A Method to Evaluate the Performance of Predictors in Cyber-Physical Systems

Leonardo Passig Horstmann*
Software/Hardware Integration Lab,
Federal University of Santa Catarina
Florianopolis, Santa Catarina, Brazil
horstmann@lisha.ufsc.br

Matheus Wagner*
Software/Hardware Integration Lab,
Federal University of Santa Catarina
Florianopolis, Santa Catarina, Brazil
wagner@lisha.ufsc.br

Antônio Augusto Fröhlich
Software/Hardware Integration Lab,
Federal University of Santa Catarina
Florianopolis, Santa Catarina, Brazil
guto@lisha.ufsc.br

ABSTRACT

Cyber-Physical Systems (CPS) rely on sensing to control and optimize their operation. Nevertheless, sensing itself is prone to errors that can originate at several stages, from sampling to communication. In this context, several systems adopt multivariate predictors to assess the quality of the sensed data, to replace data from faulty sensors, or to derive variables that cannot be directly sensed. These predictors are often evaluated based on their accuracy and computing demands, however, such evaluations often do not consider the system's architecture from a broader perspective, ignoring the way components are interconnected and how they cascade as inputs of other Machine Learning (ML) models. In this work, we introduce a method to evaluate the performance of interdependent predictors based on the stability of the estimation error dynamics in faulty scenarios. The proposed method estimates the ability of a predictor to produce accurate predictions while accounting for the impacts of cascading predicted values as its inputs. The prediction correctness is estimated based solely on information acquired during the training of the multivariate predictors and mathematical properties of the ML activation functions. The proposed method is evaluated with a meaningful dataset in the scope of monitoring and control of a Cyber-Physical System, and the evaluation demonstrates the ability of the proposed method to account for the interdependence of data predictors.

CCS CONCEPTS

• **Mathematics of computing → Functional analysis**; • **Information systems →** *Data mining*; • **Computing methodologies** → *Machine learning*; **Machine learning algorithms**; • **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

stability analysis, predictors, machine learning, estimation error dynamics

*Both authors contributed equally to this research.

ICPE '23, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0068-2/23/04…$15.00
https://doi.org/10.1145/3578244.3583732

ACM Reference Format:
Leonardo Passig Horstmann, Matheus Wagner, and Antônio Augusto Fröhlich. 2023. A Method to Evaluate the Performance of Predictors in Cyber-Physical Systems. In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3578244.3583732

1 INTRODUCTION

Cyber-Physical Systems (CPSs) can be seen as a composition of software and hardware components in which the outputs produced by some are used as inputs by others. Some components are modeled around sensors, some derive variables from sensors and other inputs, and, in fact, the whole composition is usually very sensitive to sensed values. Notwithstanding, sensing operations are prone to errors ranging from incorrect sampling (e.g., gain, stuck-at, spike, and noise errors) to communication problems. Therefore, CPSs often use predictors to assess the quality of the sensed data, derive variables that cannot be directly sensed, and to replace data from faulty sensors.

In this context, works like Richman et al. [13] have shown multivariate predictors using Machine Learning (ML) mechanisms like Artificial Neural Networks (ANNs) and Support Vector Regression (SVR) demonstrate higher prediction accuracy as they are able to better mimic data variance aspects. [14, 15] proposed a semi-supervised multivariate ANN to cope with sensing faults. These predictors are usually evaluated for accuracy, with Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), and computing power demands. However, as pointed by Yang et al. [18], traditional methods for evaluation of predictors usually consider Independent and Identically Distributed samplings, do not account for the different impacts of prediction errors and misclassifications, and do not take into consideration specificities of the domain or real-world applications. For instance, in the CPS domain, traditional evaluation techniques usually do not consider the system's architecture from a broader perspective, ignoring the way components are integrated and the resulting topologies, which directly impact the quality of predictors as they take other predictions as input.

In this work, we introduce a methodology to evaluate predictors based on their ability to produce accurate predictions while accounting for the impacts of adopting predicted values as their input. First, ML multivariate models are built for data prediction considering data correlation. To do so, a fault-free period of the time-series is selected and feature selection algorithms are applied to determine the subset of most correlated features for each of the variables. Next, an ANN is trained to predict a variable based on the input composed of its most correlated features. Finally, the prediction expected correctness when dealing with inputs composed of predicted data is then evaluated as the error propagation between predictors. The evaluation method is based solely on information acquired during the predictor's training and mathematical properties of the ML activation functions[1]. Furthermore, the presented method provides one with the ability to assess, at design time, whether the prediction errors would remain bounded when dealing with scenarios where predicted data is used as input for new predictions. The main contributions of this work are:

- A mathematical analysis of the error propagation for interdependent predictors.
- A method to evaluate data predictors based on the impacts of the utilization of predicted inputs and successive predictions.
- A method that allows design-time assessment of the boundedness of prediction error when using inputs generated by predictors.

The remaining of this paper is organized as follows. Section 2 presents correlated works and discusses the main differences when compared to the approach presented here. Section 3 describes the method proposed to evaluate the performance of data predictors based on their interdependence. Section 4 presents our case study and shows the results achieved through the proposed method. Finally, Section 5 discusses the proposed method in light of the presented results and finishes this paper presenting its final remarks.

