Random Walk Generation and Classification Within an Online Learning Platform

Afrah Mousa
Department of Software Engineering
Bethlehem University,
Palestine
afrahmousa33@gmail.com

Thorsten Auth
Institution for Advanced Simulation
Forschungszentrum Jülich,
Germany
t.auth@fz-juelich.de

Anas Samara
Departmen of Software Engineering
Bethlehem University,
Palestine
asamara@bethlehem.edu

Suhail Odeh
Departmen of Software Engineering
Bethlehem University,
Palestine
sodeh@bethlehem.edu

Abstract: *Advancements in technology have introduced new approaches in teaching and learning processes. Machine learning algorithms analyse and recognize patterns of data and subsequently become able to make reasonable decisions. In playing complex games, such as chess and go, machine learning algorithms have even already outperformed humans. This paper presents a software platform 'Discimus$_{RW}$' that introduces a novel approach for teaching, learning, and researching random walk theory and getting hands-on experience in machine learning. Random walk theory represents the foundations of many fundamental processes, including the diffusion of substances in solvents, epidemics' spread, and financial markets' development. 'Discimus$_{RW}$' is composed of three main features: 1. Random walk generation using mathematical Equations, 2. Random walk classification using supervised learning algorithms, and 3. Random walk visualization. A few users who explored 'Discimus$_{RW}$' showed an interest and positive feedback that assured the experiential learning experience achieved using this software, which will therefore reinforce random walk teaching and learning.*

Keywords: *Random walk, machine learning, experiential learning*.

Received April 8, 2022; accepted April 28, 2022
https://doi.org/10.34028/iajit/19/3A/14

1. Introduction

Random walks are mathematical objects that consist of a sequence of random steps. A popular application is the characterization of Brownian motion, the random motion of particles dispersed in a solution, first described for pollen in water by the botanist Robert Brown in 1827 [3]. In 1905, Einstein's paper on the random walk theory of Brownian motion had an enormous impact on our view of the world: it provided evidence for the existence of atoms at a time when most scientists still believed that matter was a continuum [7]. Nowadays, the concept of random walks is nearly ubiquitous in science, engineering, and beyond. On the microscale, for example, the motility of cells under the microscope depends on the age of the donor [29]. Characterizing the swimming of bacteria helps us to understand how they find food [2], and studying sperm enables us to learn how they find the egg [16]. Artificial, micrometer-sized self-propelled particles may serve as intelligent drug-delivery vehicles in the future [27]. On the macroscale, the motion of pedestrians has random aspects, which can be crucial to designing evacuation routes [10].

Artificial intelligence is intelligence demonstrated by machines, with the Turing test as an early suggestion on how to decide whether a machine shows intelligent behaviour [32]. A landmark in the development of artificial intelligence was the Dartmouth workshop in 1956, where the term 'artificial intelligence' was first Introduced [23]. Nowadays, we often connect artificial intelligence with machine learning and handling 'big data', i.e., processing large amounts of data that cannot be analysed in a reasonable time frame using conventional approaches. Applications in daily life include a variety of tasks, such as optical character recognition [24], speech recognition [22], and shopping suggestions [20]. However, machine learning has also opened new routes and widened the horizons in scientific research. Scientific applications include but are not limited to protein folding [15], efficiently solving physics problems (e.g., modeling the weather) [30], self-driving cars [38], robotics [8], medical image analysis [34], and image classification in general [19].

E-Learning is a well-established format for learning, where distance learning has been adapted to the online environment [6]. Virtual course offers have been of very high interest recently when in-person courses had to be suspended in many places worldwide [33]. Compared with traditional formats, online formats offer novel possibilities for course design, such as introducing gamification into education [42]. However, learning based on own experience is vital to achieving a thorough understanding [18]. In particular, numerous studies on teaching in Science, Technology, Engineering, and Mathematics (STEM) highlight the importance of hands-on experiences. Thus, new approaches in STEM course development range from a biology lab course that aims to enhance experimental design skills [17] to a project-based course on statistical physics [21]. Furthermore, efforts are made to develop cost-effective tools for hands-on experiments; recent examples are the widely available paper-microscope 'Foldscope' [5] and instructions that allow constructing a microscope made of LEGO® bricks at home [43].

