Fault Diagnosis For Drilling using a Multitask Physics-Informed Neural Network

Marios Gkionis * **Nils Christian Aars Wilhelmsen** *
Ole Morten Aamo *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway*

Abstract:
Mechanical faults, mud loss, and insufficient cuttings transport bear significant costs and can appear at unpredictable times during drilling operations. Early detection and diagnosis of such faults to support decisions is essential to avoid severe consequences and lengthy delays. We propose a novel system for Drilling Fault Diagnosis that combines principles of Physics-Informed Neural Networks (PINN) and Multitask Learning (MTL). Since measurements down-hole in the well are rarely available in real time, our proposed system uses measurements of flow and pressure at the drilling rig, only, from which type of fault and accompanying diagnostics such as depth in the well of the fault and its severity are predicted. State-of-the-art strategies of MTL with PINNs are deployed for effective Neural Network (NN) training. Generalization performance is shown to be high as evaluated using randomly generated values for the diagnostic variables. Drilling data collected during normal drilling-ahead conditions may be utilized in the training phase to identify uncertain characteristics of the well, thereby increasing the quality of the physics prior available to the PINN, and in turn improving prediction accuracy of faults. The potential usefulness of the method is illustrated in a simulation, admittedly under quite ideal conditions.

Keywords:
Fault Diagnosis, Neural Networks, Multitask Learning, Physics-Informed Neural Networks, Drilling, Washout, Pack-off, Mud-loss

1. INTRODUCTION

Fault Detection and Diagnosis (FDD) is a general field of engineering that is tasked with detecting the presence of a system fault (i.e., specifying which one is active) and diagnosing it (i.e., quantifying and localizing it). To this day, accurate and reliable FDD remains a challenge. Many traditional techniques have been developed for FDD (Isermann (2006)). These can be divided in *data-based* and *model-based* techniques. Data-based techniques (Venkatasubramanian et al. (2003)) leverage historic process knowledge. In factories, for instance, data from prior faults can be used to detect faults or even predict faults before they happen. In the drilling case, no historic data is available, because the "process" is new for every drilling operation. Most relevant to the present work, therefore, are the model-based methods. Within this category, it is common to design a *bank of observers* (Zhang (2000)), which may be based on Kalman filters, such as for example in Jiang et al. (2020). The individual faults are incorporated into separate models for which observers are designed that estimate process states and predict process outputs. The difference between the outputs predicted by the observers and the actual process outputs, referred to as residuals, are then analyzed for statistical changes, which then identify the faults. The observers and the statistical change detection algorithm must be carefully designed. An example of an application of this method to drilling can be found in Willersrud et al. (2015). A drawback of the results in Willersrud et al. (2015) is that they rely on measurements taken down-hole the well, which are rarely available in practice. In the present work, we rely on topside measurements taken at the rig, only.

Rapid growth of data collection and data-based techniques has driven significant research in Deep Learning (DL), making it a viable alternative to traditional methods for many applications. Improvements in DL have resulted in the increase of prediction accuracy, achievement of explainability, and reduction of training time and memory utilization (Alzubaidi et al. (2021)). Multitask Learning (MTL) and Physics-Informed Neural Networks (PINNs) are techniques that can enhance generalization performance and data efficiency. PINNs improve the data efficiency of a Machine Learning (ML) algorithm (Raissi et al. (2019)) by incorporating mathematical models that help the NN encode underlying physics that should hold true. Data can be combined with physics priors in this manner, or the physics prior may entirely replace process measurement data. The latter is the attractive option for drilling operations since historic field data is not available in this case.

MTL NNs (Caruana (1997)) are types of NNs that are being trained for multiple separate prediction tasks, using a set of shared and task-specialized parameters. As a Deep Neural Network (DNN), analysis of convergence guarantees and appropriateness for the parameter estimation scheme is not applicable, making it suitable for quicker and less specialized implementation. MTL is still in an early stage of testing on FDD problems. The current work serves as a starting point for investigating the application of MTL and PINNs for FDD.

