Machine Learning Recovers Corrupted Pharmaceutical 3D Printing Formulation Data

Olima Uddin[1], Yusuf Ali Mohammed[1], Simon Gaisford[2] and Moe Elbadawi[1,*]

[1]School of Biological and Behavioural Sciences, Queen Mary University of London, Mile End Road, London E1 4DQ, UK.

[2]UCL School of Pharmacy, University College London, 29-39 Brunswick Square, London WC1N 1AX, UK.

* Corresponding Author: m.elbadawi@qmul.ac.uk

Abstract

Pharmaceutical 3D printing is an emerging digital manufacturing technology capable of autonomously producing personalised medicines. However, the same reliance on digital workflows that enables this innovation also introduces new vulnerabilities, most notably is the risk of cyberattacks. In such scenarios, malicious actors could corrupt formulation data, either by deleting critical information or introducing subtle noise that is difficult to detect, potentially compromising patient safety. To address this challenge, we investigate the application of machine learning, specifically denoising autoencoders (DAEs), for the reconstruction of corrupted pharmaceutical formulation data. Our dataset comprises 1,623 formulations across 336 ingredients, totalling over 545,000 individual data points. To simulate potential cyberattack scenarios, we deliberately corrupted the dataset in two ways: (1) randomly removing 1%, 5%, or 10% of the data points, mimicking targeted data deletion, and (2) introducing noise across all data points, simulating tampering or injection attacks. We evaluate multiple DAE configurations and demonstrate their ability to recover corrupted data, achieving R^2 scores of 0.989 ± 0.0017, 0.983 ± 0.0031, and 0.976 ± 0.0007 for 1%, 5%, and 10% data loss, respectively. Among the parameters tested, the learning rate was found to have a significant effect on DAE performance. In contrast, a traditional machine learning approach (kNN) failed to produce positive R^2 values across all missingness levels, further demonstrating the superior performance of the DAE. Therefore, we demonstrate the potential of DAEs to safeguard formulation data against data

corruption, underline the broader role of machine learning in enhancing digital resilience and maintaining data quality across the pharmaceutical sector.

Keywords: Artificial Intelligence; Quality Control; Fused Deposition Modelling; Digital Resilience; Drug Development.

Introduction

Three-dimensional (3D) printing is poised to revolutionise pharmaceutical manufacturing [1-6]. It enables the precise digital fabrication of drug products and opens the door to on-demand, personalised medicines that can be tailored to individual patients' needs in terms of dose, release profile, and even physical form. This capability has wide implications for clinical settings, especially hospitals, where personalised treatment regimens can be manufactured directly on-site, reducing the need for large-scale centralised production and allowing for rapid therapeutic intervention [7-9].

3D printing is a digitalised manufacturing technology, which makes it amenable to automation and provides an opportunity to integrate artificial intelligence (AI) into the design, control, and quality assurance of pharmaceutical products. Recent work in the use of machine learning (ML), a subset of AI, has demonstrated that it has the potential to determine the formulation, optimise processing parameters and inspect the quality of the 3D printed product [10-12], thereby obviating the need for trial-and-error experiments in time sensitive applications and the dependence on continuous supervision by a user [13, 14]. However, this requires much trust in the AI-3D printing framework to operate unsupervised. Unfortunately, like all digital systems, pharmaceutical 3D printing is susceptible to cyberattacks, software bugs, and data corruption. Such cyberattacks against 3D printers are well documented in other sectors and hence there is potential for them to be applied in pharmaceutical 3D printers [15-17]. A malicious actor altering formulation files or printing parameters, for example, could unintentionally (or intentionally) affect the safety and efficacy of the final product, leading to serious clinical consequences. Even in the absence of intentional interference, poor data management, missing values, or noisy sensor inputs can compromise product quality. Given the life-critical nature of medicines, ensuring the integrity and reliability of digital manufacturing data is not just a technical challenge but. it is a public health imperative. However, few studies have explored the

use of AI for detecting corrupted pharmaceutical data [18] and, to the best of authors' knowledge, none have investigated the use of AI for tackling corrupted formulation data.

AI can be used to detect and repair data quality issues. In particular, techniques such as anomaly detection, noise filtering, and imputation can help mitigate the effects of corrupted or missing data. These capabilities are crucial in contexts like pharmaceutical formulation, where even small data errors can propagate through models and yield misleading outputs.

Missing data, in particular, is a pervasive issue. For example, when compiling datasets from literature or electronic lab notebooks, formulation components may be unreported due to inconsistent experimental reporting, varied naming conventions, or simply because they were deemed unimportant by the original authors [19-22]. In multi-ingredient dosage forms, this results in sparse datasets where one or more excipients or process parameters are missing, making it difficult to use the data for AI model training, formulation prediction, or regulatory analysis [23, 24]. Without intervention, these gaps undermine the reliability of data-driven pharmaceutical development [25] [26-29].

To address this, we explore the use of denoising autoencoders (DAEs), a class of artificial neural networks trained to reconstruct original data from intentionally corrupted inputs [30-33]. By learning internal data representations that are resilient to noise, DAEs are capable of recovering missing or distorted values, even in complex, high-dimensional datasets [34, 35]. Unlike traditional imputation methods, DAEs do not rely on simplistic assumptions like mean substitution or linear interpolation, and instead, they infer missing values based on nonlinear relationships across the dataset [35, 36].

In this study, we investigate whether DAEs can be used to "uncorrupt" incomplete pharmaceutical formulation data. Using a large-scale dataset of 1,623 formulations involving 336 unique ingredients and over half a million data points, we simulate missingness levels of up to 10% and evaluate how well various DAE architectures can recover the original data. We provide examples of how missingness typically arises in formulation data, detail the model architectures used, and assess reconstruction performance under different levels of data corruption. This work not only demonstrates

the utility of DAEs in pharmaceutical data recovery but also contributes toward the development of resilient, AI-driven frameworks for future digital drug manufacturing.

Method

Data collection, Software version and Hardware

The formulation dataset was a combination of in-house and literature-extracted data. A detailed description of the data collection, structure and labelling is provided in Refs [12, 24]. The dataset consisted of the composition of formulations and whether they were printable or not. The dataset was cleaned to remove any incomplete formulation data. In total, 1623 formulations from 336 different combinations of ingredients were used for training. The ingredients were a combination of active pharmaceutical ingredients and excipients. The values used represented the composition of ingredient used. Python (v3.12.2) was used for all programming, analysis and data plotting. The specific packages used were torch (v.2.0.0), numpy (1.24.0) and scikit-learn (v.1.3.0). An Apple M3 Max with a total of 16 CPUs, an integrated 40-core GPU and supports up to 128 GB of LPDDR5-6400 memory with up to 409.6 GB/s bandwidth.

Data Pre-processing

All composition values were scaled to between 0 and 1 using **MinMaxScaler**. To evaluate imputation performance under controlled conditions, we introduced missingness at random at three levels: 1%, 5%, and 10%. For each level of missingness, a binary mask was generated using torch.rand_like, thresholded to match the target missingness percentage, and repeated across three random seeds (42, 50, and 100) to ensure reproducibility. During training, the values at masked positions were set to zero to simulate missing data. To encourage robustness through denoising, Gaussian noise with a standard deviation of 0.1 was added to the masked inputs at each epoch. No explicit training-validation split was used, as the task is self-supervised; model performance was evaluated solely on the masked (i.e., held-out) entries.

DAE Architecture Development

The DAE was designed to reconstruct clean input data from corrupted versions by learning internal representations through layered transformations. A DAE differs from a standard AE in that the data is deliberately corrupted before being fed into the neural

network architecture [37]. We implemented two corruption mechanisms. The first involved the removal of data at fixed proportions (i.e., 1%, 5%, and 10%), with the missing values subsequently set to zero, as described in the previous section. Thereafter, the entire dataset was further corrupted by applying additive Gaussian noise with a mean of 0 and a standard deviation of 0.1. This noise was added to all input values, meaning the entire dataset was uniformly perturbed with random deviations. The noise was applied during both training and evaluation, allowing us to test the model's ability to handle not only missing values but also random noise. Hence, the DAE not only had to infer the missing values at 1%, 5%, and 10%, but also had to reconstruct clean data from inputs corrupted by additive noise. If the DAE performs well, we can infer that it has learned to generalise despite input corruption from both masking and noise.

