

Article

Two-Phase Flow Pattern Identification in Vertical Pipes Using Transformer Neural Networks

Carlos Mauricio Ruiz-Díaz ¹, Erwing Eduardo Perilla-Plata ² and Octavio Andrés González-Estrada ^{2,*} 

¹ Industrial Multiphase Flow Laboratory (LEMI), São Carlos School of Engineering, University of São Paulo, São Carlos 13563-120, Brazil; carlosruiz978@usp.br

² Research Group in Energy and Environment (GIEMA), Universidad Industrial de Santander, Santander, Bucaramanga 680002, Colombia; plataperillaodraude@gmail.com

* Correspondence: agonzale@uis.edu.co

Abstract: The oil and gas industry consistently embraces innovative technologies due to the significant expenses associated with hydrocarbon transportation, pipeline corrosion issues, and the necessity to gain a deeper understanding of two-phase flow characteristics. These solutions involve the implementation of predictive models utilizing neural networks. In this research paper, a comprehensive database comprising 4864 data points, encompassing information pertaining to oil–water two-phase flow properties within vertical pipelines, was meticulously curated. Subsequently, an encoder-only type transformer neural network (TNN) was employed to identify two-phase flow patterns. Various configurations for the TNN model were proposed, involving parameter adjustments such as the number of attention heads, activation function, dropout rate, and learning rate, with the aim of selecting the configuration yielding optimal outcomes. Following the training of the network, predictions were generated using a reserved dataset, thus facilitating the creation of flow maps depicting the patterns anticipated by the model. The developed TNN model successfully predicted 9 out of the 10 flow patterns present in the database, achieving a peak accuracy of 53.07%. Furthermore, the various predicted flow patterns exhibited an average precision of 63.21% and an average accuracy of 86.51%.

Keywords: transformer neural network; two-phase flow; flow pattern identification; machine learning; fluid transport



Citation: Ruiz-Díaz, C.M.; Perilla-Plata, E.E.; González-Estrada, O.A. Two-Phase Flow Pattern Identification in Vertical Pipes Using Transformer Neural Networks. *Inventions* **2024**, *9*, 15. <https://doi.org/10.3390/inventions9010015>

Academic Editors: Haibao Hu, Xiaopeng Chen and Peng Du

Received: 26 November 2023

Revised: 10 January 2024

Accepted: 12 January 2024

Published: 18 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The transportation of multiphase flows is a commonplace practice in today's industrial landscape. These flows typically consist of two or more phases and may include various physical states, such as liquid, solid, or gaseous components. Multiphase flows find applications across diverse industrial sectors, including agriculture, food production, pharmaceuticals, and oil extraction, among others. To categorize multiphase flows effectively, they are classified based on the specific phases involved. Three-phase flows, for instance, encompass gas, oil, and solid substances. However, the primary focus of study within multiphase flows often centers on two-phase flows, which can manifest as gas–liquid, gas–solid, liquid–solid, or liquid–liquid configurations [1,2]. The subcategory of two-phase liquid–liquid flows is particularly relevant in the oil and gas industry. It pertains to the production of crude oil from water and oil mixtures within pipelines. These mixtures are then transported to processing facilities or refineries for further treatment and refinement [3–5].

An essential aspect of multiphase flows is their internal geometric configuration or flow pattern. This configuration arises from the interactions between the different substances involved and is contingent on various factors, including the flow rates of these substances, the diameter of the pipe, and the transport properties unique to each of the flows [1]. A crucial variable in determining the type of flow pattern within a pipe is its

inclination. This is due to the fact that the flow patterns observed in horizontal pipes differ from those encountered in vertical pipes [6]. Investigating the behavior of two-phase flow patterns in pipelines is of utmost importance, particularly in addressing challenges like the reduction of pumping costs associated with the extraction of heavy crude oils, which often possess viscosities exceeding 10,000 cP [7], given that these make up a third of the world's hydrocarbon reserves [8]. Furthermore, the oil industry faces a significant challenge related to pipeline corrosion, both internally and externally. Studies have estimated that around 80% of pipeline failures in the oil transportation sector are attributed to internal corrosion. These corrosion-related failures result in substantial maintenance costs. Therefore, it becomes imperative to accurately determine the flow pattern within the pipeline as a means to mitigate these corrosion-related failures and reduce maintenance expenses [9].

Furthermore, in reference [10], predictive models are explored for calculating the volumetric fraction of a horizontal two-phase water–oil flow using an artificial neural network. The results indicate a mean absolute percentage error (MAPE) of 3.01% and a coefficient of determination of 0. Additionally, in reference [11], it is mentioned that utilizing Computational Fluid Dynamics (CFD) predictions of a two-phase flow with STAR-CCM+ software (Version 8.06) enables precise control of the corrosion risk in pipelines. Moreover, simulation has yielded significant contributions to gas lift processes based on the two-fluid model [12], as well as the capture and recovery of CO₂ with the formation of gas combustion hydrates in both frozen and unfrozen sediments [13]. Furthermore, it enables the development of two-dimensional models to simulate asphaltene deposition in wells and pipelines [14] and optimization in determining the minimum miscibility pressure between CO₂ and petroleum using hybrid methods [15]. Additionally, thanks to the progress brought about by the fourth industrial revolution (Industry 4.0), benefits have been implemented in various fields with the assistance of artificial intelligence (AI), which has the potential to process vast amounts of data [16]. The scope of application of this learning technique extends to various industrial sectors, as is the case with the oil and gas sector. Therefore, innovative methods are required to replace the old monitoring methods in hydrocarbon transport, where the presence of multiphase flows in different combinations of liquids, gases, and solids is recorded. Consequently, the simulation of processes involving multiphase flows has allowed for the generation of solutions to various engineering problems through modeling. This extends to a wide range of applications that ensure the proper flow of hydrocarbons inside pipelines [17–20].

The extensive domain of machine learning has generated various alternatives to address complex problems associated with data recognition and classification [20]. Among the implemented techniques is the Transformer Neural Network (TNN), a neural network originally created for translating long strings of text [21]. It has also demonstrated high performance in multilingual models [22]. Additionally, leveraging the BERT architecture, derived from the original TNN, it has been applied in a variety of applications, such as text analysis for text classification [23] or emotion classification [24]. Tasks traditionally handled by convolutional neural networks, such as image detection and classification, have been adapted to TNN in a model known as the Visual Transformer [25]. Moreover, in DNA research, TNN-based models have been developed for genome sequence identification [26,27], as well as protein sequence profile prediction [28]. Furthermore, this technique has been implemented in engineering for wind energy prediction [29,30]. Additionally, a fundamental aspect of using neural networks is the allocation of 80% of the database for model training, 10% for testing, and 10% for validation [31].