2 RELATED WORKS

As pointed by Yang et al. [18], traditional methods for evaluation of predictors usually consider Independent and Identically Distributed samplings, do not account for the different impacts of prediction errors and misclassifications, and do not take into consideration specificities of the domain or real-world applications. In this section we review works that propose different mechanisms to evaluate predictive models.

Ghobbar and Friend [5] adopted an Analysis of Variance to explain relationship between variation in the input variables (factors) with the variations in the predicted variable (response variable). They used a general linear model consisting of the extension of simple and multiple regression to measure the impact of the variation of each factor in the response variable. In this way, they estimate prediction accuracy for any set of factors. The best prediction mechanism would be the one with the lowest impact of the inputs (i.e., lowest factors). Besides using regression to estimate the impacts of the variation of the input variables on the prediction accuracy, the solution they propose does not account for the utilization of predicted data and their inherent prediction error impact on the input of other prediction mechanism, not considering the possible error propagation.

Nicolis et al. [10] derived properties of the evolution of meteorology prediction errors under the effect of initial errors (i.e., variations to the initial assumptions of the predictor) and model natural errors (i.e., deviations in terms of the representation of the nature). In their analysis, they bounded prediction errors for specific scenarios. The solution proposed in this paper, on the other hand, focuses on errors introduced and propagated by the utilization of predicted values as inputs for data prediction. Schillaci et al. [16] analyzed the prediction error dynamics to enhance the learning performance of ML models. They track two types of error. The first one, a general approach based on the MSE of the forward model calculated on a test dataset and the second one based on the distance between the goal and the predicted sensory state estimated by the forward model. While the Schillaci et al. work focuses on the prediction error dynamics, it does consider a traditional metric in the evaluation, namely MSE, and does not track the impact of the propagated prediction error over other predictors.

Alessandri et al. [1] presented an stability analysis for prediction error dynamics based on the sensitivity of a system to the variation of data to demonstrate the error introduced by a Moving Horizon Estimation (MHE) is bounded. Liu and Wang [8] stochastically analyzed the estimation error of a MHE with binary encoding schemes to ensure its boundedness. Similarly, Karg and Lucia [7] analyzed the impact that the approximation error of different learned components on the closed-loop control performance to evaluate a MHE. The authors approximated the resulting changes in the successor state of the controller with respect to the deviations of optimal input by considering a first order Taylor approximation. Similar the approach presented in this paper, authors also adopted Taylor series approximation to estimate impact caused by a deviation in optimal input. However, we differ from both works by considering scenarios with interdependent predictors, taking into consideration expected prediction error as the optimal input deviation, and accounting for error propagation.

Finally, Cao and Hovakimyan [3] demonstrated the boundedness of the error for a Neural Network (NN)-based adaptive control system analytically by limiting the adaptation gain of the NN model to a constant value. Similarly, Chen et al. [4] presented an analytic analysis of a Fuzzy NN-based adaptive control algorithm to demonstrate boundedness of the Fuzzy NN approximation error by using Lyapunov analysis. Besides analyzing the boundedness of a NN solution, these works also do not analyze the specific case of interdependent NN models and the error propagated by adopting the approximations produced by an NN as input of another model.

On the other hand, considering the problem of evaluating the performance of CPSs, Muttillo et al. [9] presented a framework using Machine Learning to estimate the timing performance of CPS based on code analysis. Smith [17] analyzed software patterns that deteriorated the performance of CPS and proposed less intrusive alternatives. Postema et al. [12] studied the interaction between power management strategies and thermal-aware controllers. These works differ from our approach since they focus on the timing and thermal aspects of the performance of CPS while we focused on the data quality of the CPS. In fact, Arlitt et al. [2] also focused on the data aspect of a CPS, however, they proposed a benchmark to represent IoT use cases while we proposed a mechanism to evaluate the quality of the predictors that are used to replace or generate new data for the CPS they compose.

3 THE METHOD

The solution proposed in this paper evaluates multivariate predictors in terms of its ability to deal with input data composed of

[1]The code and data produced for the experiments presented in this paper are available at https://gitlab.lisha.ufsc.br/matheuswgr/icpe-predictor-performance.

predicted values. In other words, this work proposes a method to estimate the impact of using predicted values as input for a multivariate predictor. In this way, it is important to notice that the proposed method does not concern the design of high-performance predictors, but their robustness in terms of being able to provide bounded predictions when dealing with interdependence and having predicted data as inputs. In this way, the study of the behavior of a predictor operating under normal conditions (i.e., no predicted data as input) and the optimization of a predictor are not in the scope of this work. In the following subsections, we recapitulate the main concepts and assumptions behind the predictor's model and present the method proposed to estimate predictor's interdependence.