Unfortunately, today's E-learning courses often do not include experiential learning opportunities. At the same time, the wide availability of mobile phones, tablets, and computers allows us to develop digital tools for experiential learning without the need for expensive lab equipment. Although such websites, programs, and applications have often been developed to be applied in in-person courses, their easy accessibility allows teachers to include them also in E-Learning course modules. One example is phyphox, an application that enables teachers and students to conduct traditional physics experiments by exploiting the measuring capabilities of mobile phones [36]. Also, entirely virtual web applications have been developed, such as a multi-agent programmable modeling environment [39], a collection of interactive applications to explore complex systems [44], and many other virtual or remote laboratory experiments [14]. Machine learning is heavily based on computational statistics [13]; therefore, it should be well suited for developing digital, experiential learning opportunities.

In this paper, we present a web application that classifies random walks in two dimensions using machine learning. Users can easily visualize the 2D walks on the screen, which is convenient for judging the machine learning model's performance by eye and for rationalizing in which cases the model works well and when problems may occur. We use data sets consisting of step lengths and angles between consecutive steps as feature vectors. We have trained three machine learning models using Python and imported the models into a web application based on Django. Furthermore, we used several programming languages for the project, including Python 3, HTML/CSS, JavaScript, and SQL for the project database. For dynamic development, agile methodology has been adopted, which is a people-focused, results-focused approach to software development that respects our rapidly changing world.

The structure of the paper is as follows. In section 2, three types of random walks and the generation of trajectories is described. In section 3, the training of the machine learning models along with the evaluation metrics is discussed. In section 4, the software features and the development methodology of Discimus$_{RW}$ are presented. Finally, in section 5, a discussion and conclusions are given.

2. Random Walk Dataset Generation

An ideal random walk can be thought of as a trajectory of a diffusing particle, see Figure 1. Each trajectory can be represented by a set of points, obtained by determining the particle position in regular time intervals using a microscope. It can be characterized by the size of the particle and the viscosity of the fluid used for the experiment. However, in addition to these two well controlled experimental parameters, the motion of a particle will be subject to thermal noise. Therefore, trajectories of equal length recorded independently of each other can-on the first view-appear to be fundamentally different. Mathematically, a simple random walk can be generated by N consecutive steps that are connected with random angles. For machine learning, the random walk can be represented as a feature vector composed of the set of step lengths and angles.

However, for example in biological systems, the random motion of cells may not be well described using ideal random walks. For example, swimming *E.coli* bacteria move straight driven by their flagellar propulsion and randomly change their direction of motion by tumbling in irregular time intervals. In the presence of a food gradient, the bacteria preferably move in the direction of increasing food concentration. Their trajectories can be described by persistent and biased random walks, respectively, see Figure 1. We mimic the trajectory of a persistent random walk by consecutive steps of fixed average lengths, connected with angles randomly chosen in a restricted range, $\alpha < \alpha_{max}$. For biased random walks, the restricted random angles are determined with respect to a specified, fixed direction in space.

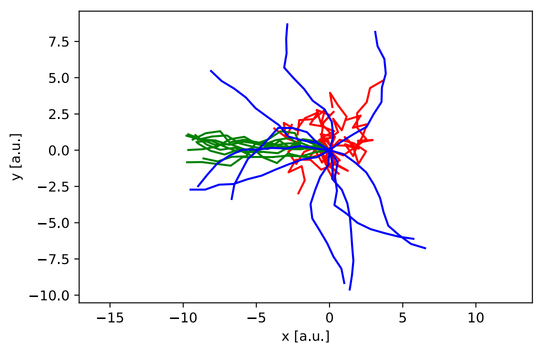

Figure 1. Random walk trajectories with fixed step length 1: ideal (red), persistent (blue), and biased random walk (green).