In numerous DL training cases, there are prediction tasks that are inter-related through a latent/unknown mechanism which can be encoded in a MTL Network. Learning each training task separately with separate NNs would not only require calculation the forward passes of shared features as many times as the separate task-networks, but they would also miss encoding the common features leading to poorer generalization. For example, this is the case of the application in Wang et al. (2021), where vibration signals in rolling bearings are taken into account, thus enriching the information encoded in the shared features. MTL can offer enhancement of learning performance through the application of auxiliary tasks. This is the case of Amyar et al. (2020), wherein the COVID classification task is added to enhance feature representation of the main tasks. In addition, MTL can be useful in cases wherein the sensor information is not as rich as required for successful Single Task Learning (STL), as stressed in Wang et al. (2021).

In the present work, the input training data was generated using a steady-state drilling hydraulics model, therefore we consider it a type of PINN. Producing time-series and leveraging automatic differentiation is a subsequent goal for continuation of this work. The PINN approach resembles the model-based approach in the sense that separate models for each fault are required to provide the training data ("*Physics Information*"). Observer design is not explicitly needed, since feature representations of the various models can be shared in the NN. What is more, the statistical assessment of residuals in the case of *design bank of observers* is in principle incorporated into and learned by the NN. We considered three (3) different flow-related faults, which are described at the beginning of Section 3.

Despite not being mentioned in analytic overviews with respect to NN variants and FDD applications such as Qiu et al. (2023), a few publications on Drilling FDD using NNs and PINNs exist. For example, in Jeong et al. (2020) and Jan et al. (2022), Convolutional NNs are applied, in order to learn the faults given multi-channel time series as inputs. Jeong et al. (2020) and Jan et al. (2022) focus solely on the Washout Fault Detection and are the only published works that apply PINNs and MTL for Drilling applications up to our knowledge. Specifically, in Jan et al. (2022), the different tasks are the classification task, and the task of Physical Constraints. However, the Physics Prior involved in the estimation is conditioned to a specific parametric model, which limits the generality of the NN.

Using a slight variation of the typical MTL architecture, we were able to train a DNN that can diagnose the faults with a high level of accuracy, both in the training set, validation sets, and relatively low training time (around 20 minutes) for 800 epochs. This was achieved through intense overparameterization, large NN depths, and the leveraging of *last-layer, hard parameter sharing MTL*. Up to our knowledge, this work introduces the first MTL-PINN algorithm for Fault-Diagnosis of multiple different drilling faults. What is more, the algorithm is data efficient in that it requires only topside pump pressure and return flow measurements with only 300 training datapoints and achieves high generalization performance. This is a clear advantage compared to existing solutions that require more measurements across the drillstring.

2. FDD WITH PINN

The process we consider has input $u \in \mathbb{R}^{D_u}$ and output $x \in \mathbb{R}^{D_x}$, which are both measured at every time t. At any time, one of T different faults may occur in the process, affecting the relationship between u and x. Every fault $\tau \in \{1, \ldots, T\}$ can be characterized by $d_\tau \in \mathbb{R}^{D_\tau}$, referred to as the *diagnostics* of the fault. In the application under study, the values are $D_x = 2$, $D_u = 3$, $D_1 = D_2 = 2$, and $D_3 = 1$. Among the τ indices, one corresponds to the fault-free state, in which the process is considered to be operating under *nominal* conditions. We assume that we have at our disposal mathematical models describing the relationship between the input and the output, given the diagnostics. In other words, it is assumed that the models

$$x = f_\tau(u, d_\tau), \tau \in 1, \ldots, T \tag{1}$$

are available. We assume further, that for fixed d_τ, d_τ can be determined uniquely from a set of sufficiently varying outputs x_i, obtained by varying the input u in a predetermined way u_i, all for $i \in \{1, \ldots, M\}$. The main idea of the paper is to train a MTL-PINN to predict the diagnostics, given such a set of outputs from the process. Several potential configurations of the NN can be considered, and three options are sketched in Figure 1. We distinguish in particular between *a-priori-detection* (Figure 1-a)) and *simultaneous detection* (Figure 1 b-c). In the former case, the prediction of type of fault, that is the index τ, is provided by a separate classification NN, while in the latter case, it is incorporated into the network for the diagnostics. Quite often, detecting the type of fault is feasible with less variation in the data than what is required for predicting the diagnostics. Since varying the input to the process for collecting varying output data is associated with a cost, it is desirable to only do so after the occurrence of a fault has been detected, and avoid disturbing the process when it operates normally. This is the case in the drilling application, which motivates this work and is presented in the next section, and therefore we focus on the *a-priori* configuration of Figure 1-a) in the following.

