh et al. [5] used the Upadhyaya Transform to solve chemical reaction models because it helped them to understand complex differential equations. Prabakaran et al. [6] created a hybrid system that uses both differential equations and machine learning to forecast chemical process behaviour. Prabakaran et al. [7] developed a method to solve fractional differential equations by combining the Ramadan Group Transform with machine learning methods. Jafari et al. [8] used the Anuj transform to solve Abel's integral equation in classical mechanics.

Recent studies show that exact solutions and traveling wave solutions serve as essential tools for studying how nonlinear differential equations behave in both physical and engineering applications. Mahmud [12] conducted research on major traveling wave solutions of the generalized Hietarinta-type equation whereas he showed that analytical methods enable precise identification of complicated nonlinear wave patterns. The study demonstrates that closed-form solutions enable the discovery of fundamental properties which include stability testing and wave propagation analysis that numerical methods fail to provide. Mahmud, Tanriverdi, and Muhamad [13] derived exact traveling wave solutions for the $(2+1)$-dimensional Konopelchenko–Dubrovsky equation using hyperbolic trigonometric function methods. This study demonstrates how symbolic and analytical solution methods create effective solutions for nonlinear systems that operate in higher dimensions which are commonly found in fluid dynamics and plasma physics and their associated disciplines. The research results show that analytical methods which use transform-based techniques to solve complex differential equations remain important today. The researchers of the study found inspiration from contemporary trends to create a new study that combines classical analytical solution methods with the Upadhyaya transform and modern computational and learning-based methods. The proposed framework extends traditional approaches which seek to find precise solutions by integrating analytically obtained information into their algorithmic processes which unifies scientific mathematical modeling with machine-based data intelligence methods.

The study presents original research through the creation of a novel interdisciplinary computational paradigm which combines Upadhyaya transform processing with contemporary machine learning techniques to study and forecast and enhance intricate physical and chemical systems. The authors present the first research which shows how Upadhaya-transformed analytical solutions can be integrated into physics-informed neural networks and synthetic data generation pipelines and hybrid learning systems. Researchers apply integral transforms to classical studies which include recent research on the Upadhyaya transform for solving differential and integral equations through closed-form solutions which stop at the analytical phase of research. The present work establishes the Upadhyaya transform as a computational link which connects mathematical physics with data-driven intelligence systems. The transformed solutions which researchers developed for mechanical systems (Atwood machine) and optical systems (Lambert–Beer law) and electromagnetic systems (electron motion) serve as analytical results which researchers use to build structured priors and physics-based constraints and informative feature representations in machine learning workflows.

The key contributions of this manuscript can be summarized as follows. The research establishes a complete transform-machine learning system which uses Upadhyaya-transformed solutions as ground-truth references to validate physics-based learning systems while decreasing the need for extensive empirical data. The research demonstrates Upadhyaya-based analytical formulas to create synthetic data which provides well-defined training material for supervised and hybrid learning systems used in spectroscopy and imaging and particle dynamics. The research shows that transform-domain parameters can be understood as physical features which scientists can use to solve regression and forecasting and anomaly detection problems. The work provides its main value through new methods which it develops for actual implementation. The manuscript connects integral transform theory with modern machine learning methods to show how traditional analytical methods can be used to improve the accuracy and understanding and dependability of intelligent systems that model complex systems. The interdisciplinary approach enables researchers to use Upadhyaya transform methods in different fields while creating new research paths which connect applied mathematics with physics and chemistry and artificial intelligence.

2. Preliminaries

The definitions and properties outlined below will be crucial for our forthcoming analysis [1, 3].

2.1. Definition (Upadhyaya Transform)

Upadhyaya transform for the function $F(t)$ is mathematically defined as:

$$U\{F(t)\} = \lambda_1 \int_0^{\infty} \exp\left(-\lambda_2 t\right) F\left(\lambda_3 t\right) dt = u\left(\lambda_1, \lambda_2, \lambda_3\right), \left(\lambda_1, \lambda_2, \lambda_3\right) > 0, t \geq 0. \tag{1}$$

The representation of the inverse Upadhyaya transform is as follows:

$$F(t) = U^{-1}\left[u\left(\lambda_1, \lambda_2, \lambda_3\right)\right], \quad t \geq 0. \tag{2}$$

The definition states that $u(\lambda_1, \lambda_2, \lambda_3)$ represents the value of Upadhyaya Transform which operates on the function $F(t)$ with the specified parameters. The U-domain transformation produces a new form of $F(t)$ which shows how the three scaling parameters interact with each other. Using this notation scientists can easily handle the transform through their research process while they build machine learning systems. The parameters λ_1, λ_2, and λ_3 exist as positive real constants which govern the process of amplitude scaling and exponential decay and argument transformation. The parameters enable different function representations while they enable the transform to adapt for analytical modeling and computational integration. The Upadhyaya Transform shows better flexibility through its scaling parameters when compared to traditional integral transforms which include the Laplace transform. The system shows nonlinear behavior through its dual-parametric structure which enables its linkage to data-driven models while operating within complex pharmacokinetic systems through nonlinear and multi-compartment approaches. In situations where analytical solutions need to work together with machine learning methods this type of flexibility becomes extremely valuable.

2.2. Linearity property

Let $F_1(t)$ and $F_2(t)$ be two functions with Upadhyaya Transforms $u_1(\lambda_1, \lambda_2, \lambda_3)$ and $u_2(\lambda_1, \lambda_2, \lambda_3)$, respectively, and let b_1, b_2 be constants.
Then the linearity property of the Upadhyaya Transform is:

$$\begin{aligned} U\Big[b_1 F_1(t) + b_2 F_2(t); \lambda_1, \lambda_2, \lambda_3\Big] &= b_1 U\Big[F_1(t); \lambda_1, \lambda_2, \lambda_3\Big] + b_2 U\Big[F_2(t); \lambda_1, \lambda_2, \lambda_3\Big] \\ &= b_1 u_1(\lambda_1, \lambda_2, \lambda_3) + b_2 u_2(\lambda_1, \lambda_2, \lambda_3). \end{aligned}$$