3.1 Predictor's Model

A myriad of strategies to build data predictors are proposed in the literature. The simplest methods use linear interpolation, use the dataset mean value, or even the last observed value (called zero order hold). Another simple way to perform interpolation is to adopt non-linear interpolation methods, using polynomial, spline, and Autoregressive Integrated Moving Average (ARIMA) algorithms. Finally, there are complex alternatives like to use a multivariate ANN Predictor. Using a multivariate predictor based on an ANN produces more accurate values compared with the other methods. This happens mostly because linear solutions are not able to reproduce variability. Second-order interpolation, and other similar strategies, are not able to simulate the behavior of a variable that presents inherent dependency to other components of a multivariate system. Similarly, forecasting methods like ARIMA are penalized because they use the model that was trained before the first prediction to predict the whole time-series, not considering new data or predicted values for the successive predictions. In fact, Richman et al. [13] demonstrated that simple prediction mechanisms underestimate the actual variance of data, while ANN and other multivariate solutions presented higher ability to capture this variance.

In light of the higher accuracy presented by multivariate predictor's and the increasing usage of ML-based applications, this work assumes data predictors to be ANNs with multivariate inputs. To build a predictive model for each variable, first we select a subset of variables of interest that can be used as the predictor's input, using some feature selection method. With this subset of variables, the models should be trained using moments of the time-series with correct sampling. After the training process, a predictor should be available for each of the sensed variables.

Finally, a predictor is a function $\hat{v}_i = g_i(\vec{x}_i)$ that maps a vector of inputs $\vec{x}_i$ to a scalar quantity $\hat{v}_i$ that represents the prediction of the i^{th} variable in a set. In this work, $g_i(\vec{x}_i)$ is assumed to be infinitely differentiable, so it can be expanded into a Taylor series representation[2]. This assumption holds, for example, if the predictor is based on ANNs with infinitely differentiable activation functions, for instance, hyperbolic tangent or sigmoid.

[2]Several of the most-used activation functions, including Sigmoid, TanH, Softmax, Swish, and CoLU are infinitely differentiable. Therefore, the assumption makes the proposal suitable for several applications. The proposed solution does not apply to ANNs built with other (not continuously differentiable) activation functions.

3.2 Evaluating Interdependent Predictor's

We can estimate the impacts of dealing with data points coming from other predictors by calculating a metric for the expected correctness for each prediction. To measure and track correctness of a prediction, we first need an estimate of the prediction error. This estimate can be built based on the expected error of the predictor and on the deviation of the inputs from their respective predictions. We assume predictor's expected error as the MAE computed over the training dataset. The deviations of the input variables from their respective predictors can be propagated to the current predictor's output using a linear approximation.

Given a predictor $\hat{v}_i = g_i(\vec{x}_i)$, there is a Taylor series expansion, as described by equation (1), centered at $\hat{v_i}^* = g_i(\vec{x}_i^*)$, that denotes the predictor output when the input values match exactly their predicted values (this condition is denoted by $\vec{x}_i^*$). The operator $\frac{\partial g_i(\vec{x}_i)}{\partial \vec{x}_i}$ is the gradient of the function g_i with respect to it's input vector.

$$\hat{v}_i = \hat{v_i}^* + \frac{\partial g_i(\vec{x}_i^*)}{\partial \vec{x}_i}(\vec{x}_i - \vec{x}_i^*) + \frac{1}{2}\frac{\partial^2 g_i(\vec{x}_i^*)}{\partial \vec{x}_i^2}(\vec{x}_i - \vec{x}_i^*)^2 + \ldots \quad (1)$$

When the deviation of the predictor's input from $\vec{x}_i^*$ is small or when $g_i(\vec{x}_i)$ resembles a linear function, the terms with order greater than one may be negligible. In this scenario, a truncation of the Taylor series at the first-order term provides a linear approximation of the model. Note that highly correlated variables are linearly related. Therefore, using correlation as a metric to select input variables leads to models that should preserve this linear relationship.

Based on this correlation between variables, we consider solely the first-order term of equation (1) and rewrite it as equation (2). One can relate the deviation of a prediction from a most accurate prediction to the deviation of the predictor's inputs from those that yield the maximum accuracy of their respective predictors. It characterizes a measure of the prediction's correctness, even in the absence of some data in the input to be compared with.

$$\hat{v}_i - \hat{v_i}^* = \frac{\partial g_i(\vec{x}_i^*)}{\partial \vec{x}_i}(\vec{x}_i - \vec{x}_i^*) \quad (2)$$

In compact form, equation (2) is rewritten again in equation (3), in which the inner product is expressed as a summation over the N components of the input vector. The terms $x_{i,j}$ used in the partial derivatives stand for the j^{th} component of the input vector to the predictor of the i^{th} variable.

$$\hat{e}_i = \sum_{j=0}^{N-1} \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} e_j \quad (3)$$

In equation (3), $\hat{e}_i$ is the estimated error for the predictor of the i^{th} variable computed based only on the individual deviations of its inputs from their respective predictions, expressed as e_j.

Once the predictor gradient with respect to its inputs is a function of $\vec{x}_i^*$ (the prediction of each input variable), in case all the necessary information to make a prediction is available, all the information necessary to estimate its error is also available.

The error estimation in equation (3) can be enhanced by adding the predictor's expected error, leading to equation (4). By doing so, we assume that even if the input errors are equal to zero, the predicted output error is at least equal to the *MAE* obtained while training the model.

$$\hat{e}_i = \sum_{j=0}^{N-1} \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} e_j + MAE_i \tag{4}$$

3.3 Stability of Error Estimation Chains

The prediction method established by equation (4) assumes that the deviations between inputs and their respective predictions can be measured. However, input data for the predictors could also consist of predicted values and hence their deviations must also be estimated, forming an error estimation chain.