Following corruption, the AE architecture was constructed, consisting of two main components – the encoder and the decoder. The original development of AE was to compress data for storage purposes, where the encoder compresses the input space into a lower dimensional space and when needed, using the decoder to reconstruct the latent space back to the original space. This is referred to as the undercomplete AE. Due to their prowess, the application of AE has since expanded, including for use to impute missing data. For such a task, studies have found that an overcomplete AE, whereby the latent space is larger than the input/original space, performs better. Since this is the first study of its kind in pharmaceutical formulation, we experimented with both under- and overcomplete AE architectures, as detailed below. Figure 1 depicts the difference between under- and overcomplete AE architectures, as well as a simplified schematic of the DAE process.

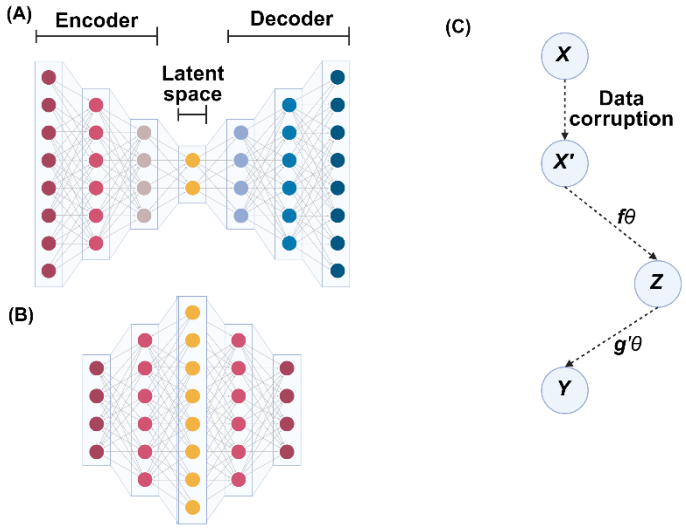

Figure 1. (A) a schematic depicting the architecture of an undercomplete AE architecture and (B) of an overcomplete architecture. (C) a schematic depicting a simplified representation of DAE. X is the original data, which is subsequently corrupted (X') and inputted into the DAE. fθ represents the encoder and g'θ represents the decoder; while z and y represent the encoding process and the output, respectively.

The AE architecture consisted of both an encoder and decoder. The encoder has two fully connected hidden layers, each followed by BatchNorm1d and LeakyReLU. The second layer defined the size of the latent space. The decoder consists of a single linear output layer with a Sigmoid activation to map values back to [0,1]. The decoder's role was to take the internal representation from the encoder and reconstruct the full input vector, including the masked values. The model was trained at set epochs using the Adam Optimizer. The loss function used was the mean squared error (MSE), which was computed only on the masked values (i.e., the deliberately removed values). As with most neural network structures, there are a myriad of designs that can be pursued. Herein, we focused and experimented on the number of neurones per hidden layer (256, 512, and 1024 neurone sizes), the Adam optimizer learning rate (10^{-1}, 10^{-3} and 10^{-5}) and the number of epochs (100, 500, 1000 and 1200). As mentioned, since the encoder's second hidden layer defined the latent space, the latent space was set to 256, 512, or 1024 dimensions in different experiments.

kNN Technique

k-nearest neighbour (kNN) is a common ML technique used for imputting missing data and was used herein to benchmark the performance of the DAE. kNN imputation filled each missing entry by locating the most similar samples using distances computed over the observed features only, then taking the mean of the corresponding neighbour

values. Prior to distance calculation, all features were scaled to the [0, 1] range. We evaluated neighbour counts and similarity settings to balance locality and smoothing, with the number of neighbours explored were 3, 5, 10, 20 and 50; weighting schemes at both uniform and distance; and distance metrics explored were euclidean, Manhattan and cosine. For each missingness level (1%, 5%, 10%), identical masking and random seeds were used as in the comparative DAE experiments.

Model Evaluation

Following training, the missing values were imputed by replacing the masked entries with the model's predictions, and performance was evaluated only on these reconstructed values. The metric used to assess imputation quality was the coefficient of determination (R^2), which was calculated over the masked positions. Each experiment was repeated across the three random seeds to account for variability introduced by different missingness patterns. The final results were reported as the mean and standard deviation across these runs.

Results

Exploratory Data Analysis (EDA)

The dataset was originally developed for predicting formulation printability using machine learning techniques. It comprised over 545,000 individual data points, of which only approximately 1% were non-zero values, with the remaining 99% being zeros (Figure 2 (A)). This high level of sparsity reflects the way the data was structured: each formulation is represented across 336 possible ingredient slots, and if an ingredient is not used in a formulation, its value is recorded as zero. Since most formulations contain only 1 to 7 ingredients out of the 336, the vast majority of entries across the dataset are necessarily zero. As a result, the formulation composition data is inherently and significantly sparse.

Among the non-zero values, there were 296 distinct composition entries, though their distribution was notably uneven. As shown in Figure 2 (B), ingredient concentrations at or below 20% w/w were far more commonly recorded than higher concentrations, such as those exceeding 60% w/w. Figure 2 (C) presents the distribution of non-zero values across the 336 ingredient slots. It can be seen that over 250 ingredients were

recorded in less than 1% of all formulations, while only a single ingredient appeared in more than 75% of cases, which was identified as a lubricant.

These observations highlight three key characteristics of pharmaceutical formulation data. These are (i) a significant imbalance between non-zero and zero entries, (ii) a skewed distribution of non-zero values used, and (iii) a high proportion of ingredients that are rarely used. Together, these traits present challenges for ML techniques, which typically perform best with balanced datasets. Without such balance, models tend to overfit to the most frequent patterns, which in this case, predicting zeros or common concentration values (i.e., < 20 w/w%). As a result, the model has the challenge of not only detecting relatively rare non-zero value entries but also predict the correct value from among 296 possibilities, many of which occur infrequently.

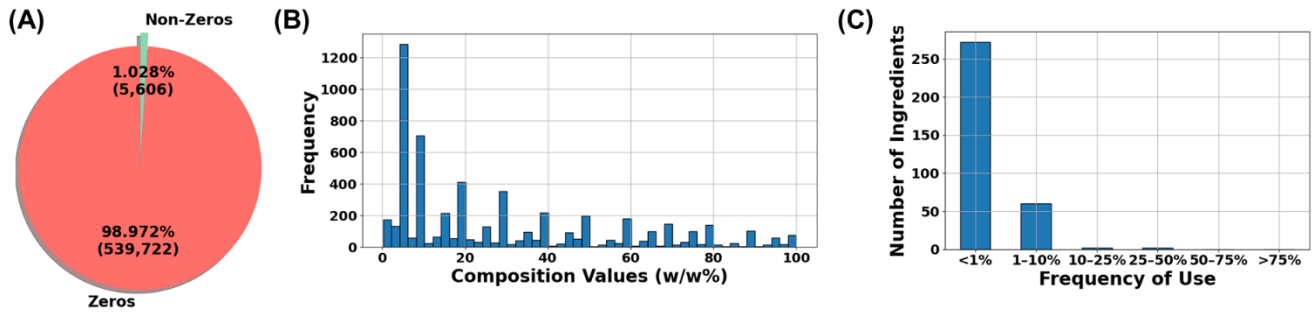

Figure 2. Exploratory data analysis depicting the characteristics of the pharmaceutical formulation dataset. (A) ratio of non-zeros to zeros. (B) the frequency of composition values in w/w% used. (c) the frequence of non-zero values per ingredient.

ML Analysis

In this study, we explored the use of kNN and DAE, a traditional ML method and deep learning method respectively, in imputing 1%, 5% and 10% of the data. These correspond to over 5,450, 27,250 and 54,500 missing values, respectively. The missing data points were randomly removed from the raw dataset and hidden from both the kNN and DAE before model training. In this first study on imputing pharmaceutical 3D printing data, we evaluated multiple configurations of kNN and DAE.

kNN Analysis

The kNN analysis was investigated at different neighbour settings, which is the most common parameter for the ML techniques. Furthermore, we explored different distance settings and weightings thereof. The analysis revealed that kNN was

unsuccessful in imputing the missing values, across all missing amounts. For when 1% of the data was removed, the kNN resulted in negative R^2 values, which indicate that the technique performed worse than random guessing. As portrayed in Figure 3, increasing the neighbours from 3 to 50 resulted in the R^2 converging to -0.3. For both 5% and 10% missingness, kNN resulted in negative R^2 values that increased with the number of neighbours indicating slight improvements in fit, though performance remained below that of a naive mean imputation (Figure S1). To further probe this behaviour, we extended the number of neighbours up to 400 and confirmed that R^2 did not improve beyond the previously observed plateau (Figure S2).