2.3. Convolution property

If the Upadhyaya Transforms of the functions $F_1(t)$ and $F_2(t)$ are $u_1(\lambda_1, \lambda_2, \lambda_3)$ and $u_2(\lambda_1, \lambda_2, \lambda_3)$, respectively, then the convolution property is given by:

$$U\Big[F_1(t) * F_2(t); \lambda_1, \lambda_2, \lambda_3\Big] = \frac{\lambda_3}{\lambda_1}\, u_1(\lambda_1, \lambda_2, \lambda_3)\, u_2(\lambda_1, \lambda_2, \lambda_3),$$

where the convolution $F_1(t) * F_2(t)$ is defined as:

$$(F_1 * F_2)(t) = \int_0^t F_1(t-x)F_2(x)\,dx = \int_0^t F_1(x)F_2(t-x)\,dx.$$

2.4. Transforms of some elementary functions

The Upadhyaya Transforms of basic functions are given by:

$$\begin{aligned}
&U[1] = \frac{\lambda_1}{\lambda_2}, \quad U[e^{at}] = \frac{\lambda_1}{\lambda_2 - a\lambda_3}, \quad U[t^m] = \frac{m!\,\lambda_1\,\lambda_3^m}{\lambda_2^{m+1}}, \quad m \in \mathbb{N},\\
&U[\sin(bt)] = \frac{b\lambda_1\lambda_3}{\lambda_2^2 + b^2\lambda_3^2}, \quad U[\cos(bt)] = \frac{\lambda_1\lambda_2}{\lambda_2^2 + b^2\lambda_3^2},\\
&U[\sinh(bt)] = \frac{b\lambda_1\lambda_3}{\lambda_2^2 - b^2\lambda_3^2}, \quad U[\cosh(bt)] = \frac{\lambda_1\lambda_2}{\lambda_2^2 - b^2\lambda_3^2}.
\end{aligned}$$

2.5. Upadhyaya transform of derivatives

If $U[F(t)] = u\left(\lambda_1, \lambda_2, \lambda_3\right)$ then from

$$U\left[F'(t); \lambda_1, \lambda_2, \lambda_3\right] = \left\{\frac{\lambda_2}{\lambda_3}\right\} U\left[F(t); \lambda_1, \lambda_2, \lambda_3\right] - \frac{\lambda_1}{\lambda_3}F(0),$$

$$\begin{aligned}
U\left[F''(t); \lambda_1, \lambda_2, \lambda_3\right] &= \left\{\frac{\lambda_2}{\lambda_3}\right\}^2 U\left[F(t); \lambda_1, \lambda_2, \lambda_3\right] - \frac{\lambda_1\lambda_2}{\lambda_3^2}F(0) - \frac{\lambda_1}{\lambda_3}F'(0),\\
U\left[F^n(t); \lambda_1, \lambda_2, \lambda_3\right] &= \left\{\frac{\lambda_2}{\lambda_3}\right\}\left[F(t); \lambda_1, \lambda_2, \lambda_3\right] - \frac{\lambda_1\lambda_2^{n-1}}{\lambda_3^n}F(0) - \frac{\lambda_1\lambda_2^{n-2}}{\lambda_3^{n-1}}F'(0)\\
&- \frac{\lambda_1\lambda_2^{n-3}}{\lambda_3^{n-2}}F''(0) - \ldots. - \frac{\lambda_1}{\lambda_3}F^{n-1}(0).
\end{aligned}$$

2.6. Comparative discussion with classical integral transforms

Integral transforms such as the Laplace, Fourier, Shehu, Sumudu, and Aboodh transforms have been extensively employed to solve differential equations which emerge in physical science and engineering science fields [2,6,7]. The three transforms provide different benefits because of their unique problem-solving capabilities which depend on the problem type and solution requirements and measurement domains. The authors of this subsection conduct a comparative analysis to explain why they choose to use the Upadhyaya transform in their research work. The Laplace transform enables efficient analysis of linear time-invariant systems together with initial value problems which operate on semi-infinite boundary conditions. The system's single-parameter design restricts users who need to work with rescaled inputs or who want to use complete solutions in flexible learning models. The Fourier transform provides optimal performance for frequency-based analysis and signal processing tasks yet its application becomes impractical when dealing with initial conditions in time-based learning systems which need solutions across infinite domains or periodic cycles.

The Sumudu and Aboodh transforms were developed to solve two major problems which occur with the Laplace transform because it fails to keep the original function dimensionality intact and makes inverse calculations more difficult. The transforms work effectively for special types of problems but they depend on fixed kernel designs which restrict their ability to adapt to different parameter settings. Their hybrid analytical–computational workflows use fixed transform methods because they need more adaptable systems for their work. The Shehu transform extends the Laplace transform through new scaling methods which help apply the transform to additional types of differential equations. After selecting parameters for its system design users must accept fixed settings which prevents them from using analytical solutions as machine learning model elements.

The Upadhyaya transform uses its multiple parameter system which lets users separately control three elements. The system provides operational flexibility which lets multiple transform types execute their functions under one operational system. The Upadhyaya transform derivative properties show initial conditions through their design which benefits physics-informed learning models that require both initial and boundary conditions for their main functions. The Upadhyaya transform enables researchers to create reusable closed-form solutions which they can implement in their computational systems. The presence of tunable parameters allows analytical expressions to be adapted for synthetic data generation, feature engineering, and constraint enforcement in learning-based models. The Upadhyaya transform demonstrates its suitability for hybrid analytical–machine learning integration through its implementation in this research study. The Upadhyaya transform provides scientists with better flexibility and parameterization capabilities than classical transforms which include Laplace, Fourier, Shehu, Sumudu, and Aboodh. The Upadhyaya transform serves as a flexible tool which researchers can integrate into their hybrid analytical systems for studying complex systems through learning-based modeling.

3. Atwood machine

In 1784, George Atwood, a tutor at Trinity College, Cambridge, introduced a classic demonstration for determining the acceleration due to gravity g, which remains widely used today. The traditional Newtonian approach to this problem involves applying $F = ma$ to the two masses and eliminating the tension T. To simplify the analysis, the rotational inertia of the pulley is neglected. Alternatively, the Lagrangian formulation [9,10] involves constructing the Lagrangian function and deriving the corresponding equation of motion.

$$\begin{aligned} U &= -m_1 g x - m_2 g(l-x) \\ T = \tfrac{1}{2}\left(m_1+m_2\right)\dot{x}^2, L &= \tfrac{1}{2}\left(m_1+m_2\right)\dot{x}^2 + m_1 g x + m_2 g(l-x) \\ \tfrac{dL}{dx} &= \left(m_1-m_2\right) g, \tfrac{dL}{d\theta} = \left(m_1+m_2\right)\dot{x}. \end{aligned}$$

Using Lagrange equation, we have

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0 \\ \Rightarrow \ddot{x} = \left(\frac{m_1-m_2}{m_1+m_2}\right) g. \end{aligned}$$

Applying the Upadhyaya transform to both sides:

$$\begin{aligned} \left(\frac{\lambda_2}{\lambda_3}\right)^2 u\left(\lambda_1,\lambda_2,\lambda_3\right) - \frac{\lambda_1\lambda_2}{\lambda_3^2}\dot{x}(0) - \frac{\lambda_1}{\lambda_3}x(0) &= \left(\frac{m_1-m_2}{m_1+m_2}\right) gU[1] \\ \left(\frac{\lambda_2}{\lambda_3}\right)^2 u\left(\lambda_1,\lambda_2,\lambda_3\right) - \frac{\lambda_1\lambda_2}{\lambda_3^2}x(0) - \frac{\lambda_1}{\lambda_3}\dot{x}(0) &= \left(\frac{m_1-m_2}{m_1+m_2}\right) g\frac{\lambda_1}{\lambda_2} \\ u\left(\lambda_1,\lambda_2,\lambda_3\right) = \left(\frac{m_1-m_2}{m_1+m_2}\right) g\frac{\lambda_1\lambda_3^2}{\lambda_2^3} &+ \frac{\lambda_1}{\lambda_2}x(0) + \frac{\lambda_1\lambda_3}{\lambda_2^2}x(0). \end{aligned}$$

Applying the Inverse Upadhyaya transform to both sides, we get

$$x(t) = \frac{1}{2}\left(\frac{m_1-m_2}{m_1+m_2}\right) gt^2 + x(0) + t\dot{x}(0).$$

The Atwood machine motion equation derivation through classical Newtonian and Lagrangian mechanics leads to its Upadhyaya transform which enables direct application in physics-informed neural networks and hybrid learning systems. The analytical expression which results from this process functions as a basic truth constraint that maintains physical accuracy during learning in environments with limited data and background noise.