3. Random Walk Classification

A wide range of supervised machine learning techniques exists that can be used for building predictive models for random walk trajectories. The random walk feature vectors are classified by identifying patterns in the sets of step lengths and angles to predict the random walk type, see Figure 2 several machine learning models have been trained using a random walk dataset that contains on average 300 walks of each type with N=20 steps and a_{max}=20º. In this study, further supervised learning techniques were examined in addition to the three techniques that were explored in the initial study [25].

Various classifiers have been investigated in the current study, see Table 1.

1. A logistic regression classifier that comprises classification trees with logistic regression functions at the leaves [40].
2. A neural network, a multi-layer perceptron with backpropagation [12]. Here, 200 iterations for training three layers [512 x 20 x 10] with the ReLu activation function were used for the random walk classification.
3. A K-Nearest Neighbor (KNN) algorithm, where the classification is made according to type of most local neighbors [4].
4. Naïve Bayes, a probabilistic classifier based on Bayes' theorem with the assumption of feature independence [31].
5. A decision tree, which splits the data into nodes according to class purity- making the dataset features the branches, and the dataset targets as the leaves of the tree [28].

Moreover, ensemble learning techniques were exploited in the experiments as well [41].

6. Random forest, an ensemble supervised learning technique that generates a multitude number of decision trees whereas the decision of the random forest is the decision selected by most trees [11]; 10 trees were used in the Random Forest classifier.
7. Adaboost, yet another ensemble supervised learning technique that combines weak learners, where each learner adapts a data sample to boost the performance [9], 50 weak learners were constructed for the AdaBoost classifier.

The classification models were evaluated using stratified cross-validation with 10 folds. The classification performance metrics have been adopted in this study to evaluate the performance of different predictive models:

1. Classification Accuracy (CA), the fraction of correct predictions to the total number of predictions [1].
2. Precision, the fraction of true positives to the total number of positive predictions, which shows the quality positive predictions [37].
3. Recall, the fraction of positive predictions to the total number of positive cases [37].

Table 1 presents a summary of the evaluations for the aforementioned seven trained models. The Neural network classifier achieved the highest accuracy as 97.31%. Moreover, the ensemble learners (i.e. AdaBoost and Random forest) achieved a similar classification accuracy. The logistic regression classifier performance was worst.

Table 1. Results of classification models using stratified 10-fold cross-validation.

Model	Accuracy	Precision	Recall
AdaBoost	97.21%	97.21%	97.21%
kNN	93.22%	93.39%	93.22%
Log-Regression	57.23%	58.03%	57.23%
Naive Bayes	75.57%	75.36%	75.57%
Neural Network	97.31%	97.37%	97.31%
Random Forest	96.41%	96.51%	96.41%
Tree	97.21%	97.29%	97.21%

Table 2. Confusion matrix of the random walk classification model using neural network.

		Predicted		
		Biased	Ideal	Persistent
Actual	Biased	286	0	20
	Ideal	0	267	2
	Persistent	5	0	423

The confusion matrix of the neural network model, given in Table 2 shows the misclassified instances, where 20 biased random walk instances were incorrectly classified as persistent, and 2 ideal random walk instances classified as persistent. Additionally, 5 persistent random walks were classified as biased random walk. Consequently, one can tell that there is a similarity in the patterns between persistent and biased random walks, which can be inferred visually from Figure 1.