Given the increasing implementation of AI in the petrochemical industry for controlling multiphase fluid-related processes, this study develops an intelligent model based on a TNN capable of predicting flow patterns within vertical pipelines with liquid–liquid two-phase flow of oil and water. First, the most suitable transformer-type neural network for the desired application is selected. Then, the data structure related to the water/oil two-phase flow is developed using available information from various sources. The database is filtered and divided, ensuring that the training dataset contains the minimum and max-

imum values of each parameter used in the model’s inputs. Different configurations of the Transformer Neural Network are implemented to select the most suitable one based on the error and accuracy of each trained model. Finally, the most relevant conclusions are presented.

2. Materials and Methods

The architecture of the original TNN is based on the encoder–decoder, as depicted in Figure 1. This architecture is widely used for tasks such as machine translation, where a sequence of words is translated from one language to another. This architecture consists of two components: the encoder, which converts a sequence of input tokens into a sequence of embedding vectors, and the decoder, which iteratively uses the hidden state of the encoder to generate an output sequence of tokens, one token at a time.

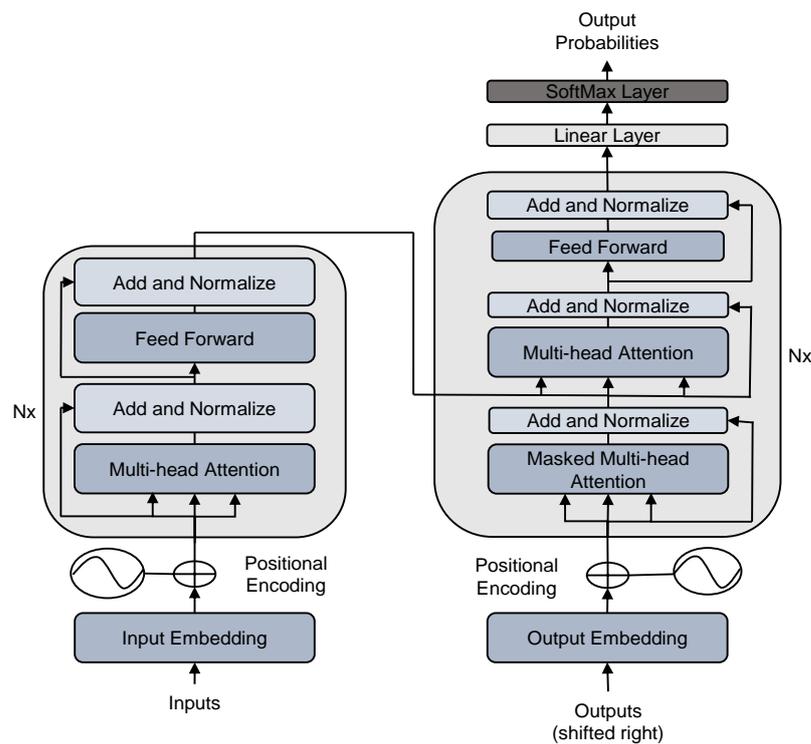


Figure 1. A comprehensive overview of the neural network structure for two-phase flow pattern identification in vertical pipelines using a transformer encoder–decoder architecture.

2.1. Types of Transformers

The architecture of the transformer was initially designed for sequence-to-sequence tasks like machine translation, but it was soon adapted with independent encoders and decoders, resulting in three different types of transformers [32]:

- Encoder-only: This model can convert an input sequence of text into a numerical representation suitable for tasks such as text classification or named entity recognition. The representation calculated for a given token in this architecture depends on bidirectional attention, meaning it relies on both the left and right context of the token.
- Decoder-only: This type of model is capable of automatically completing a sequence by iteratively predicting the most probable next word. In this case, the model relies on causal or autoregressive attention, meaning that the representation calculated for a token in this architecture depends solely on the left context.
- Encoder–decoder: these are used to model complex mappings from one sequence of text to another, making them suitable for tasks such as machine translation and summarization.

For this work, the encoder-only model is used. Therefore, the structure of the encoder block is explained in detail below. The structure of the encoder block is shown in Figure 2.

A sequence of embeddings enters this block and is processed through the following layers: a multi-head self-attention layer and a feed-forward layer.

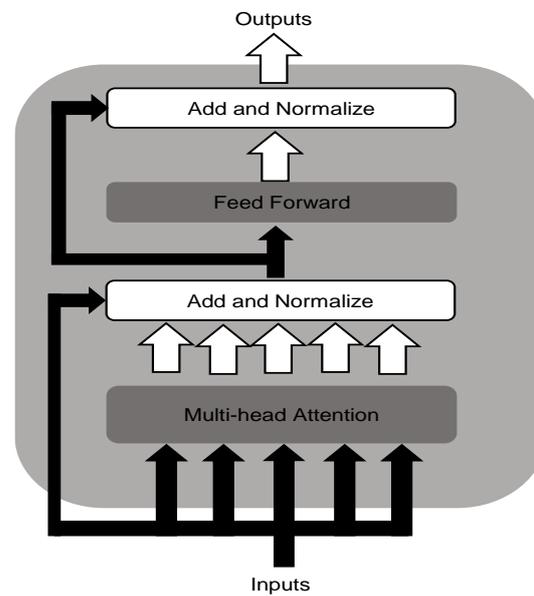


Figure 2. Structural components and workflow of the neural network encoder.

Self-attention is a mechanism that allows neural networks to assign a different amount of attention, meaning a different weight value, to each element of a sequence. Therefore, the main idea of self-attention is to use the entire sequence to calculate a weighted average for each embedding. One way to formulate this is that given a sequence of tokens X_1, \dots, X_n , self-attention produces a sequence of new embeddings X_1', \dots, X_n' , where each X_i' is a linear combination of all the X_j , as shown in Equation (1):

$$X_i' = \sum_{j=1}^n w_{ji} x_j \quad (1)$$

where the coefficient w_{ji} is the attention weight and is normalized in such a way that $\sum_j w_{ji} = 1$.

The self-attention process involves three vectors: Q , queries, K , keys, and V , values, which are derived by multiplying each input entry by weight matrices W (trained during training). A dot product is computed between the vector Q and the vector K . The result is then divided by the square root of the dimension size of the original input vectors (d_k), normalized using the *SoftMax* operation, and multiplied by the vector V . For computational efficiency, the information is treated in the matrix form. Equation (2) represents the attention process.

$$Attention(Q, K, V) = Z = SoftMax\left(\frac{Q * K^T}{\sqrt{d_k}}\right) * V \quad (2)$$