It is assumed that no real data is available from the process under faulty operation, and only the models of (1) can be used for the NN training. For this reason, the concept of PINNs is employed, incorporating the models of the various faults into the loss function. Thus, we train the MTL-PINN using the loss function

$$\mathcal{L} = \sum_{\tau=1}^{T} w_\tau \sum_{d \sim \mathcal{D}_\tau} \mathcal{L}_\tau(\hat{d}, d), \;\; \hat{d} = f_o^\tau(f_{sh}(x_\tau)) \tag{2}$$

where $x_\tau = [f_\tau(u_1, d), \ldots, f_\tau(u_M, d)]^T$, $f_{sh}(\cdot)$ denotes the network evaluation producing shared feature representations, $f_o^\tau(\cdot)$ denotes the task-specific network evaluation producing d_τ from the shared feature representations and $\mathcal{D}_\tau = [0,1]^{D_\tau} \subset \mathbb{R}^{D_\tau}$. The notation $d \sim \mathcal{D}_\tau$ indicates uniform sampling of d from $\mathcal{D}_\tau$. The first summation in (2) renders the problem a MTL problem, with task weights w_τ that can be optionally dictated by a scheduling algorithm. $\mathcal{L}_\tau(\hat{d}, d)$ is the functional form of the loss term for each task (for example the MSE functional, $\mathcal{L}_\tau(\hat{d}, d) = |\hat{d} - d|^2/|\mathcal{D}_\tau|$). Notice that the effect of f_o^τ is that the loss term for each task is calculated by deactivating (masking) the other terms, which constitutes a slight modification from the typical *last-layer MTL* NN.

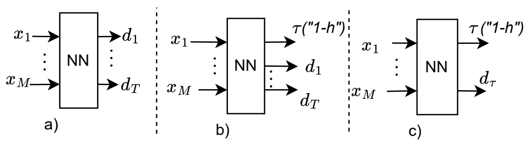
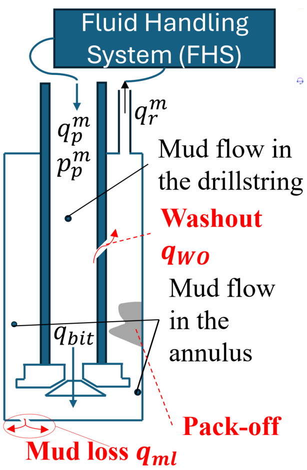


Fig. 1. Alternative designs for Fault Detection schemes. 1-h: "one hot encoding"

3. APPLICATION TO DRILLING

The drilling process is sketched below. Measured quantities are denoted with the superscript: m and the faults given in red font. Drilling fluid ("mud") is pumped through the drill string towards the drill bit, and then returns through the annulus into the Fluid Handling System (FHS), where it is cleaned and circulated back into the well. The pump rate q_p is the input to the process ($u = q_p$), resulting in a pump pressure p_p and return flow q_r as outputs ($x = [p_p, q_r]$). The relationship between the input and output depends on whether a fault has occurred. The FHS is assumed open to the atmosphere on the annulus side, so the pressure at that boundary is 1 bar. Managed Pressure Drilling (MPD) can easily be accommodated provided pressure at the inlet of the MPD choke and flow rate of the backpressure pump are measured.



A **Washout** happens when the flow shortcuts from the drillstring to the annulus due to a crack or hole in the string. Its diagnostics are crack location and size, denoted as z_{WO} and C_{WO}. **Mud loss** occurs when there is a leakage of mud from the well into the reservoir. The diagnostics of a mud loss are reservoir pressure and a so-called production index, denoted as p_r and k_I.