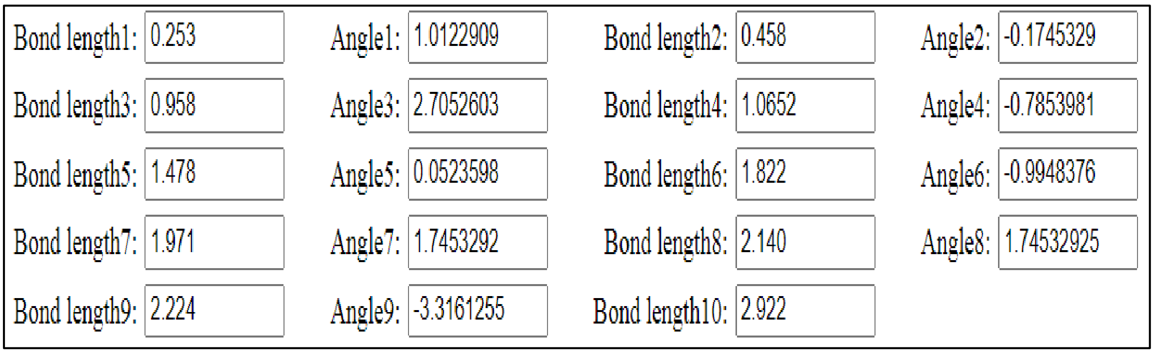

Figure 2. Form for entering step lengths and angles between subsequent steps of random walks, which constitute the feature vector.

4. DiscimusRW, E-Learning Platform for Random Walk

Discimus$_{RW}$ is a web-based online system based on Django web technology to support teaching random walks and machine learning. The application uses machine learning models to classify 2D random walks that a user can provide. In a teaching setting, the application can for example be applied to determine the sizes of spherical particles that diffuse based on their trajectories.

4.1. Discimus$_{RW}$ Features

Discimus$_{RW}$ comprises two roles: admins (e.g., teachers) and students associated with an admin. In this section, we focus on the functionality that the application offers to students, see Figure 3. Students can enter step lengths and angles between steps as feature vectors representing random walks, visualize, and classify random walks. For classification, the application has been trained using data sets of random walk trajectories generated with the help of mathematical models. Beyond the basic functionality, the application allows users to download the plots of their random walks and classification results.

a. *Entering a data set for a random walk*
Users can enter feature vectors for random walks via their step lengths and angles between subsequent steps using an online form, see Figure 1. For example, these values can be extracted from trajectories of the Brownian motion of passive particles observed through a microscope or of the active motion of swimming bacteria. Alternatively, the application offers default values as examples for particular types of random walks, which an admin provides.

b. *Visualize a random walk trajectory*
Users can visualize the random walk for the data set they provided on screen, see selected trajectories shown in Figure 2. Visualization is an important step for both experimental and simulation data. Furthermore, seeing the random walks allows users to attempt to classify the data by eye, "This looks like a biased random walk.", and to later better understand the classification result that the application provides.

c. *Classify a rndom walk trajectory*
Users can classify random walks for the feature vectors they provided. The application allows the user to select one of several types of pre-trained machine learning models available. Different stages of training can be chosen for each of the models, which will allow the user to directly experience the improved classification if the model has been trained with more data.

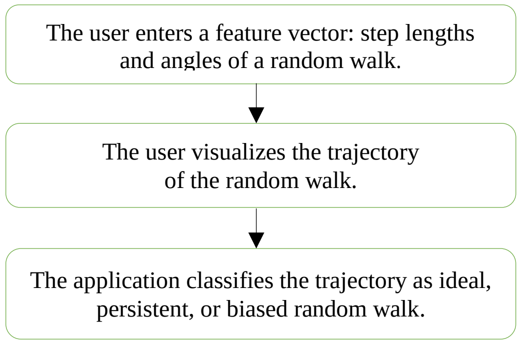

Figure 3. Workflow for visualization and classification applied in Discimus$_{RW}$.

4.2. Development Methodology

Discimus$_{RW}$ has been developed to serve as a tool for teaching. However, app development has also been an educational project, conducted as a thesis project at Bethlehem University. The application requirements were formulated based on interviews with teachers and students in both Palestine and Germany. Furthermore, requirements that were not clear at the beginning were clarified by involving users in the development process and obtaining their review and feedback after completing each step. Therefore, agile software development was adopted because it is flexible, fast, and keeps the project updated with changes of the environment. It aims at continuous improvements in quality, using practices like Scrum and extreme programming. Agile development contributed to the completion of the application in a shorter and faster time and allowed us to give the stakeholders and potential customers qualified services. All this led to customer satisfaction with the application.