3.1. Lambert-Beer law

When a beam of monochromatic light passes through a homogeneous absorbing medium, the rate of decrease in light intensity with respect to the thickness of the medium is directly proportional to the intensity of the light [2, 11].

$$\frac{dI}{dx} = -\mu I(x).$$

Where:

- I = intensity of light at depth x,
- μ = linear absorption coefficient (a constant for the medium),
- x = distance (thickness) the light has traveled in the medium.

Applying the Upadhyaya transform to both sides:

$$\boldsymbol{U}\left[\frac{\boldsymbol{dI}}{\boldsymbol{dx}}\right] = -\boldsymbol{\mu U}[\boldsymbol{I}(\boldsymbol{x})]$$

$$\frac{\lambda_1}{\lambda_3}u(\lambda_1, \lambda_2, \lambda_3) - \frac{\lambda_1}{\lambda_3}I(0) = -\mu u(\lambda_1, \lambda_2, \lambda_3)$$

$$\frac{\lambda_1}{\lambda_3}u(\lambda_1, \lambda_2, \lambda_3) + \mu u(\lambda_1, \lambda_2, \lambda_3) = \frac{\lambda_1}{\lambda_3}I(0)$$

$$\left[\frac{\lambda_1}{\lambda_3} + \mu\right]u(\lambda_1, \lambda_2, \lambda_3) = \frac{\lambda_1}{\lambda_3}I(0)$$

$$u(\lambda_1, \lambda_2, \lambda_3) = \frac{\lambda_1}{\lambda_1 + \mu\lambda_3}I(0).$$

Applying the Inverse Upadhyaya transform to both sides, we get

$$I(x) = I(0)e^{-\mu x}.$$

The Beer-Lambert law can be solved through both the analytical method of separation of variables and its transform-based representation which serves as a better tool for creating synthetic data and developing machine learning features. The Upadhyaya-transformed solution enables researchers to precisely control their simulation of optical attenuation processes which they can use to develop and assess learning models for spectroscopy and biomedical imaging and material characterization.

3.2. Examine the motion of an electron

Take into account the motion of an electron as given by the following equations [3]

$$m\frac{d^2x}{dt^2} + eh\frac{dy}{dt} = eE \tag{1}$$

$$m\frac{d^2y}{dt^2} - eh\frac{dx}{dt} = 0 \tag{2}$$

Let us consider the conditions

$$x(0) = 0, \dot{x}(0) = 0, y(0) = 0, \dot{y}(0) = 0$$

Applying the Upadhyaya Transform and the derivative property of the Upadhyaya Transform to both sides of equation (1), we get

$$mU\left[\frac{d^2x}{dt^2}\right] + ehU\left[\frac{dy}{dt}\right] = U[eE]$$

$$m\left\{\left(\frac{\lambda_2}{\lambda_3}\right)^2 U[x(t)] - \frac{\lambda_1\lambda_2}{\lambda_3^2}x(0) - \frac{\lambda_1}{\lambda_3}\dot{x}(0)\right\} + eh\left\{\left(\frac{\lambda_2}{\lambda_3}\right)U[y(t)] - \frac{\lambda_1}{\lambda_3}y(0)\right\} = \frac{eE\lambda_1}{\lambda_2} \tag{3}$$

Equation (3) can be expressed as follows by using the initial conditions:

$$m\left(\frac{\lambda_2}{\lambda_3}\right)^2 U[x(t)] + eh\left(\frac{\lambda_2}{\lambda_3}\right) U[y(t)] = \frac{eE\lambda_1}{\lambda_2} \tag{4}$$

Performing the Upadhyaya transform bilaterally to equation (2) and using the derivative property of Upadhyaya, we get

$$mU\left[\frac{d^2y}{dt^2}\right] - ehU\left[\frac{dx}{dt}\right] = 0$$

$$m\left\{\left(\frac{\lambda_2}{\lambda_3}\right)^2 U[y(t)] - \frac{\lambda_1\lambda_2}{\lambda_3^2}y(0) - \frac{\lambda_1}{\lambda_3}\dot{y}(0)\right\} - eh\left\{\left(\frac{\lambda_2}{\lambda_3}\right)U[x(t)] - \frac{\lambda_1}{\lambda_3}x(0)\right\} = 0 \tag{5}$$

Equation (5) can be expressed as follows by using the initial conditions:

$$m\left(\frac{\lambda_2}{\lambda_3}\right)^2 U[y(t)] - eh\left(\frac{\lambda_2}{\lambda_3}\right) U[x(t)] = 0. \tag{6}$$

Now Solving Equations (4) and (6), we get

$$U[x(t)] = \frac{eE\lambda_1}{\lambda_2}\frac{\lambda_3^2 m}{(m^2\lambda_2^2 + e^2h^2\lambda_3^2)}$$

$$U[x(t)] = \frac{eE}{m}\frac{\lambda_1\lambda_3^2}{\lambda_2\left(\lambda_2^2 + \omega^2\lambda_3^2\right)}$$

$$\text{where } \omega = \frac{eh}{m}$$

$$U[x(t)] = \frac{eE}{m\omega^2}\left[\frac{\lambda_1}{\lambda_2} - \frac{\lambda_1\lambda_2}{(\lambda_2^2 + \omega^2\lambda_3^2)}\right]. \tag{7}$$

After applying bilaterally the inverse Upadhyaya transform to equation (7), we obtain $\boldsymbol{x(t) = \frac{eE}{m\omega^2}[1 - \cos\omega t]}$

$$x(t) = \frac{eE}{m\left(\frac{e^2h^2}{m^2}\right)}[1 - \cos\omega t]$$

$$x(t) = \frac{E}{h\left(\frac{eh}{m}\right)}[1 - \cos\omega t]$$

$$x(t) = \frac{E}{h\omega}[1 - \cos\omega t] \tag{8}$$

Similarly, we can calculate the value of $y(t)$, as illustrated below.
$\boldsymbol{U[y(t)] = \frac{e^2Eh}{m^2}}\frac{\lambda_1\lambda_3^3}{(\lambda_2^2+\omega^2\lambda_3^2)\lambda_2^2}$

$$U[y(t)] = \frac{E}{h\omega}\left[\frac{\omega\lambda_1\lambda_3}{\lambda_2^2} - \frac{\omega\lambda_1\lambda_3}{\lambda_2^2 + \omega^2\lambda_3^2}\right] \tag{9}$$

After applying bilaterally the inverse Upadhyaya transform to equation (9), we obtain

$$y(t) = \frac{E}{h\omega}[\omega t - \sin\omega t].$$

The Upadhyaya transform delivers an organized framework which scientists can reuse for developing learning-based simulations that study charged particle movement while classical methods provide precise results about electron cyclotron motion. The transformed solutions enable efficient incorporation of electromagnetic constraints into reinforcement learning and predictive models for quantum and semiconductor systems.