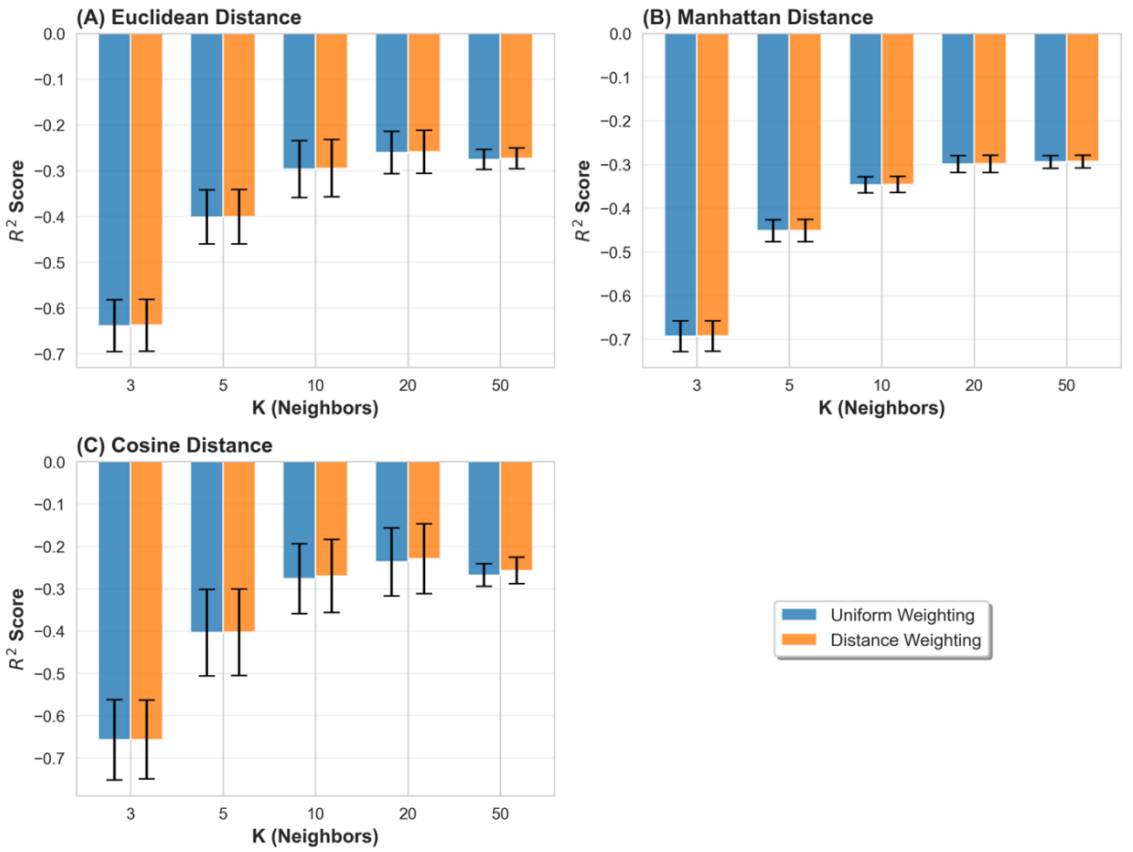

Figure 3. The imputation results for kNN at 1% missing data, using (A) Euclidean, (B) Manhattan and (C) Cosine distances.

Further examination of the predictions revealed that the machine learning technique tended to overpredict zero values and underpredict non-zero values (Figure 4). Notably, accurate predictions were more common at lower target values, which aligns with the higher frequency of these values observed in Figure 2 (B).

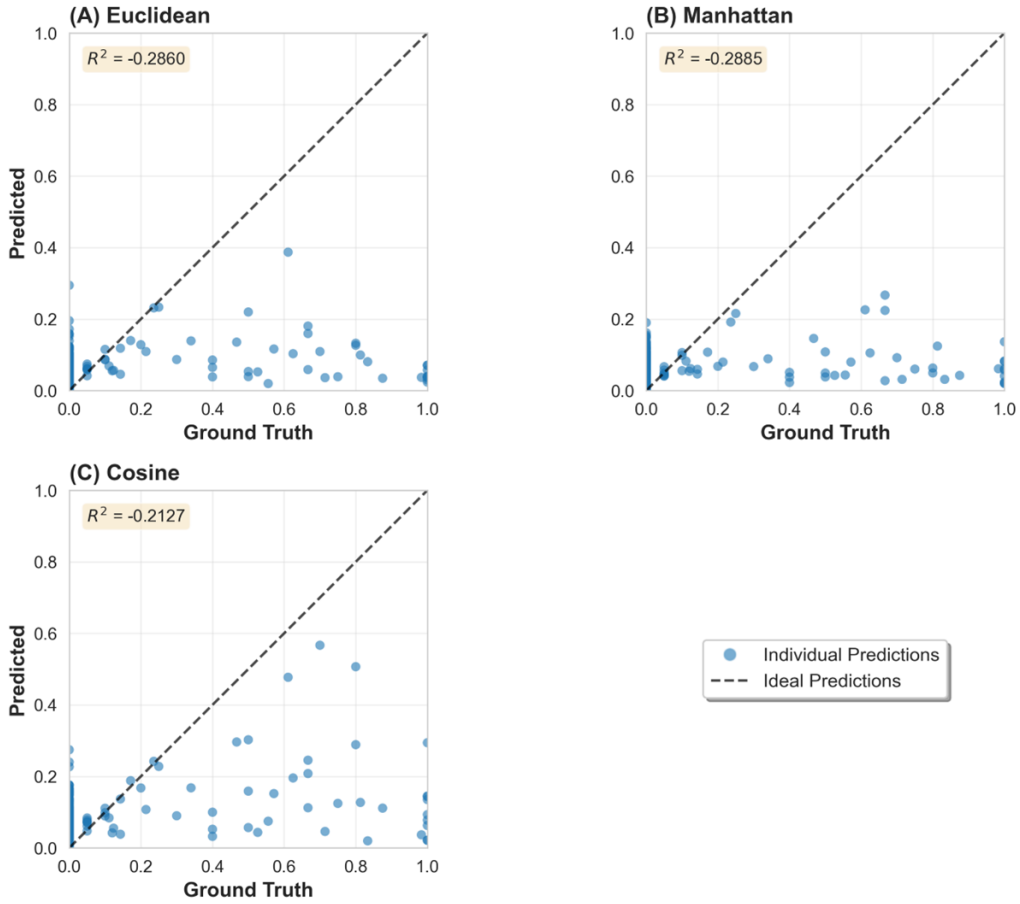

Figure 4. The kNN ground truth vs predicted plots for 1% missing data using (A) Euclidean, (B) Manhattan and (C) Cosine distances. The results for 5% and 10% can be found in the supplementary (Figure S3). The scale ranges from 0.00 to 1.00, as the composition dataset was normalised from 0–100 w/w% to [0, 1] prior to model training (e.g., 20 w/w% is represented as 0.2).

DAE Analysis

DAEs are neural networks and learn by going through the dataset iteratively during training, with each complete pass through the dataset is referred to as an epoch. During each epoch, the model updates its internal weights to reduce the difference between its predictions and the actual data – in other words the neural network tries to minimise the error after each epoch cycle. This is measured by the loss function, which herein was the MSE. In tracking the loss value over successive epochs, we can analyse how well the model is learning. These loss curves provide insight into model convergence, overfitting, and overall performance stability, key indicators when evaluating the effectiveness of different training configurations.