The Agile methodology depends on continuous communication with customers, which has been done in this project [35]. Weekly meetings were held with customers to record their reactions and reviews about the application; these reviews and reactions were used to improve the application quality and develop what the customers might like.

The Agile methodology was followed to gain the agile development values:

1. Planning: The customers' requirements are taken and classified, in addition to identifying the specific customers for the project and the technologies that will be used for implementing the project.
2. Design: The software development diagrams are made, which explain and clarify how the program works, such as a workflow diagram.
3. Development: The design is translated into code, and the application constructs become alive. All programming languages and tools for the program development are selected, including the operating system, the Integrated Development Environment (IDE), and more.
4. Testing: The accomplishments of the project are verified and validated by regularly presenting it to the

customers. Reactions, reviews, and customer suggestions are recorded and considered for the project's subsequent development.

5. Deployment: The project is completed and released to the end-users in the production environment. Upon completion of all steps, the activities cycle was repeated, as the project was divided into smaller parts to facilitate working on them and conducting work faster.

Agile testing was used to ensure that the project was running correctly and reliably. Regular testing after each agile iteration facilitated monitoring and inspecting vulnerabilities. Furthermore, the application was regularly validated by user acceptance testing, which determines whether the application conforms to the users' requirements collected in the interviews throughout the development process, and not only just before deploying the application for use.

5. Discussion and Conclusions

Brownian motion can be easily demonstrated in high schools at the level of diffusion of potassium permanganate in cold and hot water; even single-particle motion can be observed by students watching milk under the microscope. Unfortunately, neither observing collective diffusion nor following the motion of single particles under the microscope allows students to rationalize random walks using probability theory and statistical error. Similarly, artificial intelligence is abundant in everyday life, but the concept of predicting the behaviour of systems for which noise plays a crucial role appears mysterious to many. One of the main reasons may be that statistics, the basis of machine learning as the modern approach to implementing artificial intelligence, is not very popular. Despite the growing importance of machine learning in research for applications, teaching statistics is often underrepresented in high schools and universities outside the mathematics' core discipline.

Using machine learning to classify trajectories of random walks, which can be obtained from tracing particles that exhibit Brownian motion, is a natural combination for including digital tools in in-person and E-Learning courses. Brownian motion and diffusion taught in high schools are relevant for all three classical disciplines: biology, chemistry, and physics. Furthermore, the importance of probability and statistics is immediately evident by simply comparing trajectories of a few walks. At the same time, the approach of classifying single trajectories using machine learning is novel and conceptionally different from classical analysis approaches, such as mean-squared displacements calculated for ensembles of walks. We developed Discimus$_{RW}$, a web application to provide an experiential learning opportunity for statistics using random walks and machine learning. The requirements and specifications for the application were extracted from interviews with teachers and students in Palestine and Germany and continuously refined during the development. We used agile methodology, which is based on adaptive planning and self-organization; a short delivery time was exploited to manage the software development activities.

From a technical point of view, Discimus$_{RW}$ allows students to visualize and classify two-dimensional random walks. Feature vectors that describe the trajectories can either be entered by the student or suggested by the application. Various random walks can be generated using mathematical models without the need for experiments or high-level computing resources. Several machine learning algorithms were used for classification, including the classical algorithms K-nearest neighbours, Gaussian Naïve Bayes, and decision tree learning. Users can select the model with that they would like to analyse their data and compare the performance of the three algorithms and experience the importance of the size of the training set.

From a teaching point of view, our work addresses a wide range of subject fields in STEM education. The application is not limited to natural science students; it can also be used for students in computer science to teach machine learning. For teaching natural sciences, random walks are ideal for experiential teaching statistics and machine learning because they are taught regularly in science departments in universities, scientific institutes, and some high schools. Furthermore, an experiential approach to understanding the importance of statistics often cannot be achieved by wet-lab experiments but is readily provided by information technology. For teaching computer science, the application contains several machine learning models that enable students to compare their classification performance metrics. The students will thus understand the importance of feature vectors, training, and algorithms. Teachers that act as admins can tailor the user experience according to the level of education of their group of students.