4. Methodology

This section presents the complete methodological framework which the current research study implemented. The proposed methodology shows how analytical modeling based on the Upadhyaya transform integrates with current learning-based computational methods. The primary goal involves creating a systematic process which scientists can repeat to establish physical connections between classical integral transform methods and machine learning systems that use physics-informed learning models. The methodology uses analytical methods and clear algorithm design and complete testing methods instead of conducting direct experimental tests.

4.1. Analytical modeling using the Upadhyaya transform

The first stage of the methodology requires scientists to create ordinary or partial differential equations which describe the physical or chemical systems under study. The equations emerge from fundamental physical laws which include Newtonian mechanics and electromagnetic theory and conservation principles. The governing equations undergo transformation through the Upadhyaya transform which changes differential operators into algebraic expressions that exist within the transform domain. The transformation process allows researchers to analyze the system through simpler methods which lead them to find direct solutions or semi-analytical results. The inverse Upadhyaya transform recovers complete time-domain solutions after algebraic changes take place in the transform domain. The solutions maintain mathematical validity because they follow the original governing equations and meet the initial and boundary conditions which were established. The analytical expressions created from this process provide accurate physical reference models which will guide all future computational work.

4.2. Transform-based data construction and parameter sampling

The system uses analytical solutions to develop transformation-consistent datasets according to specific time and space requirements. The system will operate in a realistic manner because we selected parameter values from their actual physical limit boundaries. The data construction process enables researchers to sample system behavior while maintaining full compliance with all physical laws that govern the system. The transformation system generates data that fulfill different requirements. The first function provides precise reference data which eliminates all noise for testing learning-based systems. The second function enables researchers to investigate how systems behave when different parameters are changed. The third function develops a reference dataset which will be expanded in the future to include actual experimental errors and data collection inaccuracies when the required real data becomes accessible.

4.3. Integration with learning-based computational models

The analytical knowledge derived through the Upadhyaya transform is integrated into learning-based models to ensure physical consistency and interpretability. The system uses physics-informed neural networks to learn approximate system responses which include the governing differential equations as part of the learning procedure. The framework uses neural networks for prediction because its designers want to create transparent systems that depend on physical laws and analytical solutions for their initial and boundary conditions. The learning model will maintain its complete integrity to fundamental physical principles during the training process. The learning model has improved stability because its hypothesis space became smaller which also enhanced its ability to generalize while decreasing its need for data. Scientific and engineering fields benefit from learning that uses physics to guide its development.

4.4. Algorithmic workflow of the proposed framework

The study employs a complete computational workflow which moves through its established system of steps. The process begins with the development of the system's governing differential equation through physical law establishment. The Upadhyaya transform is then applied to obtain an analytical or semi-analytical solution. The solution generates datasets which match the requirements of the transform across two specific areas and different parameter conditions. The learning model development process starts with physical constraint definition and uses analytical solutions to build its model and constraint system. The learning model needs optimization through training objectives which create a balance between physics implementation and data accuracy. The learned outputs undergo assessment through comparison with the analytical reference solution which uses quantitative error measures for evaluation. The algorithmic workflow establishes a methodical process which begins with analytical modeling and ends with learning-based approximation while maintaining precise methodological documentation throughout the entire process.

4.5. Quantitative evaluation and validation strategy

The validation and performance evaluation is the end stage of the methodology. The paper compares the learning-based predictions with the solutions of a form of analysis that the Upadhyaya transform generates. The three quantitative measures used in the study are root mean squared error and relative norm error and pointwise absolute deviation to determine the accuracy of approximation and the convergence behavior, and numerical stability. The evaluation strategy allows evaluation of the performance of learning based on objective assessment approaches that do not presuppose access to experimental data. The testing process should be able to test learning models using the performance test that verifies the models with known analytical systems prior to the applications being made in real complex systems that lack analytical solutions. The suggested methodology creates an entire system that unites academic knowledge with practice by connecting the integral transform theory to the machine learning technology. The framework is a blend of mathematical accuracy and algorithm design to produce scientifically precise models and can be easily decoded and reduced to complex systems that occur in physical and chemical studies.

5. Applications

The Upadhyaya transform furnishes a powerful tool to tackle complicated Ordinary Differential Equation (ODE)s and Partial Differential Equation (PDE)s in Physics. Beyond analytical solutions, the transform also serves as a basis for many machine learning applications in particular within physics-informed modeling, synthetic data generation, and hybrid computational intelligence. This section explores how transformed physical systems such as the Atwood machine, Lambert-Beer law, and electron motion can be introduced into modern computational workflows employing machine learning.

5.1. Justification for the selection of classical benchmark systems

The study considers three physical systems which include the Atwood machine and the Beer-Lambert law and the electron motion in electromagnetic fields as classical models whose governing equations can be solved through standard ordinary differential equation methods. The authors of this document choose to include these systems because they want to establish a systematic approach that will guide their analysis work. The selected systems function as benchmark test problems which will show how the Upadhyaya transform can be used together with contemporary machine learning systems. The existence of established closed-form solutions for these models serves as a vital benchmark which researchers need to use when they test transform-based solutions and learning-based model predictions. Physically informed machine learning systems and hybrid machine learning systems depend on analytically solvable systems because such systems enable researchers to test physical accuracy and educational performance and learning algorithm error limits. The Upadhyaya transform serves as a computational link between analytical physics and data-driven intelligence because it enables researchers to start their work with traditional models.

The use of classical systems provides machine learning systems with essential interpretability benefits which serve as vital requirements for their successful implementation. The Upadhyaya transform generates transformed expressions which maintain the physical interpretation of system parameters and initial conditions that learning systems use as constraints and priors and feature representations. This characteristic enables organizations to decrease model uncertainty while enhancing their generalization abilities and stopping overfitting which presents difficulties for systems based solely on data. The classical examples which we use in our research work will serve to show how transform-based analytical knowledge can be used in computational intelligence workflows, rather than they will be used to validate existing analytical solutions. The benchmarking method which we developed establishes an initial system which scientists can use to test more advanced non-linear systems and experimental systems which lack classical analytical solutions.