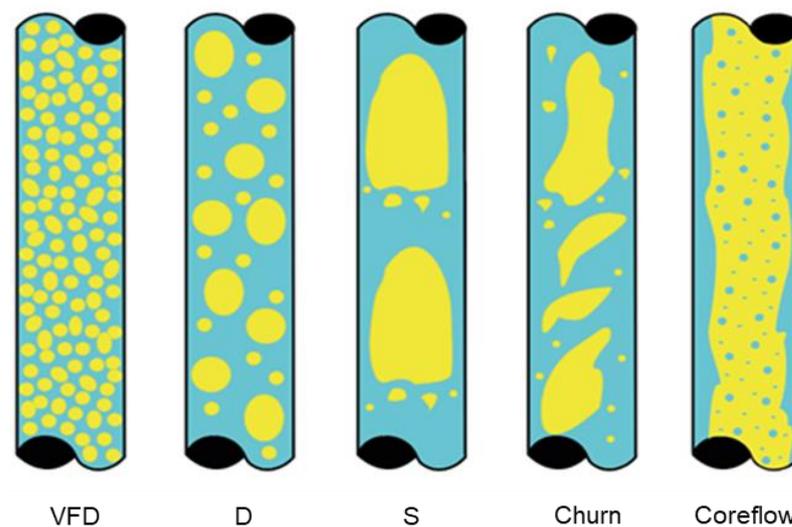
2.2. Data Structure

A database was structured based on experimental data already available in the literature related to two-phase flow consisting of water and oil in vertical pipelines. A total of 4864 data points were collected from 18 different authors, as detailed in Table 1, which provides comprehensive information regarding the data from each author. The table indicates the pipe diameter (D) used in the experiments, the viscosity (μ_o) and density (ρ_o) of the oil used, the number of flow patterns identified, and the quantity of extracted data.

Table 1. General information of the database for two-phase water and oil flow in vertical pipes: a comprehensive overview of key data parameters and characteristics.

Reference	D [m]	μ_o [Pa·s]	ρ_o [kg]/m ³	Number of Flow Patterns	Data
[33]	0.0284	0.488	925	3	392
[34]	0.1064	0.00188	800	6	49
[35]	0.02	0.011	856	3	101
[36]	0.05	0.02	850	5	126
[37]	0.04	0.035	860	4	92
[38]	0.0263	0.0201	851	2	15
[39]	0.02	0.125	801	4	175
[40]	0.127	0.001	801	2	9
[41]	0.027	0.02	801	4	109
[42]	0.0254	0.001	792	4	55
[43]	0.0254	0.001	792	5	32
[44]	0.01	0.287	1823	3	1421
[45]	0.026	0.0011	793	5	90
[46]	0.03	0.03	856	4	252
[47]	0.0284	0.5	930	1	92
[48]	0.03	0.044	860	4	98
[49]	0.02	0.09732	857	7	1699
[50]	0.04	0.0041	824	1	57

Based on the schemes developed in each study, a total of 9 distinct flow patterns were identified, characterized by variations in the continuous phase, either aqueous (w) or oily (o). These patterns include very fine dispersed droplets (VFDs) o/w (265 cases) and w/o (204 cases), drops (D) o/w (947 cases) and w/o (1292 cases), slug (S) o/w (459 cases) and w/o (656 cases), churn (21 cases) o/w and (126 cases) w/o, annular (Core flow) with 480 cases, and the transition region (TF) with 314 cases. Figure 3 provides a graphical representation illustrating the behavior of the most prominent flow patterns.

**Figure 3.** Two-phase flow patterns in vertical pipes: a comprehensive visualization of fluid dynamics and transition phenomena.

In general, when the presence of variation from one flow pattern to another is evident, it can be defined as a transition intermediate flow (TF). On the other hand, the studied database has information related to the surface velocities of the fluids involved and their mixture, the volume fraction of water and oil, the diameter of the implemented pipe, the viscosity of the oil, and the type of flow patterns developed.

2.3. Transformer Neural Network Model

The structure of the implemented model consists of the encoder implemented in TNN and a SoftMax layer at the model's output. The input vector X is composed of the superficial velocities of the fluids (oil and water), their sum, the in situ volumetric fraction (holdup) of both fluids, the viscosity of the oil, and the diameter of the circular cross-section pipe, as shown in Equation (3).

$$X = \begin{pmatrix} J_o \\ J_w \\ J_{o+w} \\ \epsilon_o \\ \epsilon_w \\ D \\ \mu_o \end{pmatrix} \quad (3)$$

The selected optimizer for training is Adam, as it integrates the characteristics of two methodologies: one is the gradient descent algorithm with momentum, and the other is the RMSP (Root Mean Square Propagation) algorithm. Gradient descent with momentum is used to accelerate the gradient descent algorithm by considering the exponentially weighted average of gradients, meaning that using averages makes the algorithm converge towards minima at a faster rate. This method is defined in Equations (4) and (5).

$$\omega_{t+1} = \omega_t - \alpha m_t \quad (4)$$

where

$$m_t = \beta m_{t-1} + (1 - \beta) \left[\frac{\delta L}{\delta \omega_t} \right] \quad (5)$$

where m_t is the aggregated gradients at the current time step t , initially set to 0. Additionally, m_{t-1} represents the aggregated gradients at time step $t - 1$, ω_t is the weight at time step t , ω_{t+1} is the weight at time step $t + 1$, α is the learning rate at time step t . Furthermore, δL is the derivative of the loss function, $\delta \omega_t$ is the derivative of weights at time step t , and $\beta = 0.9$ is the moving average parameter.

The Root Mean Square Propagation (RMSP), instead of accumulating squared gradients, uses exponential moving averages. It is defined as follows in Equation (6).

$$\omega_{t+1} = \omega_t - \frac{\alpha}{(v_t + \epsilon)^{\frac{1}{2}}} \left[\frac{\delta L}{\delta \omega_t} \right] \quad (6)$$

where

$$v_t = \beta v_{t-1} + (1 - \beta) \left[\frac{\delta L}{\delta \omega_t} \right]^2 \quad (7)$$

where v_t is the sum of past squared gradients, i.e., $\text{sum}(\delta L / \delta \omega_{t-1})$. Initially, $v_t = 0$. $\epsilon = 10^{-8}$ is a small positive constant to prevent division by zero when $v_t \rightarrow 0$, and $\alpha = 0.001$ is the learning rate parameter.

Therefore, Adam controls the gradient descent speed, making it possible to reach the global minimum with small steps and steps large enough to overcome local minima along the way, thus achieving efficiency. In this case, Equation (8) is used.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[\frac{\delta L}{\delta \omega_t} \right]; \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[\frac{\delta L}{\delta \omega_t} \right]^2 \quad (8)$$

where $\beta_1 = 0.9$ and, $\beta_2 = 0.999$ are the decay rates for the gradient averages in the two previous methods.

Given that m_t and v_t are initialized as 0 according to the previous methods, they tend to be biased towards 0 because β_1 and $\beta_2 \approx 1$. Adam solves this problem by computing bias-corrected m_t and v_t . This is carried out to have control over the weights while reaching

the global minimum and to avoid high oscillations when approaching it, meaning it adapts to gradient descent with each iteration. The equations used are (9) and (10):

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}; \widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{9}$$

therefore

$$\omega_{t+1} = \omega_t - \widehat{m}_t \left(\frac{\alpha}{\sqrt{\widehat{v}_t} + \epsilon} \right) \tag{10}$$

The loss function implemented in the training of the developed model is the Binary Cross entropy function, which is typically used in binary classification problems but can also be applied to problems where the variables to predict take values between 0 and 1. This function is defined in Equation (11):

$$\mathcal{L}_{BCE} = -\frac{1}{n} \sum_{j=1}^n \sum_{i=1}^c [y_i \log \widehat{y}_i + (1 - y_i) \log(1 - \widehat{y}_i)] \tag{11}$$

where y_i is the actual class to be predicted, \widehat{y}_i is the predicted probability for the class, c is the number of classes, and n is the number of examples.