We anticipate Discimus$_{RW}$ to provide an engaging way for teaching random walks and machine learning in schools and universities; in particular, the task of machine learning random-walk trajectories is also of current research interest [26]. The follow-up steps to establish Discimus$_{RW}$ for education and training will be-in addition to further technical development-to develop accompanying pedagogical concepts and supply teaching material for the various target groups.

Acknowledgment

All authors thank Sherin Hazboun (Bethlehem) for helping to initialize the collaboration and Rawan Gedeon (Bethlehem) for discussions on machine learning. A.M.M. acknowledges support by the Palestinian-German Science Bridge (PGSB).

References

[1] Alvarez S., "An Exact Analytical Relation among Recall, Precision, and Classification Accuracy in Information Retrieval," *Boston College, Boston*, Technical Report BCCS-02-01, pp. 1-22, 2002.

[2] Berg H., and Berry R., "E. Coli in Motion," *Physics Today*, vol. 58, no. 2, 2005.

[3] Brown R., "XXVII. A Brief Account of Microscopical Observations Made in the Months of June, July and August 1827, on the Particles Contained in the Pollen of Plants; and on the General Existence of Active Molecules in Organic and Inorganic Bodies," *The Philosophical Magazine*, vol. 4, no. 21, pp. 161-173, 1828.

[4] Cover T. and Hart P., "Nearest Neighbor Pattern Classification," *IEEE Trans Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.

[5] Cybulski J., Clements J., and Prakash M., "Foldscope: Origami-Based Paper Microscope," *PLoS ONE*, vol. 9, no. 6, 2014.

[6] Downes S., "E-learning 2.0," *ELearn*, vol. 2005, no. 10, pp. 1, 2005.

[7] Einstein A., "Über Die Von Der Molekularkinetischen Theorie Der Wärme Geforderte Bewegung," *Annalen der Physik*, vol. 4, 1905.

[8] Finn c. and Levine S., "Deep Visual Foresight for Planning Robot Motion," *in Procceding of the IEEE International Conference on Robotics and Automation*, Marina Bay Sands, pp. 2786-2793, 2017.

[9] Freund Y., Schapire R., and Abe N., "A Short Introduction to Boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, pp. 1612, 1999.

[10] Helbing D. and Molnár P., "Social Force Model for Pedestrian Dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282, 1995.

[11] Ho T., "Random Decision Forests," *in Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, pp. 278-282, 1995.

[12] Hu Y. and Tsai J., "Backpropagation Multi-Layer Perceptron for Incomplete Pairwise Comparison Matrices in Analytic Hierarchy Process," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 53-62, 2006.

[13] James R., Witten G., Hastie D., and Tibshirani T., *An Introduction to Statistical Learning, with Applications in R*, Springer, 2013.

[14] Jong T., Sotiriou S., and Gillet D., "Innovations in STEM Education: The Go-Lab Federation of Online Labs," *Smart Learning Environments*, vol. 1, no. 1, 2014.

[15] Jumper J., Evans R., Pritze l, Green T., Figurnov M., Figurnov O., Tunyasuvunakool K., Bates R., Žídek A., Potapenko A., Bridgland A., Meyer C., Kohl S., Ballard A., Cowie A., Romera-Paredes B., Nikolov S., Jain R., Adler J., Back T., Petersen S., Reiman D., Clancy E., Zielinski M., Steinegger M., Pacholska M., Berghammer T., Bodenstein S., Silver D., Vinyals O., Senior A., Kavukcuoglu K., Kohli P., and Hassabis D., "Highly Accurate Protein Structure Prediction with Alphafold," *Nature*, vol. 596, no. 7873, pp.583-589, 2021,

[16] Kholodnyy V., Gadêlha H., Cosson J., and Boryshpolets S., "How Do Freshwater Fish Sperm Find the Egg? The Physicochemical Factors Guiding the Gamete Encounters of Externally Fertilizing Freshwater Fish," *Reviews in Aquaculture*, vol. 12, no. 2, pp. 1165-1192, 2020.