The resulting loss curves for DAE applied to 1% missing data, grouped by learning rate, are shown in Figure 5. In all cases, the loss decreased as training progressed, indicating that model performance improved with additional epochs. For higher

learning rates (10^{-1} and 10^{-3}), the loss dropped rapidly, with the 10^{-1} rate exhibiting a sharp decline, suggesting unstable or potentially catastrophic learning during the initial epochs. In contrast, the lower learning rate (10^{-5}) produced a more gradual decrease in loss over the course of 1200 epochs. Notably, increasing the number of neurons in the hidden layer led to a more pronounced reduction in loss, particularly in the slower learning rate. These observations collectively suggest that training over 1200 epochs was sufficient for the DAEs to converge, allowing the models to effectively learn the underlying structure of the data. The same loss curve behaviours were also observed for when 5% and 10% of the data was removed (Figure S4).

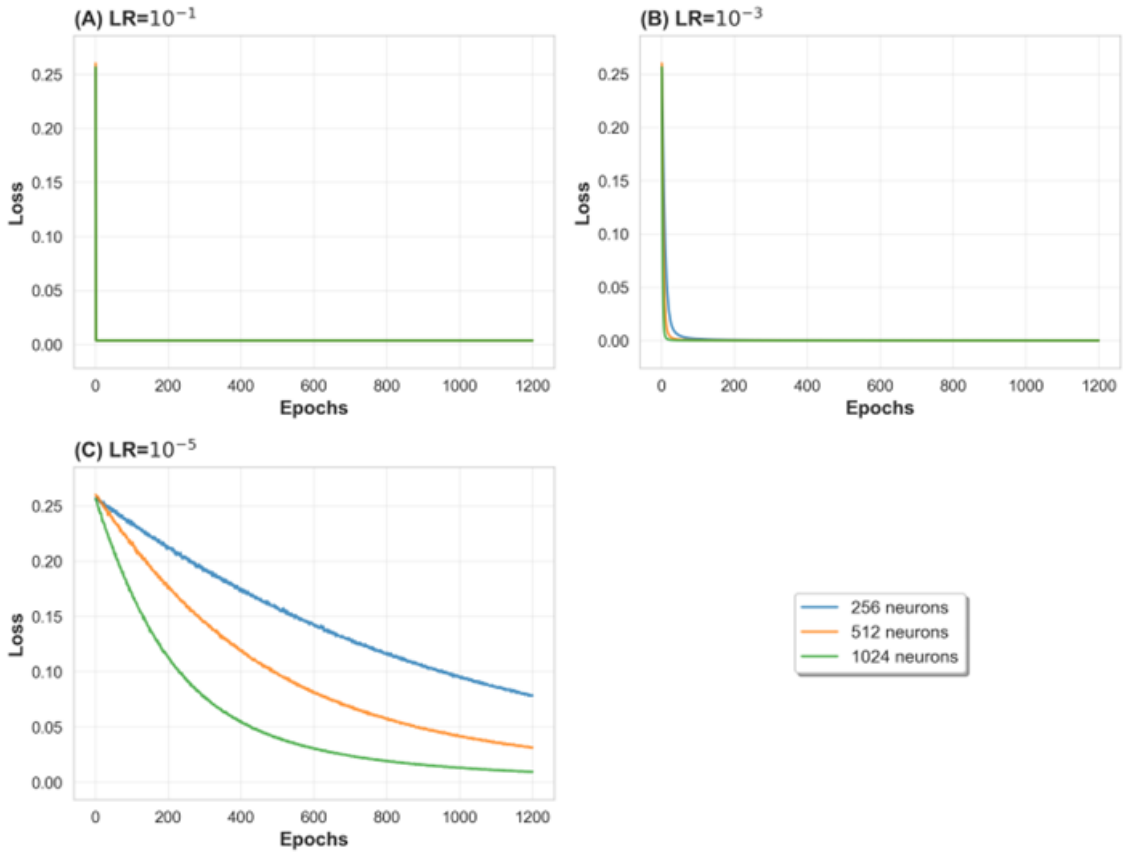

Figure 5. Loss curves for the 1% missingness analysis at learning rates of (A) 10^{-1}, (B) 10^{-3} and (C) and 10^{-5}.

Following training, the DAEs were evaluated on their ability to predict missing values (Figure 6 - Figure 8). When the learning rate was set to either 10^{-1} or 10^{-5}, the models performed poorly, with the majority of R^2 scores below zero. However, with a learning rate of 10^{-3}, the models displayed a clear improvement, producing positive R^2 scores. This suggests that the network was able to learn a meaningful relationship from the

data at this learning rate. Further observations revealed that both the number of neurones in the hidden layer and the number of training epochs influenced the results. When 1% of the data was removed, the best performance had an R^2 of 0.989 ± 0.0017, which was achieved using 1024 neurones and 1200 epochs (Figure 6 (B)). Interestingly, increasing the number of neurones beyond a certain point did not lead to significantly better results, as comparable performance was observed at the same number of epochs with fewer neurons. Overall, the learning rate had the greatest impact on performance when 1% of the data was removed. While increasing the number of epochs and adjusting the network size provided some benefit, these factors were second to choosing an appropriate learning rate.

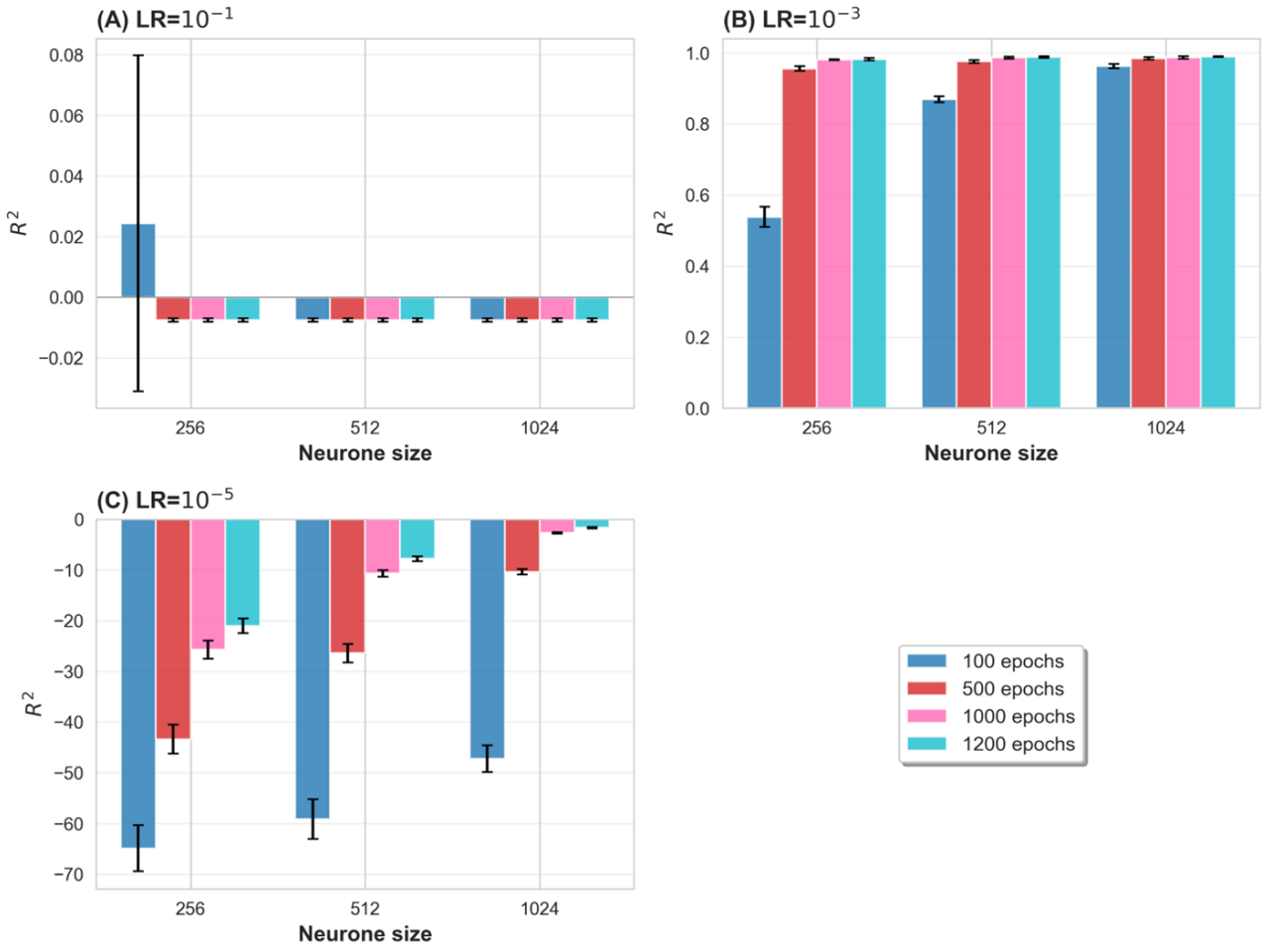

Figure 6. Performance of DAE architectures under 1% missing data at learning rates of (A) 10^{-1}, (B) 10^{-3} and (C) and 10^{-5}. Each bar represents the mean R^2 across three repeats for a given neurone size, with error bars indicating standard deviation. Note that the y-axis scales differ across subplots to accommodate the wide range of model performances.