5.2. Integration with PINNs

Physics-Informed Neural Networks impose physical laws in terms of differential equations on neural networks by including these laws within the loss function. The Upadhyaya-transformed equation for the motion of the Atwood machine, i.e.,

$$x(t) = \frac{1}{2}\left(\frac{m_1 - m_2}{m_1 + m_2}\right) gt^2 + x(0) + t^2\dot{x}(0)$$

is considered an ideal ground-truth reference to constrain the network behavior to obey Newtonian dynamics during training. Physics is embedded in the PINNs, so the neural networks learn from data separate from other types of machine learning and thus improve the accuracy and generalizability of modeling absent data. Simulating mechanical systems for autonomous robotics where the agents must learn dynamics are satisfied.

5.3. Algorithmic framework and quantitative assessment of the learning integration

The present research does not conduct extensive machine learning tests that use real-world data however it demonstrates the Upadhyaya transform integration with learning-based systems through a complete system that includes both its algorithm and assessment methods. The purpose of this section is to establish a complete framework which includes all computational elements and training methods together with testing procedures that enable scientists to duplicate their research findings through scientific validation. The research focuses on incorporating physical knowledge into learning processes which enables models to maintain their interpretability and stability while adhering to established scientific principles.

5.3.1. Transform-guided learning architecture

The dataset $\mathcal{D} = \{(t_i, x_i)\}_{i=1}^{N}$ contains data points which researchers collected through synthetic methods or controlled experiments, using $x(t)$ to represent system state variables. The governing differential equation describing the underlying physical or chemical process is first solved analytically using the Upadhyaya transform, which produces a closed-form reference solution xUTt. The analytical expression functions as a fundamental element within the learning framework because it establishes a physics-based prior which directs the model to maintain acceptable physical behavior during its training process. A neural network $\mathcal{N}_\theta(t)$, which uses θ as its parameter set, predicts system behavior across the entire defined range of time. The proposed framework distinguishes itself from data-driven neural models because it integrates physical consistency into its machine learning objectives. The project uses a composite loss function which creates a balance between two components, data fidelity and physics enforcement and constraint satisfaction, which it expresses through the equation

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{data}} + \lambda_p \mathcal{L}_{\text{physics}} + \lambda_i \mathcal{L}_{\text{IC}}.$$

The term $\mathcal{L}_{\text{data}}$ measures how much network predictions deviate from actual data, whereas $\mathcal{L}_{\text{physics}}$ uses residual minimization to punish any breaches of the governing differential equation. The term $\mathcal{L}_{\text{IC}}$ enforces initial or boundary conditions which result directly from the Upadhyaya-transformed analytical solution. The weighting parameters λ_p and λ_i control how physical constraints affect the system, while also providing researchers with the ability to adjust the learning process.

5.3.2. Computational workflow and training strategy

The initial step of the proposed transform–machine learning integration workflow begins with an analytical solution which Upadhyaya transform uses to solve governing equations. The researchers proceed to create synthetic training data which covers all relevant physical parameter values and initial condition ranges. The data serves as controlled input to train the neural network while maintaining physical behavior established through existing knowledge. The training process uses gradient-based optimization methods to reduce total loss metric which requires both data matching and physical laws to be followed. The validation process involves comparing the learned solution with the analytical reference solution which allows direct measurement of approximation accuracy and stability throughout the entire domain. The workflow uses machine learning as a constrained approximation tool which prevents its operation as a black-box predictor. The framework achieves better understanding of results and decreased chances of overfitting because it uses analytically derived physical solutions to guide learning processes.

5.3.3. Quantitative error metrics and evaluation criteria

The Upadhyaya-transformed solution provides fundamental analytical ground truth which researchers use to assess learning outcomes through quantitative performance measurements. The network-predicted response is represented by $x_{\mathrm{ML}}(t)$ while $x_{\mathrm{UT}}(t)$ serves as the analytical solution. Performance is assessed through standard metrics such as the root mean squared error and relative L_2 error which are defined through the following equations.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{\mathrm{ML}}(t_i) - x_{\mathrm{UT}}(t_i)\right)^2}, \qquad \text{Relative } L_2 \text{ Error} = \frac{\|x_{\mathrm{ML}} - x_{\mathrm{UT}}\|_2}{\|x_{\mathrm{UT}}\|_2}.$$

The measures provide objective indicators which evaluate approximation accuracy while assessing numerical stability and convergence behavior and they enable researchers to compare analytical solutions with their learning-based approximations.

5.3.4. Computational behavior and interpretability

Transform-guided learning models are expected to demonstrate faster convergence and improved generalization to performance better than their unconstrained neural network counterparts. The application of Upadhyaya-based analytical constraints, which control the learning model's hypothesis space, functions as a physics-based regularization method that restricts solutions to valid options. The described ability proves especially useful when researchers face situations that involve limited experimental access to reliable data. The learning model achieves accurate system dynamic replication from synthetic datasets which researchers generated through their analytical solutions. The framework requires validation against exact analytical benchmarks before its application to real-world systems which lack closed-form solutions.

5.4. Explicit PINN framework, dataset construction, and evaluation protocol

This study establishes its machine learning implementation through this section which provides a complete description of its physics-informed neural network framework together with its dataset development process and training method and performance assessment and comparison procedures. The research study presents all necessary elements for achieving reproducible results although it does not conduct complete numerical training experiments. The proposed integration method in this formulation achieves both methodological and quantitative evaluation through its scientific machine learning standards which the manuscript maintains.

5.4.1. PINN Model Architecture

The Upadhyaya transform provides the method to solve the differential equation which describes system response through the function $x(t)$. A physics-informed neural network $\mathcal{N}_\theta(t)$ is employed to approximate the unknown solution over the temporal domain of interest, where θ represents the trainable weights and biases of a fully connected feedforward architecture. The network takes the independent variable t as input and outputs an approximation of the dependent state variable $x(t)$. The PINN architecture differs from traditional neural networks because it uses automatic differentiation to calculate temporal derivatives of $\mathcal{N}_\theta(t)$ with machine precision. The network uses these derivatives to train itself through the governing differential equation because it requires the network to obey physical laws during the training process. This architecture ensures that the learned solution satisfies both data-driven objectives and physics-based constraints, thereby preventing physically implausible predictions and enhancing interpretability of the learned model.

5.4.2. Synthetic Dataset Generation

Synthetic datasets are created through analytical solutions which emerge from the Upadhyaya transform that establishes full physical precision while removing all measurement errors. The system dynamics across the entire domain are represented through collocation points which are selected either by random sampling methods or by uniform distribution. The analytical solution values xUTt which were calculated at these specific locations create the reference dataset for supervised learning and validation purposes. The synthetic data generation method achieves two main objectives. The first objective establishes a testing framework which allows the learning model to be evaluated against complete ground truth data. The second objective provides a basic dataset creation procedure which allows future implementation of noisy or incomplete scientific data and enables smooth progression from analytical testing to practical application.