[17] Killpack T., Fulmer S., Roden J., Dolce J., and Skow C., "Increased Scaffolding and Inquiry in an Introductory Biology Lab Enhance Experimental Design Skills and Sense of Scientific Ability," *Journal of Microbiology and Biology Education*, vol. 21, no. 2, pp. 21-2, 2020.

[18] Kolb D., *Experiential Learning: Experience as the Source of Learning and Development* Prentice Hall, 1984.

[19] Krizhevsky A., Sutskever I., and Hinton G., "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[20] Kumar V., Rajan B., Venkatesan R., and Lecinski J., "Understanding the Role of Artificial Intelligence in Personalized Engagement Marketing," *California Management Review*, vol. 61, no. 4, pp. 135-155 2019.

[21] Langbeheim E., Abrashkin A., Steiner A., Edri H., Safran S., and Yerushalmi E., "Shifting the learning gears: Redesigning A Project-Based Course on Soft Matter Through the Perspective of Constructionism," *Physical Review Physics Education Research*, vol. 16, no. 2, pp.020147, 2020.

[22] Lippmann R., "Review of Neural Networks for Speech Recognition," *Neural Computation*, vol. 1, no. 1, pp. 1-38, 1989.

[23] Moor J., "The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years," *AI Magazine*, vol. 27, no. 4, pp. 87-87, 2006.

[24] Mori S., Suen C., and Yamamoto K., "Historical Review of OCR Research and Development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029-1058, 1992.

[25] Mousa A., Auth T., Samara A., and Odeh S., "Discimus RW: An E-Learning Web Application for Classifying Random Walks with Machine Learning," *in Proceeding of the 22nd International Arab Conference on Information Technology*, Muscat, pp. 1-5, 2021.

[26] Muñoz-Gil G., Volpe G., García-March M., Metzler R., Lewenstein M., and Manzo C., "The

Anomalous Diffusion Challenge: Single Trajectory Characterisation as A Competition*," in Proceeding of the Emerging Topics in Artificial Intelligence*, California, pp. 42-51, 2020.

[27] Patra D., Sengupta S., Duan W., Zhang H., Pavlick R., and Sen A., "Intelligent, Self-Powered, Drug Delivery Systems," *Nanoscale*, vol. 5, no. 4, pp. 1273-1283, 2013.

[28] Peng W., Chen J., and Zhou H., "An implementation of ID3-Decision Tree Learning Algorithm," *From web. Arch. Usyd. Edu. Au/Wpeng/Decisiontree2. Pdf Retrieved date: May*, vol. 13, 2009.

[29] Phillip J., Zamponi N., Phillip M., Daya J., McGovern S., Williams W., Tschudi K., Jayatilaka H., Wu P., Walston J., and Wirtz D., "Fractional Re-Distribution Among Cell Motility States During Ageing," *Communications Biology*, vol. 4, no. 1, PP. 1-9, 2021.

[30] Raissi M. and Karniadakis G., "Hidden Physics Models: Machine Learning of Nonlinear Partial Differential Equations," *Journal of Computational Physics*, vol. 357, PP. 125-141, 2018.

[31] Rish, "An Empirical Study of the Naive Bayes Classifier," *in Proceeding of the IJCAI 2001 workshop on empirical methods in artificial intelligence*, Seattle, pp. 41-46, 2001.

[32] Saygin A., Cicekli I., and Akman V., "Turing test: 50 years later," *Minds and Machines*, vol. 10, no. 4, pp. 463-518, 2000.

[33] Shahzad A., Hassan R., Aremu A., Hussain A., and Lodhi R., "Effects of COVID-19 in E-learning on Higher Education Institution Students: the Group Comparison between Male and Female," *Quality and Quantity*, vol. 55, no. 3, pp. 805-826, 2021.