This work considers three configurations of error estimation chains: ***cascade***, ***feedback***, and ***loop***, presented in Figures 1, 2, and 3, respectively. In *cascade*, the output of a predictor is used as one of the inputs to another predictor. In *feedback*, the output of a predictor is used as one of the inputs to itself. In *loop*, the output of a predictor is used as one of the inputs to another predictor whose output is, in turn, used as one of the inputs to the first predictor.

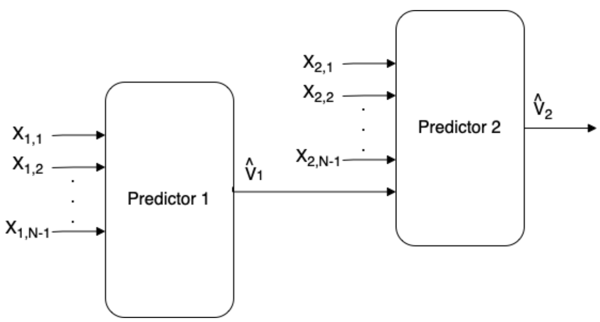

Figure 1: Two predictors associated in a cascade configuration.

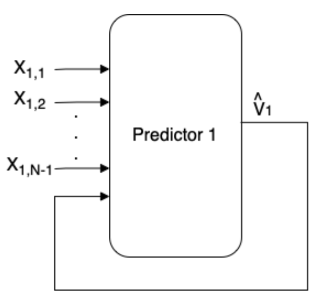

Figure 2: Single predictor in a feedback configuration due to self dependence.

For the cascade configuration, the error made by the first predictor propagates to the output of the second predictor, whose prediction will also have an associated error. Given that the expected error of each predictor is finite and that the gradient of each predictor's output with respect to its input is bounded, any finite error associated with any input will produce a finite propagated error to the predictor's output. Hence, any chain in cascade configuration with a finite number of predictors will produce a finite propagated error. Note that common activation functions, such as hyperbolic tangent and sigmoid, have bounded derivatives, hence, ANNs with bounded gradients are viable.

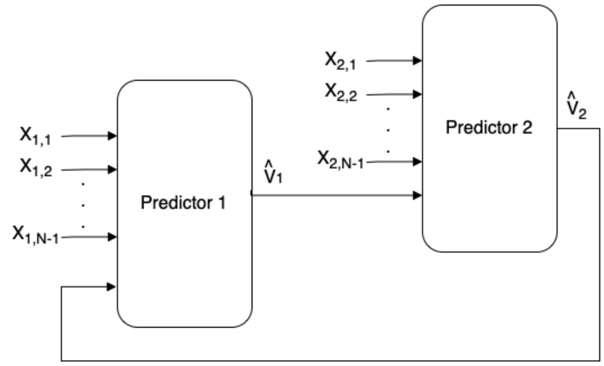

Figure 3: Two predictors in a loop configuration due to predictor's interdependence.

In feedback configurations, the error made by the predictor becomes the error associated with one of its inputs, which is then propagated to a next output, producing a cycle. In loop configurations the error made by the first predictor is propagated to the output of the second predictor whose error is propagated to the output of the first predictor, also producing a cycle. For these cases, even if predictor's expected error is finite and the gradient of each predictor's output with respect to its input is bounded, the propagated error through the cycle can become infinite. Since the problem can be modeled as a recurrence, stability analysis, a tool borrowed from the theory of linear systems, can be employed to establish the conditions in which the accumulated prediction errors do not become infinite. The application of stability analysis to discrete-time linear systems modeled as recurrences is concerned with determining whether the evolution of a given recurrence converges to a finite value. The reader may refer to [11] for a deeper discussion of the topic.

At this point, it is relevant to state that although higher-order Taylor series approximations of $g_i(\vec{x}_i)$ could be used to estimate prediction errors, a linear approximation is convenient as it allows the stability analysis to be performed using tools from the linear systems theory. Initially, equation (4) must be rewritten as a linear time-invariant system. Although it already represents a linear system, it can be considered as time-variant because the predictor gradient changes over time. By assuming that the components of the predicted gradient are bounded in absolute value, a worst-case evaluation of the error's linear approximation can be built as in equations (5) and (6). As the partial derivatives and the inputs' errors can assume positive and negative values, the sum of their absolute values will be greater or equal than the sum of their regular values, as stated in equation (5). By taking the maximum value of the predictor gradient, we ensure that we are dealing with the worst case.

$$\|\hat{e}_i\| = \left\| \sum_{j=0}^{N-1} \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} e_j \right\| \leq \sum_{j=0}^{N-1} \left\| \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} \right\| \|e_j\| \tag{5}$$

$$\|\hat{e}_i\| \leq \sum_{j=0}^{N-1} \left\| \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} \right\| \|e_j\| \leq \sum_{j=0}^{N-1} max \left\{ \left\| \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} \right\| \right\} \|e_j\| \tag{6}$$