Similar patterns were observed when the proportion of missing data was increased to 5% and 10%, particularly for the learning rates of 10^{-5} and 10^{-1}. In both cases, the R^2 scores remained negative, again indicating poor performance. However, when the learning rate was set to 10^{-3}, the results were predominantly positive (Figure 7 (B) and Figure 8 (B)). At 5% and 10% missing data, the highest R^2 achieved was 0.983 ± 0.0031 and 0.976 ± 0.0007, respectively. Hence, across all levels of missing data, it was evident that the learning rate had the strongest effect on DAE performance.

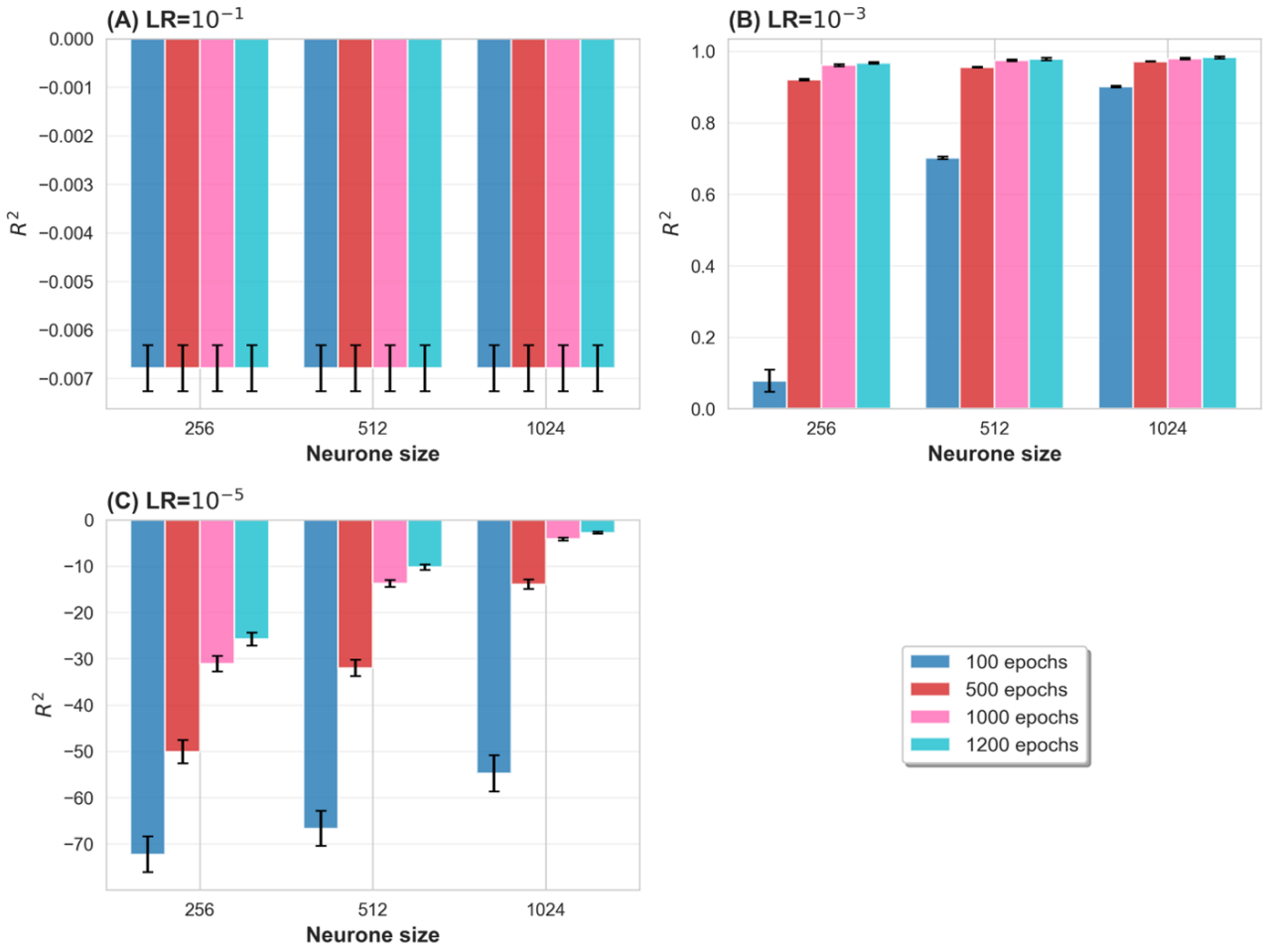

Figure 7. Performance of DAE architectures under 5% missing data at learning rates of (A) 10^{-1}, (B) 10^{-3} and (C) and 10^{-5}. Each bar represents the mean R^2 across three repeats for a given neurone size, with error bars indicating standard deviation. Note that the y-axis scales differ across subplots to accommodate the wide range of model performances.