Variations were performed using 4 different activation functions, among which we have the *ReLU* function, which transforms input values by zeroing out the negative values and leaving the positive ones unchanged. It is defined as Equation (12).

$$ReLU(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{12}$$

The second activation function is the sigmoid function, which is used in models where it is necessary to predict probabilities as an outcome, as it takes a range of values between 0 and 1. Its definition is shown in Equation (13).

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

Also, the hyperbolic tangent function was implemented, which is similar in continuity to the sigmoid function but has outputs in a range from -1 to 1 , and it is defined in Equation (14):

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

Finally, the GeLU activation function was used, as defined in Equation (15), which adds nonlinearity by multiplying a stochastic sigmoid function by the input, resembling the *ReLU* function.

$$GeLU(x) = \frac{1}{2} x \left[1 + \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right) \right] \tag{15}$$

Firstly, the learning rate (LR) was varied with the following values: 0.005, 0.001, and 0.0005. The dropout layer was considered with values of 0.1, 0.3, and 0.5. The attention heads in the multi-head-attention layer were set to 1, 2, and 4, and a single encoder was implemented. The embedding dimension was 32, and the feed-forward layer dimension was 32 as well. For each particular simulation, 55 epochs were considered during the neural network training process, and a batch size of 2800 was used. To compare the performance of different configurations, accuracy was evaluated as the number of correct predictions divided by the total number of data points for each phase.

Furthermore, a confusion matrix was generated for the models selected based on their performance, from which precision and accuracy were extracted for each of the flow patterns. It is necessary to calculate the values of true positives (TP), true negatives (TN),

false positives (FP), and false negatives (FN) first. Precision is defined in Equation (16), and accuracy is defined in Equation (17).

$$Precision = \frac{VP}{VP + FP} \tag{16}$$

$$Accuracy = \frac{VP + VN}{VP + FN + FP + VN} \tag{17}$$

3. Results

Tests were conducted with 110 different parameter configurations, from which eight models were selected based on configurations that exhibited minimal overfitting during the training phase while identifying the highest number of flow patterns in the validation phase. Table 2 presents the results obtained during the TNN training with various parameter variations. The table displays the precisions achieved by each model in different phases (training, testing, and validation).

Table 2. Precision results for the training, testing, and validation phases for different models.

Parameters	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
Activation function	sigmoid	ReLU	sigmoid	ReLU	sigmoid	Tanh	Gelu	ReLU
N° attention heads	2	4	4	4	1	2	2	2
Dropout	0.3	0.5	0.5	0.1	0.1	0.3	0.3	0.1
Learning rate	0.001	0.001	0.001	0.0005	0.001	0.001	0.001	0.001
Training accuracy (%)	99.92	99.54	99.49	99.38	99.97	99.61	99.72	99.97
Test accuracy (%)	52.25	51.23	51.43	51.64	55.53	52.87	49.80	54.10
Validation accuracy (%)	51.04	48.98	49.18	52.87	52.25	50.00	48.98	53.07

Conducting a detailed analysis of the accuracy achieved by the models developed using the Transformer Neural Network, it was possible to identify the best results during the training and testing phases, which correspond to models 1, 5, and 8. These models were selected because their accuracy exceeds 99.9% during the training phase and 52% during the testing phase.

Figure 4 displays the precision curves generated during the training phase for the three selected models. It can be observed that models 5 and 8 tend to reach their maximum value close to one in the epoch range between epochs 46 and 55.

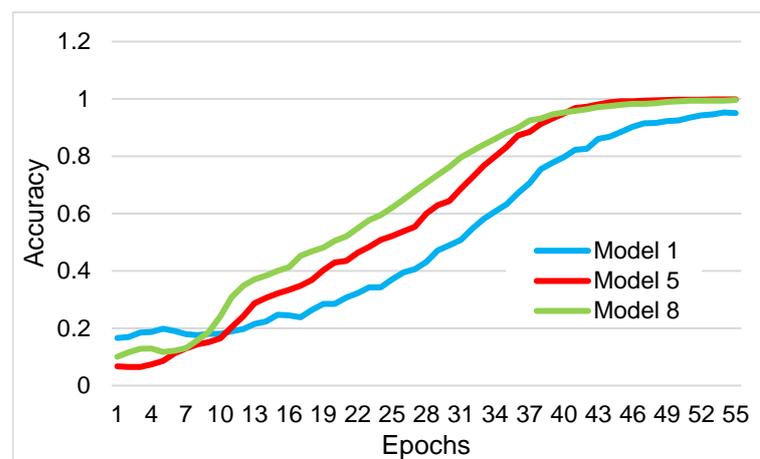


Figure 4. Precision behavior during the training phase.

Additionally, it is possible to identify that the curve generated by model 8 shows a growing trend when iterated with a smaller number of epochs compared to model 5. This is because its values, until reaching maximum precision, always remain lower than the

curve of model 8. This behavior is generated by the optimized structure created for model 8. In this case, the internal mathematical process of the TNN accelerates by implementing the ReLU activation function, which does not activate all layers simultaneously, combined with a dropout of 0.1 and two attention heads.

On the other hand, the precision curve generated by model 1 shows a slower increasing trend, which required a greater number of epochs beyond 55 to reach values close to unity. This demonstrates that despite achieving a precision of 99.92%, it is not the most efficient model. This is because the sigmoid function exhibits slow growth when the gradient descent approaches zero, delaying the learning of the model in question.

In Figure 5, the precision curves generated during the testing phase are presented. It is observed that model 5 achieves higher precision from epoch number 40 onwards, which is the same epoch when the precision curves of models 1 and 8 have already stabilized at their maximum values of 52.25% and 54.10%, respectively. Additionally, the curve generated by model 5 exhibits gradual growth, lower than the curves presented by models 1 and 8. Despite having better precision in the testing phase, model 1 consumes more computational resources as it requires a greater number of epochs to stabilize at its maximum value, which is not efficient when considering that models 1 and 8 only require 31 epochs to reach this point.

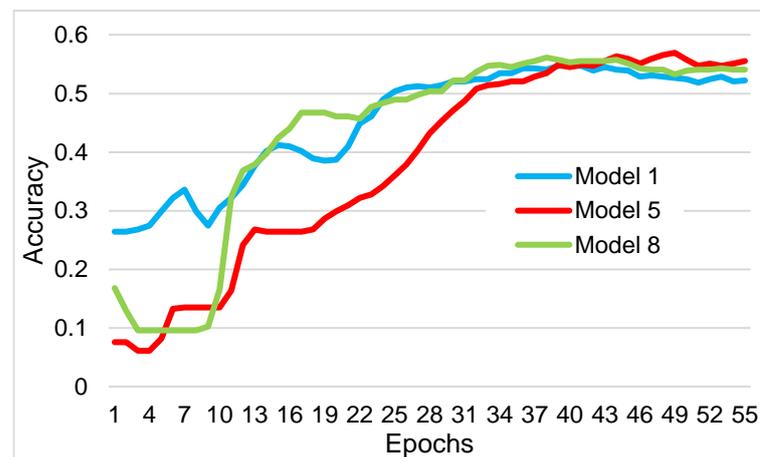


Figure 5. Accuracy behavior during the testing phase.