Equation (7) renames the maximum absolute value of the gradient components in order to keep the notation clean. λ_i^j stands for the j^{th} component of the predictor's maximum gradient in absolute value. Note that the maximum gradient components can be computed analytically or numerically by an optimization algorithm. It can also be estimated by selecting the maximum gradient evaluated over the training dataset.

$$\lambda_i^j = max \left\{ \left\| \frac{\partial g_i(\vec{x}_i^*)}{\partial x_{i,j}} \right\| \right\} \tag{7}$$

By replacing equation (7) into equation (6) and adding the predictor's expected error to the right side of the inequality in equation (6), a bound for the error estimation can be defined as equation 8.

$$\|\hat{e}_i\| \leq \sum_{j=0}^{N-1} \lambda_i^j \|e_j\| + MAE_i \tag{8}$$

The estimation error's bound will be expressed by ξ_i, defined by equation (9). To explicitly account for time in the equation, the letter k is used to represent the k^{th} instant of discrete time, hence a continuous time sample is given by $t = kT$, for a given period T. The term $e_j[k-1]$ expresses the fact that the error associated with an input is computed based on information from the previous time period.

$$\xi_i[k] = \sum_{j=0}^{N-1} \lambda_i^j \|e_j[k-1]\| + MAE_i \tag{9}$$

A system with all error estimate chains simultaneously is considered for the stability analysis. Expressing equation (9) in terms of a linear system that captures the effects of all three error prediction chain configurations demands that the input variables are separated into three sets. The first is the set I_i of input variables that are not in the feedback or loop chains. The second is the set D_i of input variables that comprise a loop chain. The third set is given by the output of the i^{th} predictor, since this is the variable that composes the feedback loop. This is denoted by $\lambda_i^r \xi_i[k-1]$ in equation (10) that describes this scenario mathematically.

$$\xi_i[k] = \lambda_i^r \xi_i[k-1] + \sum_{j \in D_i} \lambda_i^j \left\| \hat{e}_j[k-1] \right\| + \sum_{p \in I_i} \lambda_i^p \left\| \hat{e}_p[k-1] \right\| + MAE_i \tag{10}$$

Stability in the Bounded-Input Bounded-Output (BIBO) sense is a property of the natural response of a linear system. Hence, all terms that do not depend on $\xi_i[k-n]$ for any value of n can be dropped from equation (10), including MAE_i, leading to equation (11).

$$\xi_i[k] = \lambda_i^r \xi_i[k-1] + \sum_{j \in D_i} \lambda_i^j \left\| \hat{e}_j[k-1] \right\| \tag{11}$$

By realizing that $\left\| \hat{e}_j[k-1] \right\|$ respects an inequation as (8), this term can be defined as in the equation. The term $\left\| \hat{e}_j[k-1] \right\|$ can be replaced in equation (11) by $\lambda_j^i \xi_i[k-2]$ since all other terms in equation (12) that are not a function of $\xi_i[k-n]$ do not contribute to the natural response of $\xi_i[k]$.

$$\|\hat{e}_j[k-1]\| \leq \lambda_j^i \xi_i[k-2] + \sum_{\substack{n=0 \\ n \neq i}}^{N-1} \lambda_j^n \|\hat{e}_n[k-2]\| \tag{12}$$

Finally, equation (13) describes the natural dynamics of error estimates for the i^{th} variable in the presence of both loop and feedback chains. Note that this recurrence only accepts positive initial conditions due to the definition of ξ_i.

$$\xi_i[k] = \lambda_i^r \xi_i[k-1] + \sum_{j \in D_i} \lambda_j^i \lambda_i^j \xi_i[k-2] \tag{13}$$

Stability analysis can be carried out by taking the Z-transform on both sides of equation (13) and rearranging the terms, leading to equation (14). In this new equation the polynomial in z gives the characteristic equation of the system. The error estimate remains stable in the BIBO sense if, and only if, the solutions of the characteristic equation remain on the unit circle in the complex plane. This is expressed mathematically by equation (15)

$$\left(z^2 - \lambda_i^r z - \sum_{j \in D_i} \lambda_j^i \lambda_i^j \right) \hat{E}_i(Z) = 0 \tag{14}$$

$$\|z_{1,2}\| = \frac{1}{2} \left\| \left[\lambda_i^r \pm \sqrt{(\lambda_i^r)^2 + 4 \sum_{j \in D_i} \lambda_j^i \lambda_i^j} \right] \right\| \leq 1 \tag{15}$$

While equation (15) expresses the stability condition for scenarios with both loop and feedback chains, note that in the absence of the feedback chain, the stability condition becomes the stated in equation (16). If the models do not have a loop chain, the stability condition is represented by equation (17)

$$\|z_{1,2}\| = \sqrt{\sum_{j \in D_i} \lambda_j^i \lambda_i^j} \leq 1 \tag{16}$$

$$\|z_1| = \lambda_i^r \leq 1 \tag{17}$$

According to how the variables depend on each other, higher order systems may arise. In this case, new stability conditions can be derived for any specific dependency relationship. The convergence conditions defined in equations (15) to (17) are useful to estimate how long the predictor can still produce accurate outputs. When the convergence conditions are not met, the estimated errors grow in an unbounded way, leading to a fast accuracy drop.