Pack-off is a partial or complete blocking of the recirculation flow. It is related to insufficient cleaning so that cuttings remain in the well. The diagnostic of a pack-off is the pressure drop across it. More details on the formulation of the Faults is given in the Appendix.

All diagnostics are normalized in the interval $[0,1]$. The relationships between input and output subject to these three types of faults are given in Appendix A. Here, we will simply refer to them as f_1, f_2, f_3 for Washout, Mud Loss and Pack-Off, respectively. The diagnostics of the faults can be determined given measurements from *at least* two different inputs. We select three different inputs, that is $M = 3$. Therefore, the input to the PINN is $[x_1, x_2, x_3] = [p_p^{(1)}, q_r^{(1)}, p_p^{(2)}, q_r^{(2)}, p_p^{(3)}, q_r^{(3)}]^T$obtained for $[u_1, u_2, u_3] = [q_p^{(1)}, q_p^{(2)}, q_p^{(3)}]^T = [0.6, 1, 1.4]^T \cdot 10^3$ l/min. Correspondingly, the outputs of the PINN are $d_1 \in [0,1]^2, d_2 \in [0,1]^2, d_3 \in [0,1]$, which are the normalized diagnostics for the three individual type of faults. These definitions are in line with the network structure shown in Figure 1-a. The type of fault is easily identified during drilling ahead, motivating for using network structure of Figure 1-a rather than b-c. For mud loss, one will observe that the return flow decreases and becomes less than the pump flow. In the event of a wash out, the return flow remains the same while the pump pressure decreases, and for pack-off, the return flow remains the same while the pump pressure increases.

We used a fully connected feedforward NN which was trained according to the parameters in Table 1. Since real data for testing is not available, synthetic input data x_τ was generated using the models (1) with diagnostics d_τ drawn uniformly from $\mathcal{D}_\tau$. The predictions $\hat{d}_\tau = f_o^\tau(f_{sh}(x_\tau))$ provided by evaluating the PINN were subsequently compared with d_τ. In other words, the testing used noise-free data generated from the same models as those used for training.

Table 1. Training parameters

Parameter	**Value**
Batch type	Full batch
Number of datapoints	{`Normal`: 1, `Washout`: 100, `Packoff`: 100, `Mud loss`: 100}
NN structure	`[6, 50, 45, 40, 35, 30, 25, 8]`
Activation functions	`GELU`
Loss function	MSE
MTL training strategies tested	*Cosine-Regularization* (Suteu and Guo (2019)), *Gradient Normalization* (Chen et al. (2017))
Hardware	NVIDIA RTX A2000 Laptop GPU (cuda) with PyTorch

Enriching data through transformation: The Mudloss training dataset required transformation, because the original data was not rich enough and the optimizer would get trapped into local minima of high training error, even when trained without the other tasks (Figure 2). Given a sufficiently dense training dataset, the samples can be transformed using interpolated data from a data-grid of the transformed data, and then used as inputs for training the NN. This renders inference non-trivial, since it is important to know which Fault is active (Detection) and

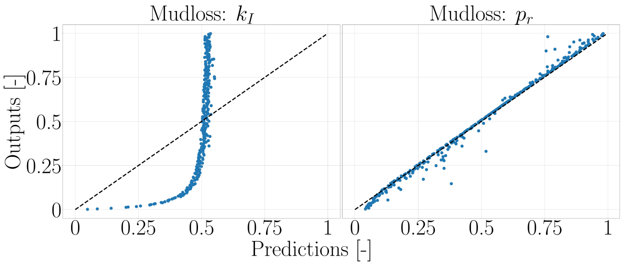


Fig. 2. Mud loss scatterplots for randomly generated (simulated) validation data. Notice the k_I predictions after training (without the other tasks) without the transformation.

that the training data was sufficiently rich in order to create a representative data grid for interpolation.