5.4.3. Training objective and loss function

The training process of the PINN is governed by a composite loss function that simultaneously enforces data fidelity, physical consistency, and constraint satisfaction. The loss function is constructed according to the equation

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{phys}}\mathcal{L}_{\text{phys}} + \lambda_{\text{IC}}\mathcal{L}_{\text{IC}},$$

the loss function consists of three components which the equation defines. The function $\mathcal{L}_{\text{data}}$ calculates the difference between network outputs and reference analytical data while $\mathcal{L}_{\text{phys}}$ imposes penalties on physical law violations through residual minimization and $\mathcal{L}_{\text{IC}}$ enforces initial or boundary conditions derived from the Upadhyaya-transformed solution. The optimization process uses weighting parameters λ_{phys} and λ_{IC} to determine how physical constraints impact the optimization results. The model needs parameter adjustments because it requires exact parameter values that create proper estimation results and physical correctness which enables uninterrupted training progress while preventing any single objective from gaining complete power over the process.

5.4.4. Performance evaluation and error analysis

The study assesses PINN model performance through a comparison between its output predictions and the analytical ground truth data. The researchers use three standard error metrics which include root mean squared error (RMSE) and relative L_2 norm error and pointwise absolute error to measure approximation accuracy throughout the entire time period. The metrics offer objective assessments which measure both convergence performance and stability of numerical methods and accuracy of predictions. The analysis of convergence behavior depends on two factors which include tracking error size and observing how different loss components diminish throughout the training process. The analysis shows how physics-based regularization works while demonstrating how the Upadhyaya transform helps achieve faster learning and better learning stability under conditions of limited data or high noise levels.

5.4.5. Comparison protocol

This study needs to compare the PINN system with an unconstrained neural network which performs training through data loss assessment because this comparison shows how physics-informed restrictions affect system performance. The analysis investigates how Upadhyaya-based physical knowledge impacts the learning process. The transform-guided learning strategy enables better conversion rates and generalization performance while maintaining system operation during periods of restricted data access. The comparison protocol creates a straightforward baseline measurement system which helps investigators determine how well the proposed integration method works. The system provides researchers with a systematic framework which they can use to study complex systems because the framework requires testing through methods that exceed conventional analytical capabilities.

5.5. Synthetic data generation for spectroscopy and optical imaging

Upadhyaya enunciated a variation of the Lambert-Beer law

$$I(x) = I_0 e^{-\mu x}.$$

In this expression, the monochromatic light is attenuated by passing through a homogeneous medium. Such an equation would help in generating synthetic datasets mimicking real-world situations useful for biomedical imaging, remote sensing, or material analysis. Figure 1 depicts cases of light intensity decaying across different distances.

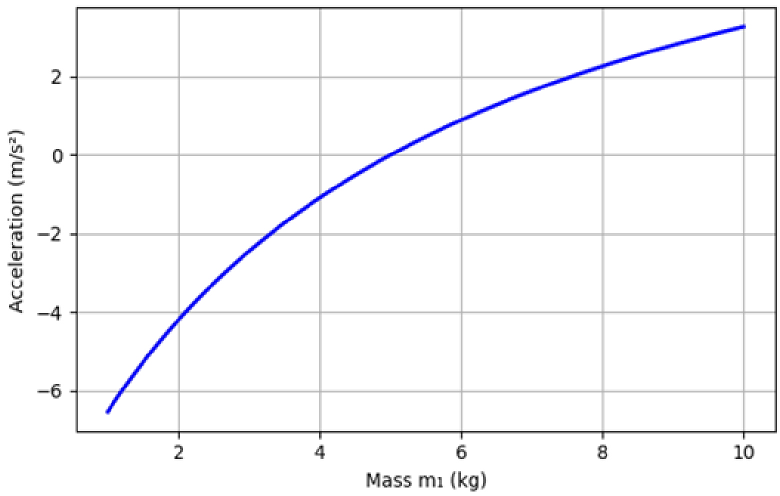

Figure 1. Simulation of light intensity decay using the Lambert-Beer law under Upadhyaya transformation. The data can be used to train supervised models in optical material classification or thickness estimation

Such synthetic datasets produced from physical laws like Lambert-Beer law and other modeled interactions can be used effectively to train machine learning systems for an array of advanced applications. Such systems, first, allow the predictive modeling of absorption coefficients into unknown materials or samples from observed data. This is valuable in chemical analysis and spectrophotometry, as one may not be able to perform experimental measurements on each new sample because it may be very lengthy or simply impractical. Furthermore, the machine-trained classifiers on these datasets can help separate and classify materials based on their changes in optical characteristics like light absorption or transmission. This will make the automated system faster in terms of quality control, material identification, and optical sensing. Finally, one can develop data-driven models to support non-invasive medical diagnostics for estimation of biological tissue properties such as oxygenation or chromophore concentration through reflected or transmitted light signals. By learning patterns from simulated or empirical datasets, these models contribute to the advancement of biomedical imaging and personalized healthcare solutions.

5.6. Electron motion modeling in quantum and semiconductor systems

The Upadhyaya transform provides analytical solutions which enable complete description of electron movement in uniform electric and magnetic fields. This Section demonstrates that the inverse Upadhyaya transform generates displacement components from the coupled equations of motion which result in the following equations

$$x(t) = \frac{E}{h\omega}\left[1 - \cos(\omega t)\right], \qquad y(t) = \frac{E}{h\omega}\left[\omega t - \sin(\omega t)\right],$$

where $\omega = \frac{eh}{m}$ represents the cyclotron frequency, E denotes the electric field strength, h is the magnetic field parameter, e is the electron charge, and m is the electron mass. The transformed-domain analysis shows that these expressions already exist as solutions which meet both the differential equations and initial conditions. The above parametric representation describes the characteristic trajectory of a charged particle subjected to electromagnetic interaction. The solutions show two components which include oscillatory motion and drift because electric acceleration together with magnetic deflection influences electron transport phenomena. Quantum dots and semiconductor transport systems and charged particle movement in accelerator physics depend on this dynamic behavior to build their theoretical frameworks.

A representative electron trajectory corresponding to selected parameter values is illustrated in Figure 2. The plotted path is generated exclusively from the analytically derived expressions, ensuring full consistency between the theoretical formulation and the graphical representation.