The error estimation needed when the predictor's input is composed by predicted values is described in algorithm 1. We assume that three global vectors are available: $\vec{X}_{star}$, that holds the predictions of all data points from the previous period; $\vec{X}_{actual}$, that holds all available data points from the previous period; and $\vec{E}$, that holds the errors – calculated or estimated – of all data points from the previous round.

At each round — time interval T in which all values are expected to arrive at least once — the arrived data is processed by algorithm 1 to update $\vec{X}_{star}$ and $\vec{E}$. When the period expires, all predicted inputs j will have their values predicted and errors estimated, updating $\vec{X}_{star}[j]$ and $\vec{E}[j]$, respectively.

The function $inputs(i, \vec{X}_{actual}, \vec{X}_{star})$ returns a vector of inputs to the predictor. If an input is produced by a predictor itself in the vector of data points $\vec{X}_{actual}$, then it is replaced by its prediction available in $\vec{X}_{star}$. The functions $input\_errors(i, \vec{E})$ and $input\_predictions(i, \vec{X}_{star})$ only take the subsets of the components of $\vec{E}$ and $\vec{X}_{star}$, respectively, that correspond to the input variables of the i^{th} predictor.

Algorithm 1 Prediction Error Update

procedure Error_Update(v, i)
    $\vec{x}_i \leftarrow inputs(i, \vec{X}_{star}, \vec{X}_{actual})$
    $\vec{E}_i \leftarrow input\_errors(i, \vec{E})$
    $\vec{x}_i^* \leftarrow input\_predictions(i, \vec{X}_{star})$
    $\hat{v} \leftarrow predict(\vec{x}_i)$
    if is_empty(v) **then**
        $grad_g \leftarrow gradient(g(\vec{x}_i^*), \vec{x}_i^*)$
        $\vec{E}[i] \leftarrow (grad_g \cdot \vec{E}_i) + MAE_i$
    else
        $\vec{E}[i] \leftarrow ||v - \hat{v}||$
    $error \leftarrow \vec{E}[i]/dataset\_mean$
    $\vec{X}_{star}[i] \leftarrow \hat{v}$

Finally, in order to clarify the process of performing the stability analysis, we modeled the algorithm 2 based on the aforementioned equations and algorithms.

Algorithm 2 Procedure for Analysis

Proceed with Feature Selection
Train the respective prediction models
Verify interdependence between predictors:
Check if the target variable of the predictor is also used as input to the predictor, this implies the existence of feedback interdependence
if the target variable of the predictor is also used as input to the predictor **then**
    this implies the existence of feedback interdependence
if the target variable of the predictor is also used as input to another predictor **then**
    this implies the existence of cascade interdependence
if the target variable of the predictor is also used as input to another predictor whose target variable is input to the first predictor **then**
    this implies the existence of loop interdependence
For each interdependence between predictors, compute the parameters $\lambda_{i,j}$, in case of loop dependence, and $\lambda_{r,i}$, in case of feedback dependence, based on the models acquired during training.
Use the stability conditions to compute ||z|| and verify whether each predictor is stable in the presence of missing data for the considered interdependence.

4 CASE STUDY

This section describes the case study used to evaluate the proposed method. We first describe the dataset and its representativeness in the scope of CPS. Next, we describe the processes of generating the predictive models, i.e., preprocessing, feature selection, and training. Finally, we present the results of the experiments using the proposed method for tracking error propagation for interdependent predictors.

4.1 Dataset

The experiments presented in this paper were conducted over a publicly available dataset for condition monitoring of a hydraulic test rig [6]. The data was collected using a test rig that allows for reversible degradation of the system's condition, aiming at providing information about different faults that can occur during the system's operation. The test rig is composed of a primary working and a secondary cooling-filtration circuit connected via the oil tank. The measurements of process values were acquired in a constant load condition that cyclically repeats. Since the goal of this work is not to detect faults in the system operation, only the portion of the dataset that does not represent faulty scenarios was used to train the model. The description of the variables monitored by the system is presented in Table 1.

Sensor	Physical Quantity	Sampling Frequency
PS1	Pressure [bar]	100 Hz
PS2	Pressure [bar]	100 Hz
PS3	Pressure [bar]	100 Hz
PS4	Pressure [bar]	100 Hz
PS5	Pressure [bar]	100 Hz
PS6	Pressure [bar]	100 Hz
MPW	Motor Power [W]	100 Hz
FS1	Volume Flow [L/min]	10 Hz
FS2	Volume Flow [L/min]	10 Hz
TS1	Temperature [$^{\circ}C$]	1 Hz
TS2	Temperature [$^{\circ}C$]	1 Hz
TS3	Temperature [$^{\circ}C$]	1 Hz
TS4	Temperature [$^{\circ}C$]	1 Hz
VS1	Vibration [mm/s]	1 Hz
CE	Cooling Efficiency [%]	1 Hz
CP	Cooling Power [W]	1 Hz
SE	Efficiency Factor [%]	1 Hz

Table 1: Description of dataset variables

4.2 Model Generation

This section describes the processes that led to the generation of the predictive models evaluated in this paper. We first normalized each variable with the Min-Max normalization (i.e., data is valued from 0 to 1, with 0 as the minimum value and 1 as the maximum observed value). The normalized data were then submitted to a feature selection process using Pearson Correlation to select the subset of the most correlated features for each one of the target variables. To do so, we first defined the number of model inputs K (total features selection). Next, the Pearson correlations between a target feature and each other features, lagged by one sample with respect to the target feature, in the dataset are computed. The K features with the greatest Pearson correlation with the target feature are selected.