The Mud-loss data was numerically transformed through the derivative of the function that computes the inputs. The transformation type is described by: $f'_{new}(x) = G(f'(x))$. Numerous forms for the function G were attempted, and simply by setting $G(\cdot) = \sqrt{\cdot}$, the curvature of the data would become "richer", thus making the learning problem better conditioned. The 3D curves of the transformation are depicted in Figure 3. Specifically, the dataset $(k_I, p_r, p_p(k_I, p_r))$ (Appendix A.2) was transformed by conditioning the new function (p_p^{tr}): $\partial p_p^{tr}/\partial k_I = \sqrt{|\partial p_p/\partial k_I|}$. Through (trapezoidal) integration, we obtained the transformed data-grid: $(k_I, p_r, p_p^{tr}(k_I, p_r))$. Then, to generate the training data, we simply used this data-grid and performed linear interpolation. Admittedly, this introduces the need for a dense dataset, such that the interpolation result is sufficiently close to the correct function.

Gradient manipulation-based training strategies for MTL. We explored popular MTL training algorithms in order to increase the accuracy and training speed of the NN. Since (up to our knowledge) this work constitutes the first attempt to perform MTL for Fault Diagnosis in Drilling, it is a natural question to pursue this goal. The MTL training strategies utilized for this work can be found in Table 1.

4. RESULTS AND DISCUSSION

Using the training parameters from Table 1, we have successfully trained a DNN to perform Fault Diagnosis. We utilized state-of-the-art MTL training strategies such as the *Cosine Regularization algorithm (CosReg)* (Suteu and Guo (2019)) and *Gradient Normalization* (*GradNorm*) (Chen et al. (2017)). The list below enumerates key observations from our simulations.

(1) **DNN, AdamW, exponential Learning Rate**. High accuracy result, both in training dataset and randomly generated validation sets.
(2) **DNN, AdamW, exponential Learning Rate + CosReg**. Result slightly better than in 1. However, the improvement does not justify the addition of the α hyperparameter and the increase in training time.
(3) **DNN, AdamW, exponential Learning Rate + GradNorm**. Algorithm does not converge to a satisfactory solution. Adjusting the other NN hyperparameters was not tested, since results were already satisfactory without further tuning.
(4) **DNN, AdamHD, exponential Learning Rate**. AdamHD often leads to a lower training error, as it adapts to the current training speed. However, this approach increases training time due to an additional backpropagation step. The training error dynamics are highly sensitive to the tuning of the β hyperparameter (Baydin et al. (2017)), particularly with ReLU-like activation functions, requiring a small β value. This resulted in slower training than with the standard exponential learning rate.

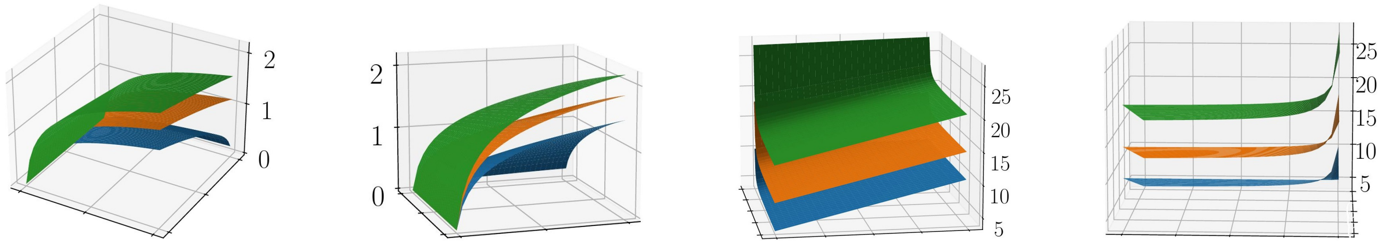
As can be seen in Figure 4 at the subplots corresponding to the case of transformed Mud Loss data (the accompanying colorbars represent the C_{WO} values), the prediction accuracy is particularly high (the points in the scatterplot are closely approaching the 1-1 line), with a few expected and acceptable deviations. Localization becomes challenging for small values of C_{WO} since z_{WO} becomes highly sensitive w.r.t. C_{WO} in those value regions. In a sense, the uncertainty/variance in the location estimate increases as it tends towards the case of no washout (alternatively zero pump flow rate) where the location probability distribution is completely uniform. Utilization of CosReg only slightly improved the prediction accuracy for each task, presumably due to the fact that the task weights are naturally tending towards orthogonalization.