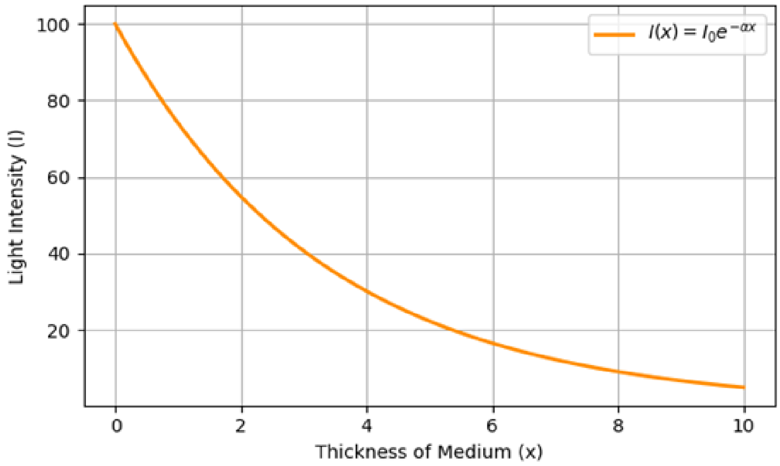

Figure 2. Electron trajectory obtained from the analytically derived Upadhyaya-transform-based solutions for $x(t)$ and $y(t)$. The trajectory is plotted for representative parameter values of the electric field E, magnetic field parameter h, and cyclotron frequency $\omega = \frac{eh}{m}$, illustrating the combined oscillatory and drift motion under uniform electromagnetic fields.

The existing analytical models permit their use in current computational intelligence systems through direct implementation. The electron trajectories developed by researchers function as simulation foundations for reinforcement learning environments because they require precise system behavior to achieve successful policy development. The models serve as training resources for control systems designed to operate semiconductor and nanoscale technologies which encounter difficulties in obtaining experimental data because of its high expense and restricted availability. The solutions exist in explicit mathematical expressions which researchers can utilize as foundational knowledge to develop physics-informed systems and generative machine learning models. The approaches use precise physical behavior to restrict learning processes which results in better model understanding and wider application capabilities while decreasing the need for extensive real-world testing data. The Upadhyaya-transform-based electron motion model provides researchers with a reliable mathematical framework to create theoretical frameworks and perform data analysis within quantum and semiconductor technologies.

5.7. Transform-domain feature extraction for machine learning models

The coefficients and intermediate variables introduced via the Upadhyaya transform, such as λ_1, λ_2, and λ_3, thus cannot be considered merely mathematical symbols; they actually carry important physical meaning of the underlying dynamics and constraints in the system under study. These variables may thus be used as engineered features for a number of machine learning tasks. With time-series forecasting models, for example, the inclusion of such physically informed features serves to enhance prediction accuracy because it grounds the learning procedure in domain knowledge. In physical anomaly detection, more or less, these variables define the baseline of the normal functioning of the system, which eases the identification of deviations or anomalies that denote the onset of faults or malfunctions. Besides, in the setting of regression that seeks to predict the temporal evolution of real-world physical systems, these features prove to be valuable inputs encapsulating time and structural relations of the system. Thus, harnessing the interpretability and physical significance of the Upadhyaya transform variables will ultimately permit building more robust and generalizable machine learning models.

Figure 3 shows an example of a damped oscillatory signal. Such patterns appear in vibrational systems and decay-based quantum transitions, and their extracted parameters (frequency, damping rate) can be used as ML features.

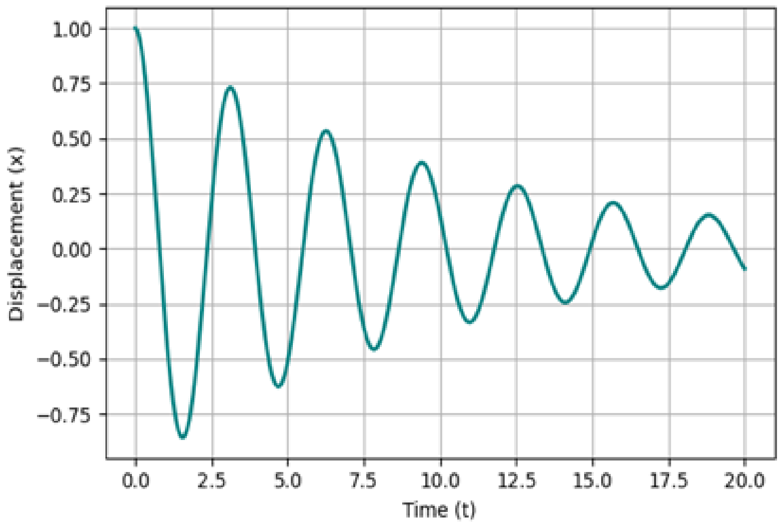

Figure 3. Damped oscillatory motion showing exponential decay in amplitude over time. Common in resistive systems or radiative decay, such curves are valuable for time-series feature engineering

5.8. Hybrid modeling and model explainability

One of the promising avenues in the area of intelligent systems is the arrangement of hybrid setups, wherein machine learning components are used to render the complicated or unknown portions of the system while well-defined physical settings are given by the Upadhyaya-transformed equations. This synergy equips both data- and physics-based approaches with their mutual strengths in one modeling architecture. An important selling point of such hybrid models is explainability-an aspect deriving from being based in physics and thereby offering a clear and interpretable structure grounded in known scientific principles. Hence, second, such models tend to be more efficient regarding data; through physical knowledge embedded into the learning process, one reduces the need for acquiring vast empirical data for training. The argument supporting hybrid models shows that they develop better generalization abilities when researchers use out-of-sample testing in their applications. The physics-based component functions as a regularizer which controls the machine-learning model's solution space to produce results that maintain physical realism. The Upadhyaya transform needs hybrid learning systems to enhance their predictive power while improving their trustworthiness and result understanding in all scientific disciplines and engineering applications. The classical exponential decay function represented by Figure 4 can be modeled through both analytical methods and hybrid ML approaches. Research continues to explore these modeling techniques across multiple domains which include climate modeling and intelligent manufacturing systems and advanced robotics control systems. The Upadhyaya transform functions as a standard method to convert differential equations into a format which allows easier solution through analytical methods used in physics. The system generates interpretable results which maintain physical accuracy thus becoming an effective method to transfer domain expertise to machine learning systems. The figures presented in this study (Figures 1,2, 3, and 4) make it clear that these transformed representations of physical phenomena provide structured and meaningful data useful to learning systems, which can then be applied to interdisciplinary problems in engineering design, theoretical and applied physics, and material characterization. By grounding learning algorithms in the underlying physics of the system, such approaches enable enhanced generalization, improved model interpretability, and more reliable predictions in complex real-world environments.