To evaluate all possible scenarios of predictor interdependence, two models were developed for each target variable. One allows for the prediction of the variable to depend on its last measured value, and the other does not, namely, autoregressive and non-autoregressive, respectively. The former leads to a predictor in *feedback*, while the latter leads to a predictor in either *cascade* or *loop* configuration.

Feature Selection results for the non-autoregressive and autoregressive cases are presented in tables 3 and 2, respectively. Note that the models are built to make predictions considering the values of the previous sampling interaction. All models use the same neural network architecture, consisting of a single hidden layer with 64 neurons with a hyperbolic tangent activation function.

Target	Selected Inputs
PS1	PS1, MPW, SE, FS1
PS2	PS2, FS1, SE, PS3
PS3	PS3, FS1, PS2, SE
PS4	PS4, CE, TS2, TS4
PS5	PS5, PS6, TS3, TS4
PS6	PS6,PS5, TS3, TS4
MPW	MPW, PS1, SE, PS2
FS1	FS1, PS3, PS2, SE
FS2	FS2, TS4, TS3, TS2
TS1	TS1, TS2, TS4, TS3
TS2	TS2, TS4, TS1, TS3
TS3	TS3, TS4, TS2, TS1
TS4	TS4, TS3, TS2, TS1
VS1	VS1, FS2, CE, CP
CE	CE, TS4, CP, TS2
CP	CP, CE, TS4, TS2
SE	SE, PS2, FS1, PS1

Table 2: Selected inputs for autoregressive predictors

4.3 Results

In this section, we evaluate the performance of interdependent predictors in five scenarios accounting for the three configurations (cascade, loop, and feedback) for both stable and unstable situations. The cascade configuration is always stable. Therefore, we analyze a single scenario for this configuration.

The autoregressive models were used for the scenarios considering a feedback configuration. For all other scenarios, the non-autoregressive models were applied.

4.3.1 Scenario 1 - Cascade Configuration. In this scenario, we replace data points from the MPW sensor by their predictions and evaluate the effect of such replacement on the performance of the predictor for PS1 sensor data in terms of absolute prediction error. Since only MPW sensor data points are replaced by their predictions, this scenario corresponds to a cascade configuration, and we do not conduct the stability analysis.

In Figures 4 and 5, a comparison of the performance of the predictions with and without interdependence is presented. It is clear that the prediction error is not highly affected by the replacement

Target	Selected Inputs
PS1	MPW, SE, FS1
PS2	FS1, SE, PS3
PS3	FS1, PS2, SE
PS4	CE, TS2, TS4
PS5	PS6, TS3, TS4
PS6	PS5, TS3, TS4
MPW	PS1, SE, FS1
FS1	PS3, PS2, SE
FS2	TS4, TS3, TS2
TS1	TS2, TS4, TS3
TS2	TS4, TS1, TS3
TS3	TS4, TS2, TS1
TS4	TS3, TS2, TS1
VS1	FS2, CE, CP
CE	TS4, CP, TS2
CP	CE, TS4, TS2
SE	PS2, FS1, PS1

Table 3: Selected inputs for non-autoregressive predictors

of the inputs by predictions. Moreover, the estimated error provides a pessimistic estimation of the effects of measurement replacement, as intended.

Still regarding Figure 4, as one can notice, the predictions with no interdependence deviate from the measurements and, when there is interdependence, the waveform resembles more the original signal than otherwise. However, by taking a close look at Figure 5, one can see that in both situations the prediction errors remain roughly the same, and since the predictors are trained to minimize error, both cases lead to predictions that are equally good from the prediction error perspective.

Nevertheless, the difference between the estimated prediction error with interdependence and the actual prediction error is expected, as described by the mathematical analysis, since the estimated prediction error is an estimate of an upper bound on prediction errors in case one of the inputs of the predictor is not available.

4.3.2 Scenario 2 - Stable Loop Configuration. In this scenario, we assume that the data points from PS6 and PS5 sensors are replaced with their predictions. We evaluate the effect of such replacement on the performance of the predictor for PS5 sensor data in terms of absolute prediction error. Since PS5 sensor data serves as input to the predictor of PS6 sensor data, and PS6 sensor data, in turn, serves as input to the predictor of PS5 sensor data, this scenario stands for a loop interdependence configuration.

By evaluating the maximum gradient of both predictor's outputs, it is possible to find $\lambda_{PS5}^{PS6} = 0.70$ and $\lambda_{PS6}^{PS5} = 0.67$, which implies in $z = 0.68$, as stated in equation 16. Hence, the predictions of PS5's values are expected to remain stable.