Effect of transformation. The transformation function led to a faster and more accurate learning for the Mud-loss fault compared to the other two Faults. However, including the Mud-loss fault into the learning problem leads to slower learning and slightly less accurate results for the other two tasks compared to when training them without including Mud-loss, especially for the Washout task (Figure 4). The poor results from Figure 2 can be somewhat surprising at first glance, given that a NN performs nonlinear transformations of the input data. On the other hand, it is widely known that certain pre-training transformations are important and can result in performance improvements (Shi (2000)).

5. CONCLUSIONS AND FUTURE WORK

Through this work, we have developed a PINN-MTL that is able to perform accurate diagnosis of each fault by only utilizing a single NN with a modification of the typical MTL architecture. It provided the first step towards using singular NNs that can cover a broader spectrum of faults, and in future work, problem-specific parameters. The lack of significant improvements in accuracy, training time, or network parameter efficiency through the utilization of task-gradient related MTL training methods compared to vanilla MTL training was surprising, given the documented benefits and thorough tests of such methods in the literature.

A natural extension of this work would be to include data generated by different well characteristics (such as friction characteristics and drillstring lengths) and develop an appropriate NN architecture. Since PINNs are powerful when it comes to learning from time series, a natural extension of the current work would be to incorporate time-series



(a) Transformed: angle 1 (b) Transformed: angle 2 (c) Untransformed: angle 1 (d) Untransformed: angle 2

Fig. 3. $q_p = 600$ **l/min**, $q_p = 1000$ **l/min**, $q_p = 1400$ **l/min**. Mud-loss dataset transformation. The x and y axes represent the diagnostic variables of mud-loss and the z-axis the output of p_p (see Appendix A.2). Notice how the curvature in the transformed case is "richer".