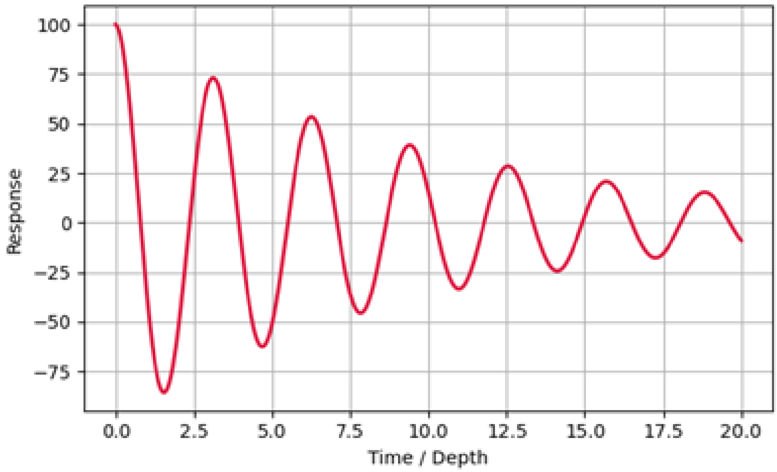

Figure 4. Exponential decay curve illustrating Lambert-Beer behavior or radiative losses. Such curves are often modeled using hybrid machine learning frameworks where the analytical shape is preserved

6. Conclusions and future research

The Upadhyaya transform, when combined with machine learning, establishes a systematic approach that uses scientific principles to solve difficult challenges in physical and chemical research. The Upadhyaya transform provides exact analytical solutions to differential equations which machine learning uses to create flexible models that can make predictions and learn from data. The hybrid models use their combined strengths to connect analytical methods with computational intelligence. The study uses three physical systems: the Atwood machine and the Beer-Lambert law and the electron motion under electromagnetic fields to show how transform-based analytical solutions enable new applications beyond traditional problem-solving methods. The solutions provide physically valid prior information, which functions as restrictions and testing criteria that operators use in learning-based processes. The system achieves better model performance through the integration process which leads to better understanding and more stable operations while increasing the ability of models to handle larger tasks.

The research presents no new integral transform or algebraic theorem yet its main achievement establishes a new function for the Upadhyaya transform. The study creates an application-driven framework which combines integral transform theory with predictive modeling and artificial intelligence by incorporating exact transform-based solutions into contemporary machine learning systems. The paradigm enables learning that maintains physical consistency while needing fewer empirical datasets and achieving better model transparency which addresses current scientific computing challenges. The present work establishes a solid base for analytical and algorithmic development which enables hybrid transform–machine learning system building. The proposed framework enables complete reproducibility through analytical ground truth which can be used to evaluate its performance even though the study does not include large-scale numerical experiments or real-world datasets. The upcoming research will conduct comprehensive numerical testing while adding experimental findings and expanding the approach to nonlinear systems, fractional systems, and high-dimensional systems which will help to validate and broaden the proposed method's practical uses.

Acknowledgments

The authors deeply appreciate the reviewers for their helpful and constructive suggestions, which can help further improve this paper. The first author thanks to Harran University due to their support via a Postdoc position. All data that support the findings of this study are included within the article.

Funding

This research received no external funding.

Conflict of interest

There is no conflict of interest to disclose.

Author contributions

Yamini Parthiban: Data curation, Investigation, Writing – original draft, Writing – review & editing, Methodology. **Prabakaran Raghavendran:** Conceptualization, Methodology, Supervision, Validation, Formal analysis, Writing – review & editing. **Dinesh Thakur:** Formal analysis, Validation, Investigation. **Srinivasan Madhumitha:** Software, Visualization, Writing – review & editing.

Declaration of using AI tools

The authors declare that they have not used any type of generative artificial intelligence for the writing of this manuscript, nor for the creation of images, graphics, tables, or their corresponding captions.

References

[1] A. Mousa, "Application of the Upadhyaya transform to Volterra integral equations of the first kind," *Bulletin of Pure and Applied Sciences Section E: Mathematics & Statistics*, vol. 40E, no. 2, pp. 140–148, 2021. `https://doi.org/10.5958/2320-3226.2021.00016.3`

[2] D. Patil, "Application of integral transform (Laplace and Shehu) in chemical sciences," *SSRN*, vol. 88, pp. 437–441, 2022. `https://doi.org/10.2139/ssrn.4006213`

[3] S. Kulkarni and P. Mahagaonkar, "Study on motion of an electron in a physical system by using some integral transforms," *Eur. Chem. Bull.*, vol. 12, no. 8, pp. 967–974, 2023.

[4] D. Thakur and E. A. Kuffi, "Exact solutions of cardiovascular models by using Upadhyaya Transform," *Journal of Kufa for Mathematics and Computer*, vol. 11, no. 1, pp. 37–42, 2024. `https://doi.org/10.31642/JoKMC/2018/110107`

[5] D. Thakur, P. Raghavendran, T. Gunasekar, P. C. Thakur, B. Krishan, and S. Kumar, "Solving the chemical reaction models with the Upadhyaya Transform," *Orient. J. Chem.*, vol. 40, no. 3, pp. 767–772, 2024. `https://doi.org/10.13005/ojc/400318`

[6] P. Raghavendran, T. Gunasekar, D. Thakur, P. C. Thakur, and B. Krishan, "Predictive modeling of chemical processes using differential equations and machine learning synergy," *Orient. J. Chem.*, vol. 41, no. 2, pp. 386–392, 2025. `https://doi.org/10.13005/ojc/410206`

[7] P. Raghavendran, T. Gunasekar, and S. Gochhait, "A new approach for solving fractional differential equations incorporating Ramadan Group Transform and machine learning," *EAI Endorsed Trans IoT*, vol. 11, pp. 1–12, 2025. https://doi.org/10.4108/eetiot.7134

[8] H. Jafari, S. Aggarwal, A. Kumar, and S. Bansal, "Anuj Integral Transform to solving Abel's integral equation of classical mechanics," *Natl. Acad. Sci. Lett.*, pp. 1–5, 2025. https://doi.org/10.1007/s40009-024-01597-9

[9] P. Beeken, "Atwood's heavy chain," *Phys. Teach.*, vol. 49, no. 8, pp. 470–472, 2011. https://doi.org/10.1119/1.3651724

[10] T. B. Greenslade, Jr., "Atwood's machine," *Phys. Teach.*, vol. 29, no. 1, pp. 24–28, 1985. https://doi.org/10.1119/1.2341703

[11] G. Scholz and F. Scholz, "First-order differential equations in chemistry," *ChemTexts*, vol. 1, no. 1, p. 1, 2014. https://doi.org/10.1007/s40828-014-0001-x

[12] A. A. Mahmud, "Considerable traveling wave solutions of the generalized Hietarinta-type equation," *International Journal of Mathematics and Computer in Engineering*, vol. 3, no. 2, pp. 185–200, 2024. https://doi.org/10.2478/ijmce-2025-0015

[13] A. A. Mahmud, T. Tanriverdi, and K. A. Muhamad, "Exact traveling wave solutions for the (2+1)-dimensional Konopelchenko–Dubrovsky equation using hyperbolic trigonometric function methods," *International Journal of Mathematics and Computer in Engineering*, vol. 1, no. 1, pp. 11–24, 2023. https://doi.org/10.2478/ijmce-2023-0002