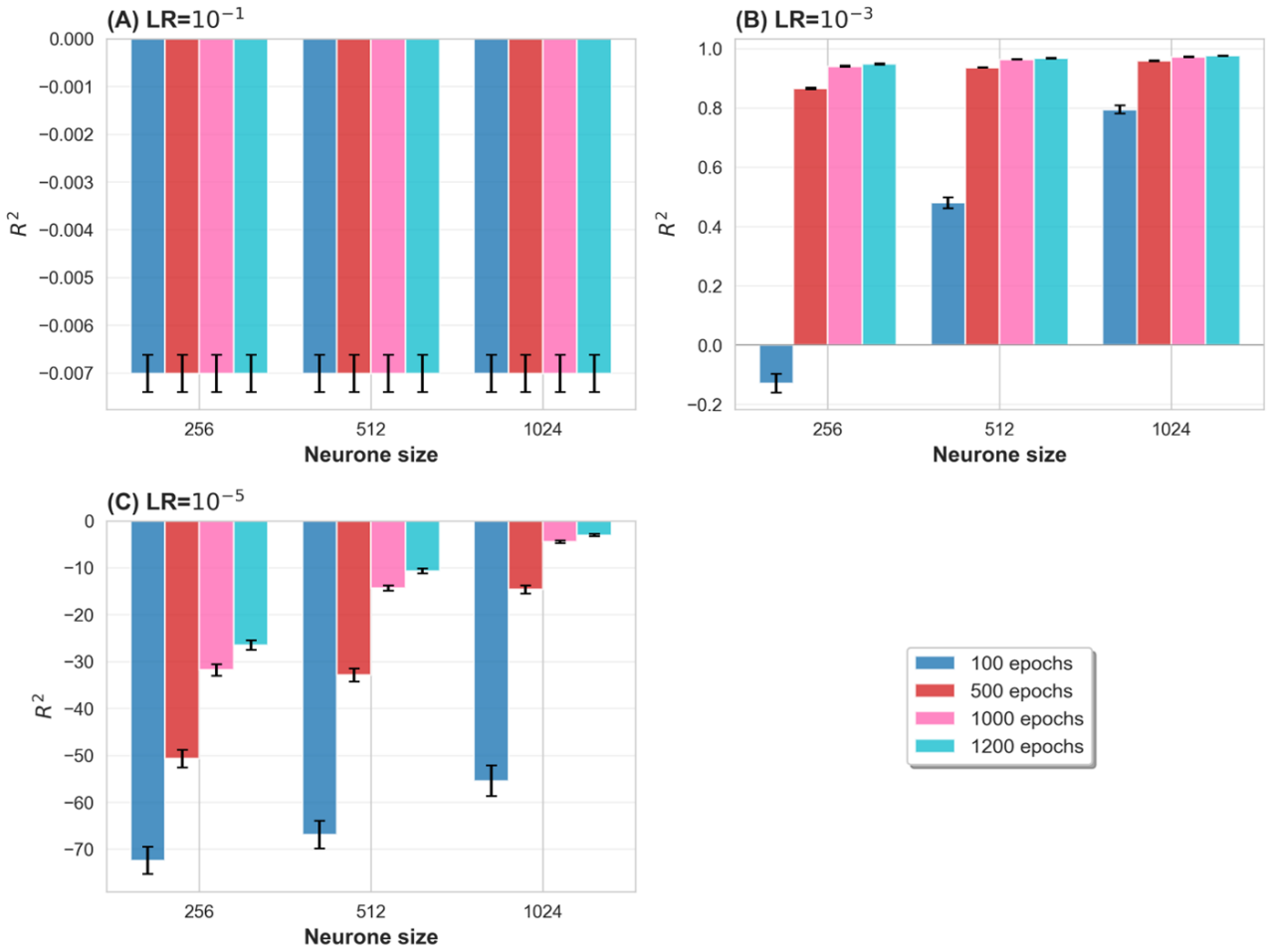

Figure 8. Performance of DAE architectures under 10% missing data at learning rates of (A) 10^{-1}, (B) 10^{-3} and (C) and 10^{-5}. Each bar represents the mean R^2 across three repeats for a given neurone size, with error bars indicating standard deviation. Note that the y-axis scales differ across subplots to accommodate the wide range of model performances.

To better understand how the model was performing, we compared the predicted values to the actual (ground truth) values in the case where 1% of the data was removed (Figure 9). At the high learning rate of 10^{-1}, the DAE consistently predicted all outputs as zero. This explains the consistent and poor R^2 values at this learning rate, since the model was not learning anything useful, just defaulting to zero. This was expected since 99% of the data contained zeros and ML techniques have a tendency to overfit to the dominant value. With the slower learning rate of 10^{-5}, a different issue appeared. The model made more of an attempt to predict non-zero values, but the predictions were often far off from the actual values. Instead of converging on useful patterns, the model seemed to struggle. This explains why architectures at this learning rate had the lowest R^2 scores across all three learning rates. At the intermediate and optimal learning rate of 10^{-3}, it could correctly predict both zero and non-zero values. Unlike the kNN results (Figure 4), no bias was seen

towards the more dominant values observed in Figure 2 (B). This behaviour demonstrates that DAEs, when properly tuned, are capable of learning meaningful imputations, representing a promising foundation for applying DAEs to the problem of imputing large-scale missing formulation data.

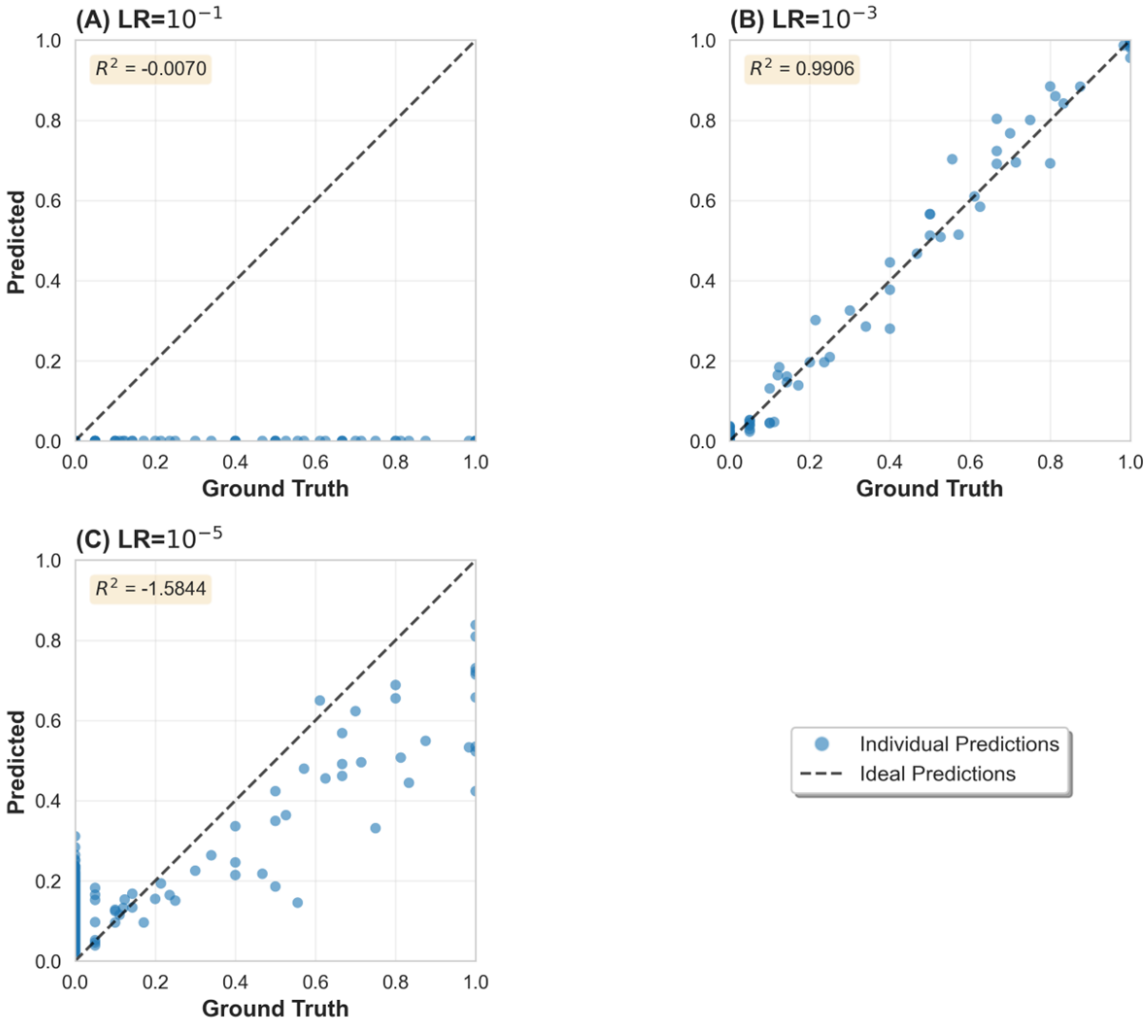

Figure 9. Example of the predicted vs ground truth results for the 1% missing data analysis at learning rates of (A) 10^{-1}, (B) 10^{-3} and (C) and 10^{-5}. Note the scale is from 0.00 to 1.00 because the composition dataset was normalised from 0-100 w/w% to between 0-1 prior to model training. The scale ranges from 0.00 to 1.00, as the composition dataset was normalised from 0–100 w/w% to [0, 1] prior to model training (e.g., 20 w/w% is represented as 0.2).

The analysis clearly shows that DAE substantially outperformed kNN imputation in terms of R^2. Interestingly, despite yielding better predictive performance, DAE was also significantly faster than kNN across all levels of missingness (Figure 10). For datasets with 1% and 5% missing values, DAE was approximately two orders of magnitude faster, and at 10% missingness, it was three orders of magnitude faster. A common imputation approach used in data science, albeit far from realistic and potentially damaging in pharmaceutics, is to impute a value with zero. As depicted in

Figure 10, simply imputing a missing value with a zero is marginally faster than DAE. Nevertheless, DAEs can impute missing values at 10^{-3} seconds, which is fast and should not delay the 3D printing process of medicines.