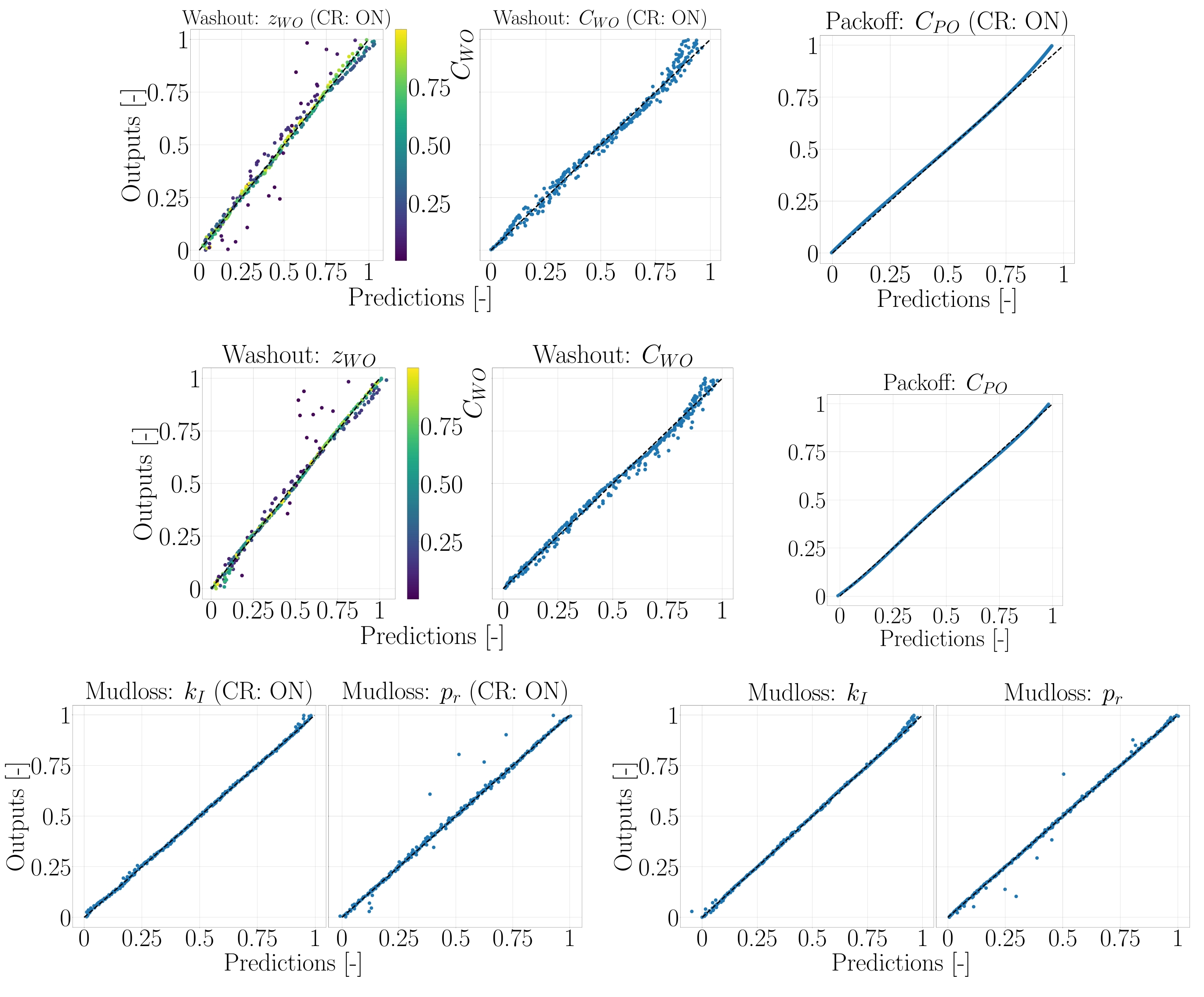


Fig. 4. Scatterplots for randomly generated data. The separate figures pertain to diagnostic variables (see Appendix for their meaning), and "CR: ON" means that the CosReg algorithm was used. All the diagnostic variables are scaled to be in [0,1].

data, which can help eliminate the need of switching to a different flow operation point, thus further automating and accelerating the process. To render the developed algorithm more practical, elaborate testing using state-of-the-art drilling simulators can be carried out. In addition, the success of the transformation applied in the Mud Loss data leads to the natural question; *"can we train a NN to learn how to optimally transform the input data?"*. Up to our knowledge, there is no existing work applying this idea. Injecting noise is an interesting challenge, since the algorithm would have to avoid random noise amplification. Finally, it became evident that more efficient learning rate schedulers can significantly reduce the training time, rendering this direction a reasonable continuation of the current work.

Appendix A. DRILLING MODEL EQUATIONS

The pressure losses in the system are

$$\begin{aligned} \Delta p_{pipe}(l,q) &= l(a_p \cdot q^2 + b_p \cdot q + c_p) \\ \Delta p_{anl}(l,q) &= l(a_a \cdot q^2 + b_a \cdot q + c_a) \\ \Delta p_{bit}(q) &= \rho\,(q/c_{bit})^2 \end{aligned} \tag{A.1}$$

where l is length, q is flow, ρ is density of the mud, and c_{bit} characterizes the bit. The coefficients related to viscous drag, $a_p, b_p, c_p, a_a, b_a, c_a$, are assumed known but could potentially be learned by the network based on measurements from the process. Next, we provide the models for the three types of faults.

A.1 Washout

We model the washout as

$$q_{WO} = C_{WO}\sqrt{\Delta p_z/\rho} \tag{A.2}$$

where C_{WO} is a constant characterizing the crack or hole in the pipe, and Δp_z is the pressure difference between the pipe and annulus at the location $z_{WO} \in (0, L)$ of the washout. That is,

$$\begin{aligned} \Delta p_z = \Delta p_{pipe}(q_p - q_{WO}, L - z_{WO}) - \Delta p_{bit}(q_p - q_{WO}) \\ - \Delta p_{ann}(q_p - q_{WO}, L - z_{WO}). \end{aligned} \tag{A.3}$$

The pump pressure is given by

$$p_p = p_{atm} + \Delta p_{ann}(q_p, z_{WO}) + \Delta p_z + \Delta p_{pipe}(q_p, z_{WO}), \tag{A.4}$$

and the return flow is $q_r = q_p$. The diagnostics of the washout are z and C_{WO}.