For the selected models, confusion matrices were constructed to analyze the behavior of predictions for each of the flow patterns. In the confusion matrix, the true positives (TPs) for each category are observed, meaning the number of data points that were correctly predicted for a specific flow pattern. Likewise, other correctly predicted patterns will be true negatives (TNs), i.e., the other values on the main diagonal. False positives (FPs) are those that the model believes it is predicting correctly; these values correspond to the column of the predicted flow pattern. False negatives (FNs) are values corresponding to the row of the true flow pattern. These confusion matrices can be seen in Figure 6a–c. For example, for model 1, taking the D w/o pattern as a reference, TP = 99, adding the other values on the main diagonal results in TN = 150, summing the other values in the D w/o column gives FP = 104, and summing the other values in the D o/w row results in FN = 31.

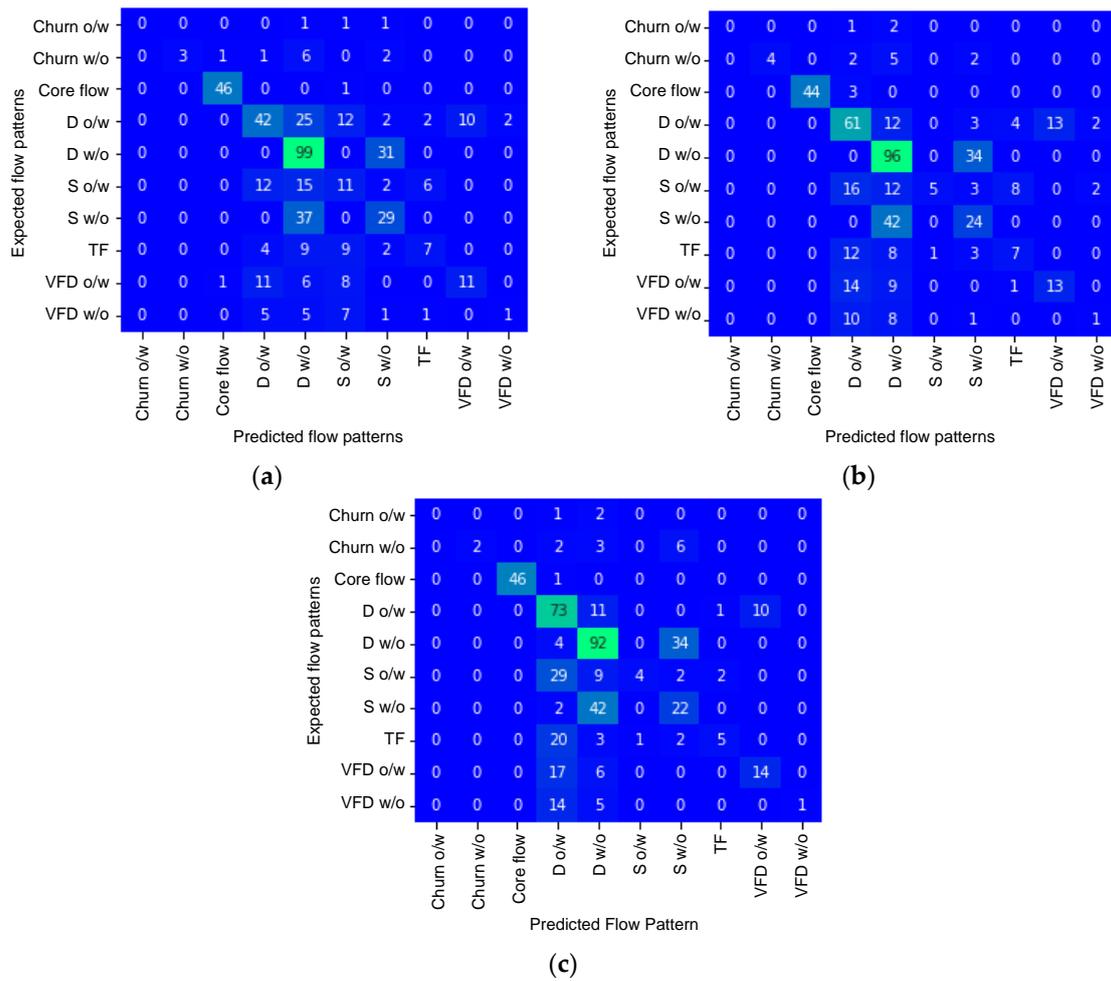


Figure 6. Confusion matrix for: (a) model 1, (b) model 5, and (c) model 8.

Table 3 displays the precision and accuracy values by flow type for the three selected models, obtained by applying Equations (16) and (17). According to Table 3, it can be observed that the best performance corresponds to model 8, with an average precision of 63.21% and an accuracy of 86.51%.

Table 3. Precision and accuracy results by flow type according to the predictions made by the models.

	Model 1		Model 5		Model 8	
	Precision [%]	Accuracy [%]	Precision [%]	Accuracy [%]	Precision [%]	Accuracy [%]
Churn o/w	0.00	98.81	0.00	98.84	0.00	98.85
Churn w/o	100.00	96.14	100.00	96.59	100.00	95.93
Core flow	95.83	98.81	100.00	98.84	100.00	99.62
D o/w	56.00	74.33	51.26	73.49	44.79	70.00
D w/o	48.77	64.84	49.48	65.89	53.18	68.52
S o/w	22.45	77.33	83.33	85.86	80.00	85.76
S w/o	41.43	76.15	34.29	74.34	33.33	74.64
TF	43.75	88.30	35.00	87.33	62.50	89.93
VFD o/w	52.38	87.37	50.00	87.33	58.33	88.70
VFD w/o	33.33	92.25	20.00	91.76	100.00	93.17
Average	49.39	85.43	52.34	86.03	63.21	86.51

Figure 7 presents the flow map structured from the gathered information, using the superficial velocities of oil and water as references. In this map, you can identify the different regions where the 10 flow patterns listed initially appear.