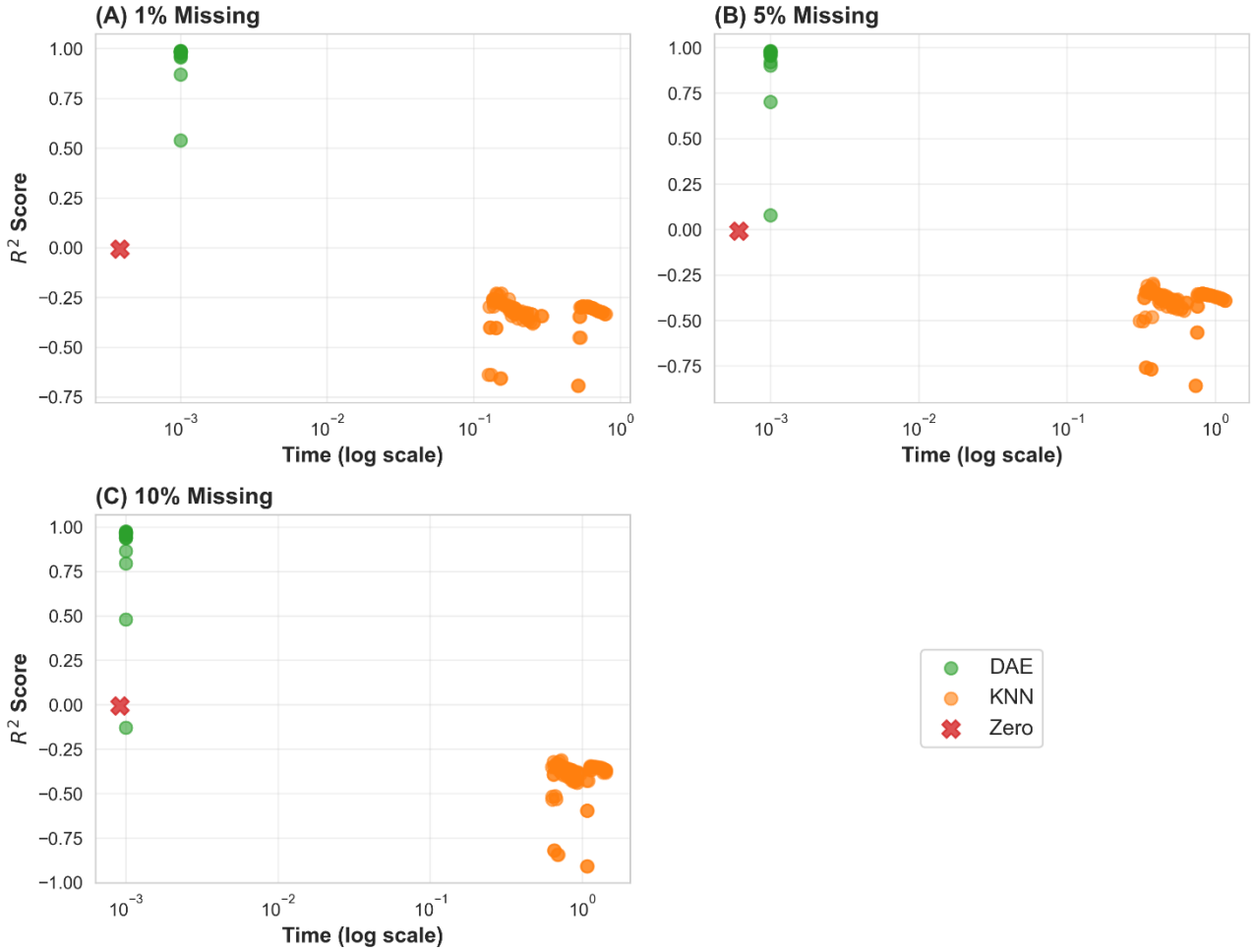

Figure 10. The computational time (in seconds) vs R^2 plot for (A) 1%, (B) 5% and (C) 10% missing data. While considerably outperforming kNN, DAEs with a learning rate of 10^{-3} also required less time for imputing missing values. "Zero" refers to a baseline method where missing values were imputed with zero.

Discussion

For the first time, we have demonstrated that an ML approach like DAE has the capacity to accurately reconstruct corrupt pharmaceutical formulation data, despite the challenging characteristics of the data. Given the optimal learning rate, which was 10^{-3}, DAE can effectively impute missing data up to 10%, which herein was over 54,500 datapoints. Detecting this level of missing data is nearly impossible by a trained expert, and the fact that DAE can correct it in under 1 second demonstrates its necessity for such applications. Previous work has demonstrated that a derivative of DAEs can be

effective at larger percentage of corrupted data [38, 39], suggesting that the technique holds strong potential for handling more substantial errors or cyber-attacks in pharmaceutical formulation data. Although root mean squared error and mean absolute error were recommended for DAE training in data imputation [36], they were not well-suited to the characteristics of this dataset. In future studies, we will consider metrics such as the R^2 or mean absolute percentage error, which may offer more meaningful insights into model performance for formulation dataset.

Future work could also explore the use of DAEs beyond correcting corrupt data. The ability of DAEs to accurately reconstruct missing values also suggests that these models are capturing the underlying structure of pharmaceutical formulations. In generative ML it has already been shown that models such as conditional generative adversarial networks [40]. We propose that imputation performance could serve as a useful early indication for generative capability. Specifically, if a model can reliably infer missing data, it implies that it has learned a meaningful representation of formulation space. This, in turn, could be leveraged to generate new formulations by sampling or modifying latent representations. However, in generative applications, a high R^2 is not necessarily desirable. A model that perfectly reconstructs its training data may simply reproduce known formulations rather than create new ones. Therefore, future work should aim to define an optimal R^2 range, whereby the score is high enough to suggest the model understands the structure of valid formulations, but not too high that it merely memorises the input. This trade-off will be essential for developing models capable of true innovation in pharmaceutical design.

Conclusion

Corrupted data, including missing data, is a concern in the field of pharmaceutics. This study investigated the use of DAE for imputing missing formulation data pertaining to pharmaceutical 3D printing. The state-of-the-art ML technique is DAE, which was used to assess its capability of imputing 1%, 5% and 10% missing data. As this was the first study to investigate the use of DAEs for missing data, a range of neural network architectures were explored, with parameters experimented with include size of neurones, the learning rate and number of epochs. The analysis revealed that DAE has the capacity to impute missing formulation data, achieving a maximum R^2 above 0.9 irrespective of the amount of data removed, considerably outperforming the

traditional kNN imputation method. Furthermore, the effect of learning rate was found to impact DAE performance, with the optimal rate being 10^{-3}. Therefore, it was concluded that DAE have a potential to address corrupted data in the pharmaceutical formulation development. It is envisaged that autoencoders could be extended beyond missing data imputation to address a broader range of data quality issues, thereby supporting the development and maintenance of high-quality datasets.

References

[1] X. Zhu, H. Li, L. Huang, M. Zhang, W. Fan, L. Cui, 3D printing promotes the development of drugs, Biomedicine & Pharmacotherapy, 131 (2020) 110644.
[2] A.A. Mohammed, M.S. Algahtani, M.Z. Ahmad, J. Ahmad, S. Kotta, 3D Printing in medicine: Technology overview and drug delivery applications, Annals of 3D Printed Medicine, 4 (2021) 100037.
[3] D. Muhindo, R. Elkanayati, P. Srinivasan, M.A. Repka, E.A. Ashour, Recent Advances in the Applications of Additive Manufacturing (3D Printing) in Drug Delivery: A Comprehensive Review, AAPS PharmSciTech, 24 (2023) 57.
[4] J. Wang, Y. Zhang, N.H. Aghda, A.R. Pillai, R. Thakkar, A. Nokhodchi, M. Maniruzzaman, Emerging 3D printing technologies for drug delivery devices: Current status and future perspective, Advanced Drug Delivery Reviews, 174 (2021) 294-316.
[5] M.E. Alkahtani, S. Sun, C.A.R. Chapman, S. Gaisford, M. Orlu, M. Elbadawi, A.W. Basit, 3D printed electro-responsive system with programmable drug release, Materials Today Advances, 23 (2024) 100509.
[6] M. Elbadawi, H. Li, P. Ghosh, M.E. Alkahtani, B. Lu, A.W. Basit, S. Gaisford, Cold Laser Sintering of Medicines: Toward Carbon Neutral Pharmaceutical Printing, ACS Sustainable Chemistry & Engineering, 12 (2024) 11155-11166.
[7] X.W. Kok, A. Singh, B.T. Raimi-Abraham, A Design Approach to Optimise Secure Remote Three-Dimensional (3D) Printing: A Proof-of-Concept Study towards Advancement in Telemedicine, Healthcare, 10 (2022) 1114.
[8] J.J. Relinque, E.M. Campos, M. León-Calero, L. Rodríguez-Rodríguez, M. Nieto-Diaz, I. Novillo-Algaba, K. Artola, R.G. Fernández, J. Mingorance, I. García, J. Rodríguez-Hernández, Fabrication of 3D printed swabs in University Hospital's: Point of care manufacturing, study of mechanical properties and biological compatibility, Polymer, 323 (2025) 128162.
[9] S. Ayyoubi, L. Ruijgrok, H. van der Kuy, R. ten Ham, F. Thielen, What Does Pharmaceutical 3D Printing Cost? A Framework and Case Study with Hydrocortisone for Adrenal Insufficiency, PharmacoEconomics - Open, 9 (2025) 207-215.
[10] Y. Abdalla, M. Ferianc, A. Awad, J. Kim, M. Elbadawi, A.W. Basit, M. Orlu, M. Rodrigues, Smart laser Sintering: Deep Learning-Powered powder bed fusion 3D printing in precision medicine, International Journal of Pharmaceutics, 661 (2024) 124440.
[11] S. Sun, M.E. Alkahtani, S. Gaisford, A.W. Basit, M. Elbadawi, M. Orlu, Virtually Possible: Enhancing Quality Control of 3D-Printed Medicines with Machine Vision Trained on Photorealistic Images, Pharmaceutics, 15 (2023) 2630.