[34] Shen D., Wu G., and Suk H., "Deep Learning in Medical Image Analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp.221, 2017.

[35] Singh M., "U-SCRUM: An Agile Methodology for Promoting Usability," *in Proceeding of the Agile 2008 Conference*, Toronto, pp. 555-560 2008.

[36] Staacks S., Hütz S., Heinke, and Stampfer C., "Advanced Tools for Smartphone-Based Experiments: Phyphox," *Physics Education*, vol. 53, no. 4, pp. 045009, 2018.

[37] Tharwat A., "Classification Assessment Methods," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 168-192, 2020.

[38] Tian Y., Pei K., Jana S., and Ray B., "DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars," *in Proceedings of the International Conference on Software Engineering*, pp.303-314, 2018.

[39] Tisue S. and Wilensky U., "Netlogo: A Simple Environment for Modeling Complexity," *in Proceeding of the Conference on Complex Systems*, pp. 16-21, 2004.

[40] Tsangaratos P. and Ilia I., "Comparison of a Logistic Regression and Na\"\Ive Bayes Classifier in Landslide Susceptibility Assessments: the Influence of Models Complexity and Training Dataset Size," *Catena (Amst)*, vol. 145, pp. 164-179, 2016.

[41] TÜYSÜZOĞLU G. and Birant D., "Enhanced Bagging (Ebagging): A novel Approach for Ensemble Learning," *The International Arab Journal of Information Technology*, vol. 17, no. 4, 2020.

[42] Urh M., Vukovic G., Jereb E., and Pintar R., "The Model for Introduction of Gamification into E-learning in Higher Education," *Procedia - Social and Behavioral Sciences*, vol. 197, pp. 388-397, 2015.

[43] Vos B., Blesa E., and Betz T., "Designing a High-Resolution, LEGO-Based Microscope for an Educational Setting," *The Biophysicist*, vol. 2, no. 3, pp. 29-40, 2021.

[44] "Complexity Explorables." https://www.complexity-explorables.org/, last visited 22021.

Afrah Mousa is a recent college graduate with a Bachelor's Degree in Software Engineering from Bethlehem University – Palestine. She works as a Machine Learning Engineer at Jereer Company-Palestine. Her main interest in the field of artificial intelligence and machine learning, and this is what prompted her to get a job related to it. Afrah enjoys using my skills to contribute to the exciting technological advances that happen at Jereer.

Thorsten Auth received his Ph.D. in Theoretical Physics from University of Cologne in 2004. He is currently a staff scientist at Forschungszentrum Jülich, where he works on theoretical physics of living matter. In particular, he focuses on computer simulations to study dynamics and self-organisation in systems containing self-propelled particles and on nanostructure-membrane interactions.

Anas Samara is an experienced Software Engineer with a demonstrated history of working in the software industry. Competent researcher in a wide range of computer science topics with a Doctor of Philosophy (PhD) from Ulster University. Proficient in data mining and development of supervised and ensemble predictive models. Main research interests are focused on Affective Computing and Human-Computer Interaction disciplines. Has many publications in affective state detection within a human–computer interaction contexts using different input channels and modalities to improve the quality of interaction through generating a more engaging user experience and intelligent and adaptive interfaces.

Suhail Odeh Born in Bethlehem - Palestine, earned PhD degree (2006) in Computer Engineering from the Department of Computer Architecture and Technology, University of Granada -Spain. He joins Bethlehem University community as an assistant professor in the Computer and Information System department in the faculty of science at (2006). Dr. Odeh has been a Postdoctoral researcher in the Department of Information Engineering and Computer science at L'Aquila University, Italy. Also, he has been a visiting professor The Department of Computer Science at the University of Cyprus, University of Granada, and University of Salamanca. He is an active researcher in the field of Artificial intelligence, Pattern recognition, Intelligent Systems, Brain Computer Interface, Multiagent.