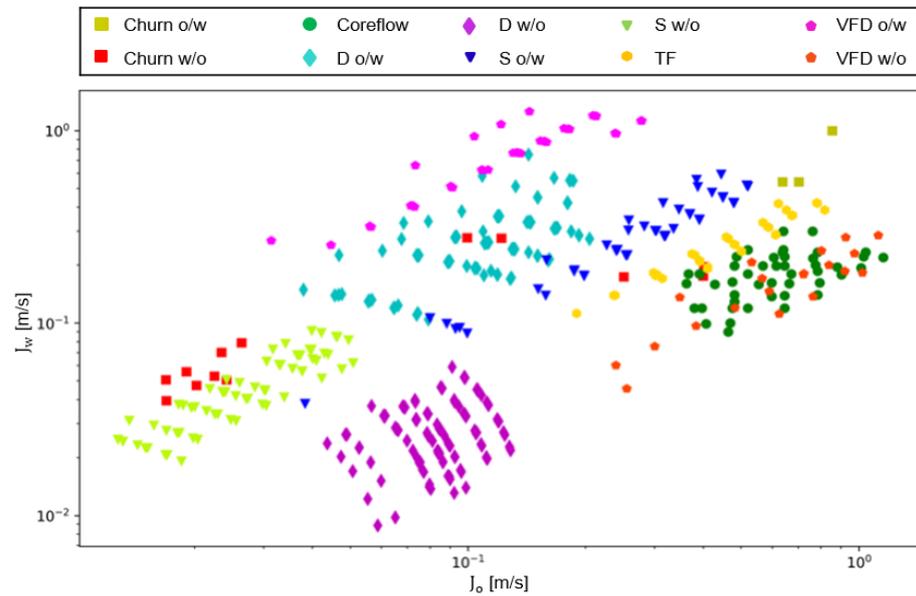


Figure 7. Flow map structured with the information collected from the literature for all authors.

In Figure 8, the predicted flow map by model 8 is presented. It is possible to observe that the selected model can predict 9 out of the 10 types of flow patterns shown in Figure 7. However, model 8 fails to predict the churn o/w flow pattern correctly. This is because in the used database, it is the flow pattern with the fewest occurrences (21), and only 15 of these are used for training when considering the defined percentage segmentation. This generates a significant data imbalance when compared to the D o/w pattern with 1196 data points, for example. The map in Figure 8 shows clusterization similar to the reference data, with some overlap in certain areas, except for the churn o/w flow pattern, which is absent in the predictions of model 8.

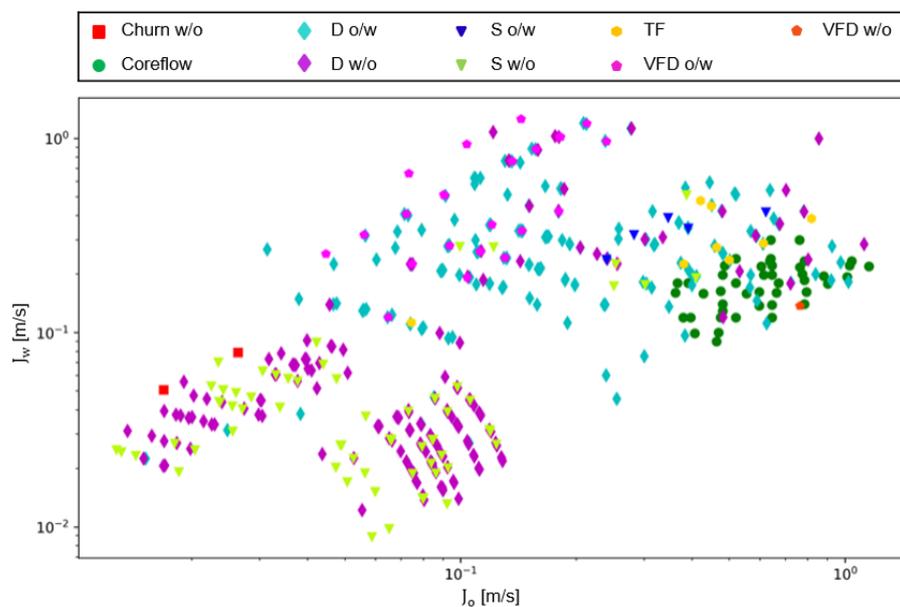


Figure 8. Flow map structured from the results obtained by model 8.

4. Conclusions

A database of two-phase flow patterns was structured based on information reported in the literature by various authors, comprising a total of 4864 experiments. This database includes seven parameters related to multiphase flows and identifies 10 different flow patterns.

A Transformer Neural Network was developed using Python to predict two-phase flow patterns, achieving a training accuracy of 99.97% and a validation accuracy of 53.07% for model 8. This model successfully predicted 9 out of the 10 flow patterns used in the training phase with an average precision of 63.21% and an average accuracy of 86.51%. This indicates that the Transformer Neural Network can predict different flow patterns with high accuracy and reasonable precision. It is worth noting that the TNN does not require excessive computational resources for training, as it achieves near-perfect accuracy during this phase with a substantial batch size and a small number of epochs, making it an efficient neural network.

Flow maps were constructed using both reference data and the predictive model, revealing a striking resemblance between the map generated from predicted data and the one generated from experimental data. Additionally, data clustering and transition zones between flow patterns were clearly evident.

The best prediction results from the models were achieved with sigmoid and ReLU activation functions, using a learning rate of 0.001. Regarding the number of attention heads, good results were observed with either one or two heads. Additionally, it is worth mentioning that a dropout rate of 0.1 or 0.3 is sufficient.

The study underscores the effectiveness of the Transformer Neural Network (TNN) model in identifying two-phase flow patterns in vertical pipelines, not reported before, and highlights its potential applications in the oil and gas industry. The TNN model's precision in recognizing flow patterns suggests opportunities for optimizing hydrocarbon transportation processes, leading to increased operational efficiency and reduced costs. Moreover, its implementation in monitoring systems could aid in the early detection of flow patterns contributing to pipeline corrosion, enhancing facility integrity management. In oil and gas production, understanding flow patterns using the TNN model could improve production efficiency by allowing real-time adjustments in extraction and processing. Furthermore, the TNN model could be incorporated into ongoing monitoring systems to formulate predictive maintenance strategies, anticipating possible concerns related to equipment wear.

As further work, it is recommended to create a database with a larger amount of data, ensuring a balanced representation of data points for each flow pattern implemented. This will improve the TNN model, mitigating the risk of biasing predictions toward the flow pattern with the highest data volume during the training phase.