A.2 Mud loss

We model the mud loss as

$$q_{ml} = k_I(p_{bit} - p_r), \tag{A.5}$$

where p_r is the reservoir pressure, k_I is a constant related to the permeability of the reservoir, and p_{bit} is the pressure at the bottom of the well. That is

$$p_{bit} = p_{atm} + \Delta p_{ann}(q_p - q_{ml}, L). \tag{A.6}$$

The pump pressure is in this case given by

$$p_p = p_{bit} + \Delta p_{bit}(q_p) + \Delta p_{pipe}(q_p, L), \tag{A.7}$$

and the return flow is $q_r = q_p - q_{ml}$. The diagnostics of the mud loss are k_I and p_r.

A.3 Pack-off

The pressure loss across the pack-off is modeled as

$$\Delta p_{po} = \rho \cdot q_p^2/C_{PO}^2 \tag{A.8}$$

where C_{PO} is a constant characterizing the restriction in the annulus due to the pack-off. The pump pressure is in this case given by

$$\begin{aligned} p_p = p_{atm} + \Delta p_{ann}(q_p, L) + \Delta p_{po} \\ + \Delta p_{bit}(q_p) + \Delta p_{pipe}(q_p, L), \end{aligned} \tag{A.9}$$

and the return flow is $q_r = q_p$. The diagnostic of the pack-off is C_{PO}.

REFERENCES

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J.I., Fadhel, M.A., Al-Amidie, M., and Farhan, L. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8.

Amyar, A., Modzelewski, R., and Ruan, S. (2020). Multi-task deep learning based ct imaging analysis for covid-19: Classification and segmentation. *medRxiv*.

Baydin, A.G., Cornish, R., Martínez-Rubio, D., Schmidt, M.W., and Wood, F.D. (2017). Online learning rate adaptation with hypergradient descent. *ArXiv*, abs/1703.04782.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28, 41–75.

Chen, Z., Badrinarayanan, V., Lee, C.Y., and Rabinovich, A. (2017). GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. *ArXiv*, abs/1711.02257.

Isermann, R. (2006). Fault-diagnosis systems : an introduction from fault detection to fault tolerance.

Jan, A., Mahfoudh, F., Draškovic, G., Jeong, C., and Yu, Y. (2022). Multitasking physics-informed neural network for drillstring washout detection. 2022(1).

Jeong, C., Yu, Y., Mansour, D., Vesslinov, V., and Meehan, R. (2020). A physics model embedded hybrid deep neural network for drillstring washout detection.

Jiang, H., hui Liu, G., Li, J., Zhang, T., and Wang, C. (2020). A realtime drilling risks monitoring method integrating wellbore hydraulics model and streaming-data-driven model parameter inversion algorithm. *Journal of Natural Gas Science and Engineering*, 103702.

Qiu, S., Cui, X., Ping, Z., Shan, N., Li, Z., Bao, X., and Xu, X.Y. (2023). Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review. *Sensors (Basel, Switzerland)*, 23.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378, 686–707.

Shi, J.J. (2000). Reducing prediction error by transforming input data for neural networks. *Journal of Computing in Civil Engineering*, 14, 109–116.

Suteu, M. and Guo, Y. (2019). Regularizing deep multi-task networks using orthogonal gradients. *ArXiv*, abs/1912.06844.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., and Yin, K.K. (2003). A review of process fault detection and diagnosis: Part iii: Process history based methods. *Comput. Chem. Eng.*, 27, 327–346.

Wang, H., Liu, Z., Peng, D., Yang, M., and Qin, Y. (2021). Feature-level attention-guided multitask cnn for fault diagnosis and working conditions identification of rolling bearing. *IEEE Trans. Neural Netw. Learn. Syst.*

Willersrud, A., Blanke, M., Imsland, L.S., and Pavlov, A.K. (2015). Fault diagnosis of downhole drilling incidents using adaptive observers and statistical change detection. *Journal of Process Control*, 30, 90–103.

Zhang, Q. (2000). A new residual generation and evaluation method for detection and isolation of faults in non-linear systems. *International Journal of Adaptive Control and Signal Processing*, 14, 759–773.