[12] M. Elbadawi, B.M. Castro, F.K. Gavins, J.J. Ong, S. Gaisford, G. Pérez, A.W. Basit, P. Cabalar, A. Goyanes, M3DISEEN: A novel machine learning approach for predicting the 3D printability of medicines, International Journal of Pharmaceutics, 590 (2020) 119837.
[13] S.J. Trenfield, A. Awad, L.E. McCoubrey, M. Elbadawi, A. Goyanes, S. Gaisford, A.W. Basit, Advancing pharmacy and healthcare with virtual digital technologies, Advanced Drug Delivery Reviews, 182 (2022) 114098.
[14] M. Elbadawi, L.E. McCoubrey, F.K.H. Gavins, J.J. Ong, A. Goyanes, S. Gaisford, A.W. Basit, Harnessing artificial intelligence for the next generation of 3D printed medicines, Advanced Drug Delivery Reviews, 175 (2021) 113805.
[15] Y. Gao, B. Li, W. Wang, W. Xu, C. Zhou, Z. Jin, Watching and Safeguarding Your 3D Printer: Online Process Monitoring Against Cyber-Physical Attacks, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2 (2018) Article 108.
[16] H. Pearce, K. Yanamandra, N. Gupta, R. Karri, FLAW3D: A Trojan-Based Cyber Attack on the Physical Outcomes of Additive Manufacturing, IEEE/ASME Transactions on Mechatronics, 27 (2022) 5361-5370.
[17] E.C. Balta, M. Pease, J. Moyne, K. Barton, D.M. Tilbury, Digital Twin-Based Cyber-Attack Detection Framework for Cyber-Physical Manufacturing Systems, IEEE Transactions on Automation Science and Engineering, 21 (2024) 1695-1712.
[18] H.M. Lisboa, A. Lúcio, R. Andrade, A.M. Sarinho, J. Lima, L. Batista, M.E. Costa, A. Nascimento, M.B. Pasquali, Leveraging quantitative structure-activity relationships to design and optimize wall material formulations for antioxidant encapsulation via spray drying, Food Chemistry Advances, 7 (2025) 100948.
[19] J. Zaslavsky, C. Allen, A dataset of formulation compositions for self-emulsifying drug delivery systems, Scientific Data, 10 (2023) 914.
[20] L.E. McCoubrey, M. Elbadawi, M. Orlu, S. Gaisford, A.W. Basit, Harnessing machine learning for development of microbiome therapeutics, Gut Microbes, 13 (2021) 1872323.
[21] F. Wang, M. Elbadawi, S. Liu Tsilova, S. Gaisford, A.W. Basit, M. Parhizkar, Machine Learning Predicts Electrospray Particle Size, Materials & Design, (2022) 110735.
[22] F. Wang, N. Sangfuang, L.E. McCoubrey, V. Yadav, M. Elbadawi, M. Orlu, S. Gaisford, A.W. Basit, Advancing oral delivery of biologics: Machine learning predicts peptide stability in the gastrointestinal tract, International Journal of Pharmaceutics, 634 (2023) 122643.
[23] Y. Abdalla, M. Elbadawi, M. Ji, M. Alkahtani, A. Awad, M. Orlu, S. Gaisford, A.W. Basit, Machine learning using multi-modal data predicts the production of selective laser sintered 3D printed drug products, International Journal of Pharmaceutics, 633 (2023) 122628.
[24] B.M. Castro, M. Elbadawi, J.J. Ong, T. Pollard, Z. Song, S. Gaisford, G. Pérez, A.W. Basit, P. Cabalar, A. Goyanes, Machine learning predicts 3D printing performance of over 900 drug delivery systems, Journal of Controlled Release, 337 (2021) 530-545.
[25] T. Duflot, L. Fayette, C. Konecki, J. Seurat, C. Feliu, J. Scala-Bertola, Z. Djerada, Assessing the Impact of Multiple Imputation Algorithms on Pharmacokinetic Model Performance: A Simulation-Based Study, The AAPS Journal, 27 (2025) 77.
[26] P. Tormay, Big Data in Pharmaceutical R&D: Creating a Sustainable R&D Engine, Pharmaceutical Medicine, 29 (2015) 87-92.

[27] G. Hole, A.S. Hole, I. McFalone-Shaw, Digitalization in pharmaceutical industry: What to focus on under the digital implementation process?, International Journal of Pharmaceutics: X, 3 (2021) 100095.
[28] P. Yadav, Digital Transformation in the Health Product Supply Chain: A Framework for Analysis, Health Systems & Reform, 10 (2024) 2386041.
[29] R. Mitra, S.F. McGough, T. Chakraborti, C. Holmes, R. Copping, N. Hagenbuch, S. Biedermann, J. Noonan, B. Lehmann, A. Shenvi, X.V. Doan, D. Leslie, G. Bianconi, R. Sanchez-Garcia, A. Davies, M. Mackintosh, E.-R. Andrinopoulou, A. Basiri, C. Harbron, B.D. MacArthur, Learning from data with structured missingness, Nature Machine Intelligence, 5 (2023) 13-23.
[30] A. Janssen, F.C. Bennis, R.A.A. Mathôt, Adoption of Machine Learning in Pharmacometrics: An Overview of Recent Implementations and Their Considerations, Pharmaceutics, 14 (2022) 1814.
[31] B. Yang, Y. Chung, A.Y. Yang, B. Yuan, T. Chen, X. Yu, QComp: A QSAR-Based Imputation Framework for Drug Discovery, Journal of Chemical Information and Modeling, 65 (2025) 7862-7873.
[32] H. Iwata, T. Matsuo, H. Mamada, T. Motomura, M. Matsushita, T. Fujiwara, K. Maeda, K. Handa, Predicting Total Drug Clearance and Volumes of Distribution Using the Machine Learning-Mediated Multimodal Method through the Imputation of Various Nonclinical Data, Journal of Chemical Information and Modeling, 62 (2022) 4057-4065.
[33] Y. Zhou, S. Aryal, M.R. Bouadjenek, Review for Handling Missing Data with special missing mechanism, arXiv preprint arXiv:2404.04905, (2024).
[34] R.C. Pereira, P.H. Abreu, P.P. Rodrigues, Partial Multiple Imputation With Variational Autoencoders: Tackling Not at Randomness in Healthcare Data, IEEE Journal of Biomedical and Health Informatics, 26 (2022) 4218-4227.
[35] I. Gjorshoska, T. Eftimov, D. Trajanov, Missing value imputation in food composition data with denoising autoencoders, Journal of Food Composition and Analysis, 112 (2022) 104638.
[36] R.C. Pereira, M.S. Santos, P.P. Rodrigues, P.H. Abreu, Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes, Journal of Artificial Intelligence Research, 69 (2020) 1255-1285.
[37] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096-1103.
[38] N. Abiri, B. Linse, P. Edén, M. Ohlsson, Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems, Neurocomputing, 365 (2019) 137-146.
[39] A.F. Costa, M.S. Santos, J.P. Soares, P.H. Abreu, Missing Data Imputation via Denoising Autoencoders: The Untold Story, in, Springer International Publishing, Cham, 2018, pp. 87-98.
[40] M. Elbadawi, H. Li, S. Sun, M.E. Alkahtani, A.W. Basit, S. Gaisford, Artificial intelligence generates novel 3D printing formulations, Applied Materials Today, 36 (2024) 102061.