Author Contributions: Conceptualization, C.M.R.-D., E.E.P.-P. and O.A.G.-E.; Data curation, C.M.R.-D.; Formal analysis, E.E.P.-P. and O.A.G.-E.; Funding acquisition, O.A.G.-E.; Investigation, E.E.P.-P. and O.A.G.-E.; Methodology, C.M.R.-D. and O.A.G.-E.; Project administration, O.A.G.-E.; Supervision, O.A.G.-E.; Validation, C.M.R.-D. and O.A.G.-E.; Writing—original draft, E.E.P.-P.; Writing—review and editing, C.M.R.-D. and O.A.G.-E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Universidad Industrial de Santander (grant number VIE-3714). We acknowledge the support of the Industrial Multiphase Flow Laboratory (LEMI) of the Sao Carlos School of Engineering, University of Sao Paulo.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Rosa, E.S. *Escoamento Multifásico Isotérmico: Modelos de Multifluidos e de Mistura*; Bookman: Porto Alegre, Brasil, 2012; ISBN 9788540700703.
2. Hernández-Cely, M.M.; Ruiz-Díaz, C. Estudio de los fluidos aceite-agua a través del sensor basado en la permitividad eléctrica del patrón de fluido. *Rev. UIS Ing.* **2020**, *19*, 177–186. [[CrossRef](#)]
3. Santos, D.S.; Faia, P.M.; Garcia, F.A.P.; Rasteiro, M.G. Oil/water stratified flow in a horizontal pipe: Simulated and experimental studies using EIT. *J. Pet. Sci. Eng.* **2019**, *174*, 1179–1193. [[CrossRef](#)]
4. Liu, H.; Duan, J.; Li, J.; Gu, K.; Lin, K.; Wang, J.; Yan, H.; Guan, L.; Li, C. Numerical quasi-three dimensional modeling of stratified oil-water flow in horizontal circular pipe. *Ocean Eng.* **2022**, *251*, 111172. [[CrossRef](#)]
5. Obaseki, M.; Elijah, P.T.; Alfred, P.B. Development of model to eliminate sand trapping in horizontal fluid pipelines. *J. King Saud Univ.-Eng. Sci.* **2022**, *34*, 425–434. [[CrossRef](#)]
6. Gomez, C.; Ruiz, D.; Cely, M. Specialist system in flow pattern identification using artificial neural Networks. *J. Appl. Eng. Sci.* **2023**, *21*, 285–299. [[CrossRef](#)]
7. Bazon, P.B.; Castro-Bolivar, J.E.; Ruiz-Diaz, C.M.; Hernández-Cely, M.M.; Rodriguez, O.M.H. Hybrid machine learning model applied to phase inversion prediction in liquid-liquid pipe flow. *Multiph. Sci. Technol.* **2023**, *35*, 35–53. [[CrossRef](#)]
8. de Almeida Coelho, N.M.; Taqueda, M.E.S.; Souza, N.M.O.; de Paiva, J.L.; Santos, A.R.; Lia, L.R.B.; de Moraes, M.S.; de Moraes Júnior, D. Energy savings on heavy oil transportation through core annular flow pattern: An experimental approach. *Int. J. Multiph. Flow* **2020**, *122*, 103127. [[CrossRef](#)]
9. Sánchez, E.; Romero, C.; Zeppieri, S.; González-Mendizabal, D. Estudio experimental sobre patrones de flujo líquido-líquido en tuberías verticales. In *Proceedings of the 8° Congreso Iberoamericano de Ingeniería Mecánica*; Universidad Pontificia del Peru: Cusco, Mexico, 2007.
10. Ruiz-Díaz, C.; Hernández-Cely, M.M.; González-Estrada, O.A. Modelo predictivo para el cálculo de la fracción volumétrica de un flujo bifásico agua- aceite en la horizontal utilizando una red neuronal artificial. *Rev. UIS Ing.* **2022**, *21*, 155–164. [[CrossRef](#)]
11. Sanguino, V. Modelamiento en CFD de Flujo Bifásico (Aceite-Agua) en una Tubería Vertical. Bachelor's Thesis, Universidad de los Andes, Bogotá, Colombia, 2014.
12. Vasheghani Farahani, M.; Jahanpeyma, Y.; Taghikhani, V. Dynamic Modeling and Numerical Simulation of Gas Lift Performance in Deviated Oil Well Using the Two-Fluid Model. In Proceedings of the 80th EAGE Conference and Exhibition 2018, Copenhagen, Denmark, 11–14 June 2018; European Association of Geoscientists & Engineers: Utrecht, The Netherlands, 2018. [[CrossRef](#)]
13. Hassanpouryouzband, A.; Yang, J.; Tohidi, B.; Chuvilin, E.; Istomin, V.; Bukhanov, B. Geological CO₂ Capture and Storage with Flue Gas Hydrate Formation in Frozen and Unfrozen Sediments: Method Development, Real Time-Scale Kinetic Characteristics, Efficiency, and Clathrate Structural Transition. *ACS Sustain. Chem. Eng.* **2019**, *7*, 5338–5345. [[CrossRef](#)]
14. Hassanpouryouzband, A.; Joonaki, E.; Taghikhani, V.; Bozorgmehry Boozarjomehry, R.; Chapoy, A.; Tohidi, B. New Two-Dimensional Particle-Scale Model To Simulate Asphaltene Deposition in Wellbores and Pipelines. *Energy Fuels* **2018**, *32*, 2661–2672. [[CrossRef](#)]
15. Norouzi, S.; Nazari, M.; VasheghaniFarahani, M. A Novel Hybrid Particle Swarm Optimization-Simulated Annealing Approach for CO₂-Oil Minimum Miscibility Pressure (MMP) Prediction. In Proceedings of the 81st EAGE Conference and Exhibition 2019, London, UK, 3–6 June 2019; European Association of Geoscientists & Engineers: Utrecht, The Netherlands, 2019; pp. 1–5. [[CrossRef](#)]
16. Sossa, J. El papel de la inteligencia artificial en la Industria 4.0. In *Inteligencia Artificial y Datos Masivos en Archivos Digitales Sonoros y Audiovisuales*; Rodríguez Reséndiz, P.O., Ed.; Univ. Nacional Autónoma de México: Mexico City, Mexico, 2020.
17. Valle Tamayo, G.A.; Romero Consuegra, F.; Cabarcas Simancas, M.E. Predicción de flujo multifásico en sistemas de recolección de crudo: Descripción de requerimientos. *Rev. Fuentes Reventón Energético* **2017**, *15*, 87–99. [[CrossRef](#)]
18. Ruiz-Díaz, C.M.; Hernández-Cely, M.M.; González-Estrada, O.A. A Predictive Model for the Identification of the Volume Fraction in Two-Phase Flow. *Cienc. Desarro.* **2021**, *12*, 49–55.
19. Jahanandish, I.; Salimifard, B.; Jalalifar, H. Predicting bottomhole pressure in vertical multiphase flowing wells using artificial neural networks. *J. Pet. Sci. Eng.* **2011**, *75*, 336–342. [[CrossRef](#)]
20. Semeraro, F.; Griffiths, A.; Cangelosi, A. Human-robot collaboration and machine learning: A systematic review of recent research. *Robot. Comput. Integr. Manuf.* **2022**, *79*, 102432. [[CrossRef](#)]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
22. Lakew, S.M.; Cettolo, M.; Federico, M. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In Proceedings of the COLING 2018—27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 641–652.

23. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *2020*, 1154–1156.
24. Safi Samghabadi, N.; Patwa, P.; Pykl, S.; Mukherjee, P.; Das, A.; Solorio, T. Aggression and Misogyny Detection using BERT: A Multi-Task Approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*; European Language Resources Association (ELRA): Marseille, France, 2020; pp. 126–131.
25. Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; Lim, S.-N. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 20–25 June 2021; pp. 12309–12318.
26. Le, N.Q.K.; Ho, Q.T. Deep transformers and convolutional neural network in identifying DNA N6-methyladenine sites in cross-species genomes. *Methods* **2022**, *204*, 199–206. [[CrossRef](#)] [[PubMed](#)]
27. Sharma, A.; Jain, P.; Mahgoub, A.; Zhou, Z.; Mahadik, K.; Chaterji, S. Lerna: Transformer architectures for configuring error correction tools for short- and long-read genome sequencing. *BMC Bioinform.* **2022**, *23*, 25. [[CrossRef](#)]
28. Behjati, A.; Zare-Mirakabad, F.; Arab, S.S.; Nowzari-Dalini, A. Protein sequence profile prediction using ProtAlbert transformer. *Comput. Biol. Chem.* **2022**, *99*, 107717. [[CrossRef](#)]
29. Qu, K.; Si, G.; Shan, Z.; Kong, X.G.; Yang, X. Short-term forecasting for multiple wind farms based on transformer model. *Energy Rep.* **2022**, *8*, 483–490. [[CrossRef](#)]
30. Wang, L.; He, Y.; Li, L.; Liu, X.; Zhao, Y. A novel approach to ultra-short-term multi-step wind power predictions based on encoder–decoder architecture in natural language processing. *J. Clean. Prod.* **2022**, *354*, 131723. [[CrossRef](#)]
31. Gomaa, I.; Gowida, A.; Elkatatny, S.; Abdurraheem, A. The prediction of wellhead pressure for multiphase flow of vertical wells using artificial neural networks. *J. Geosci.* **2021**, *14*, 1–10. [[CrossRef](#)]
32. Tunstall, L.; Von Werra, L.; Wolf, T. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*; O'Reilly: Sebastopol, CA, USA, 2022; ISBN 978-1-098-10324-8.
33. Bannwart, A.C.; Rodriguez, O.M.H.; de Carvalho, C.H.M.; Wang, I.S.; Vara, R.M.O. Flow Patterns in Heavy Crude Oil-Water Flow. *J. Energy Resour. Technol.* **2004**, *126*, 184–189. [[CrossRef](#)]
34. Abduvayt, P.; Manabe, R.; Watanabe, T.; Arihara, N. Analysis of Oil-Water Flow Tests in Horizontal, Hilly-Terrain, and Vertical Pipes. In *Proceedings of the SPE Annual Technical Conference and Exhibition*; Society of Petroleum Engineers: London, UK, 2004; pp. 1335–1347. [[CrossRef](#)]
35. Du, M.; Jin, N.-D.; Gao, Z.-K.; Wang, Z.-Y.; Zhai, L.-S. Flow pattern and water holdup measurements of vertical upward oil–water two-phase flow in small diameter pipes. *Int. J. Multiph. Flow* **2012**, *41*, 91–105. [[CrossRef](#)]
36. Flores, J.; Chen, X.; Brill, J. Characterization of Oil-Water Flow Patterns in Vertical and Deviated Wells. In *Proceedings of the SPE Annual Technical Conference and Exhibition*; Society of Petroleum Engineers: London, UK, 1997; Volume 1, pp. 55–64. [[CrossRef](#)]
37. Ganat, T.; Ridha, S.; Hairir, M.; Arisa, J.; Gholami, R. Experimental investigation of high-viscosity oil–water flow in vertical pipes: Flow patterns and pressure gradient. *J. Pet. Explor. Prod. Technol.* **2019**, *9*, 2911–2918. [[CrossRef](#)]
38. Govier, G.W.; Sullivan, G.A.; Wood, R.K. The upward vertical flow of oil-water mixtures. *Can. J. Chem. Eng.* **1961**, *39*, 67–75. [[CrossRef](#)]
39. Han, Y.F.; Jin, N.D.; Zhai, L.S.; Zhang, H.X.; Ren, Y.Y. Flow pattern and holdup phenomena of low velocity oil-water flows in a vertical upward small diameter pipe. *J. Pet. Sci. Eng.* **2017**, *159*, 387–408. [[CrossRef](#)]
40. Hasan, A.R.; Kabir, C.S. A New Model for Two-Phase Oil/Water Flow: Production Log Interpretation and Tubular Calculations. *SPE Prod. Eng.* **1990**, *5*, 193–199. [[CrossRef](#)]
41. Hasan, A.R.; Kabir, C.S. A simplified model for oil/water flow in vertical and deviated wellbores. *SPE Prod. Facil.* **1999**, *14*, 56–62. [[CrossRef](#)]
42. Jana, A.K.; Das, G.; Das, P.K. Flow regime identification of two-phase liquid–liquid upflow through vertical pipe. *Chem. Eng. Sci.* **2006**, *61*, 1500–1515. [[CrossRef](#)]
43. Jana, A.K.; Ghoshal, P.; Das, G.; Das, P.K. An Analysis of Pressure Drop and Holdup for Liquid-Liquid Upflow through Vertical Pipes. *Chem. Eng. Technol.* **2007**, *30*, 920–925. [[CrossRef](#)]
44. Guo, J.; Yang, Y.; Zhang, S.; Zhang, D.; Cao, C.; Ren, B.; Liu, L.; Xing, Y.; Xiong, R. Heavy oil-water flow patterns in a small diameter vertical pipe under high temperature/pressure conditions. *J. Pet. Sci. Eng.* **2018**, *171*, 1350–1365. [[CrossRef](#)]
45. Mazza, R.A.; Suguimoto, F.K. Experimental investigations of kerosene-water two-phase flow in vertical pipe. *J. Pet. Sci. Eng.* **2020**, *184*, 106580. [[CrossRef](#)]
46. Mydlarz-Gabryk, K.; Pietrzak, M.; Troniewski, L. Study on oil–water two-phase upflow in vertical pipes. *J. Pet. Sci. Eng.* **2014**, *117*, 28–36. [[CrossRef](#)]
47. Rodriguez, O.M.H.; Bannwart, A.C. Experimental study on interfacial waves in vertical core flow. *J. Pet. Sci. Eng.* **2006**, *54*, 140–148. [[CrossRef](#)]
48. Xu, J.; Li, D.; Guo, J.; Wu, Y. Investigations of phase inversion and frictional pressure gradients in upward and downward oil–water flow in vertical pipes. *Int. J. Multiph. Flow* **2010**, *36*, 930–939. [[CrossRef](#)]

49. Yang, Y.; Guo, J.; Ren, B.; Zhang, S.; Xiong, R.; Zhang, D.; Cao, C.; Liao, Z.; Zhang, S.; Fu, S. Oil-Water flow patterns, holdups and frictional pressure gradients in a vertical pipe under high temperature/pressure conditions. *Exp. Therm. Fluid Sci.* **2019**, *100*, 271–291. [[CrossRef](#)]
50. Zhao, D.; Guo, L.; Hu, X.; Zhang, X.; Wang, X. Experimental study on local characteristics of oil-water dispersed flow in a vertical pipe. *Int. J. Multiph. Flow* **2006**, *32*, 1254–1268. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.