As expected, Figures 6 and 7 show that even though the prediction error significantly increases when measurements are replaced by predictions, it remains bounded. The curve labeled "Estimated Prediction Error - With Interdepence" is computed via error propagation, while the other curves are computed as the difference between predictions and measurements. Besides the confirmation

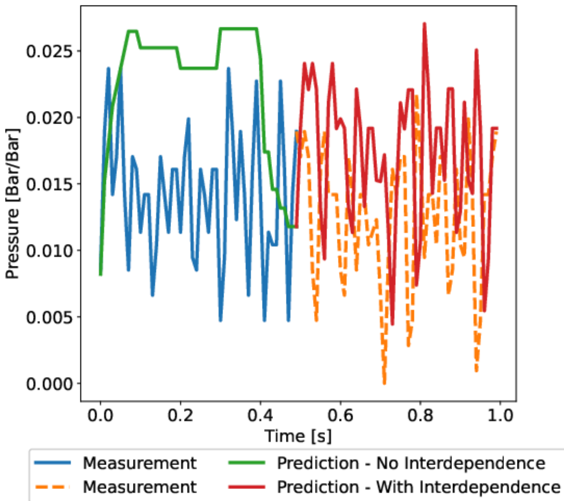

Figure 4: Measurements and predictions for the PS1 sensor before and after measurements from the MPW sensor are replaced by predictions. The measurements from MPW start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS1 and MPW.

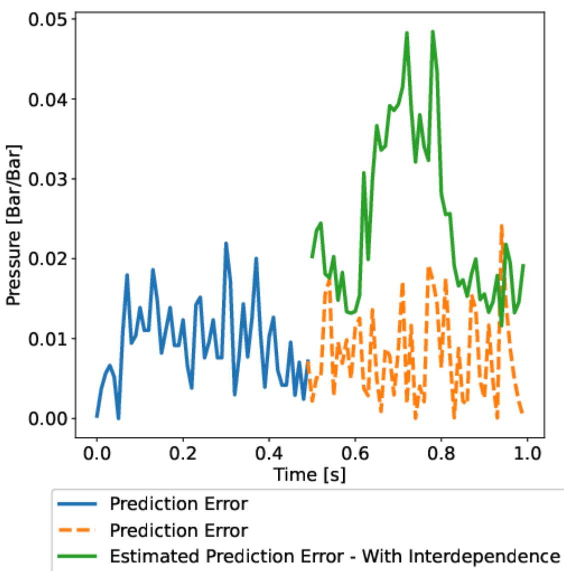

Figure 5: Prediction error and estimated prediction error for the PS1 sensor before and after measurements from the MPW sensor are replaced by predictions. The measurements from MPW start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS1 and MPW.

of the stability analysis, the results show that the prediction error estimation agrees with the observation.

4.3.3 Scenario 3 - Loop Unstable. In this scenario, we assume that the data points from PS1 and MPW sensors are replaced by their

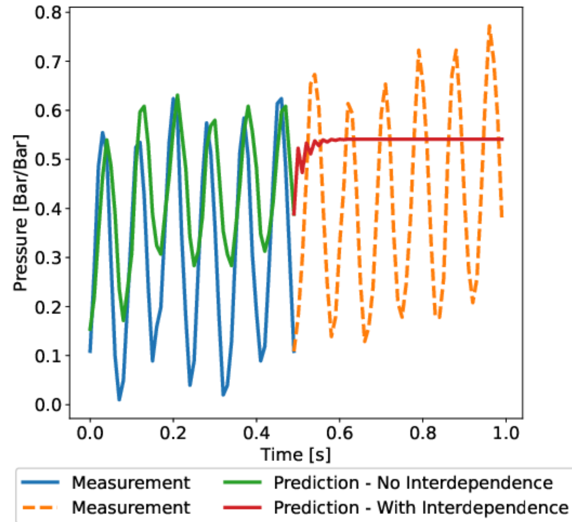

Figure 6: Measurements and predictions for the PS5 sensor before and after measurements from both PS5 and PS6 sensor are replaced by predictions. The measurements from PS5 and PS6 start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS5 and PS6.

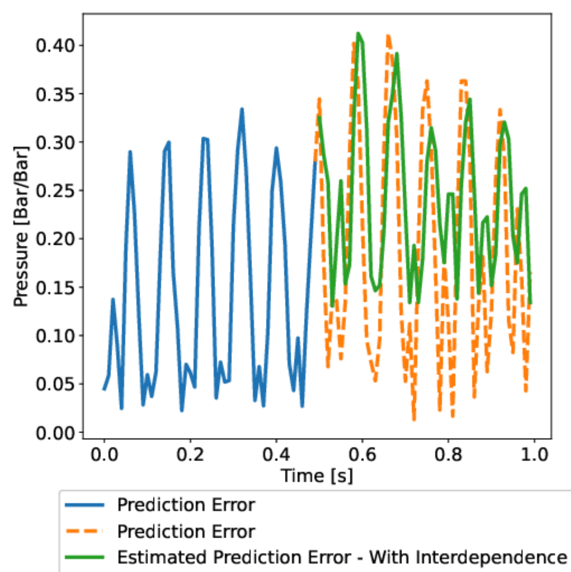

Figure 7: Prediction error and estimated prediction error for the PS5 sensor before and after measurements from both PS5 and PS6 sensor are replaced by predictions. The measurements from PS5 and PS6 start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS5 and PS6.

predictions and evaluate the effect of such replacement in the performance of the predictor for PS1 sensor data in terms of absolute

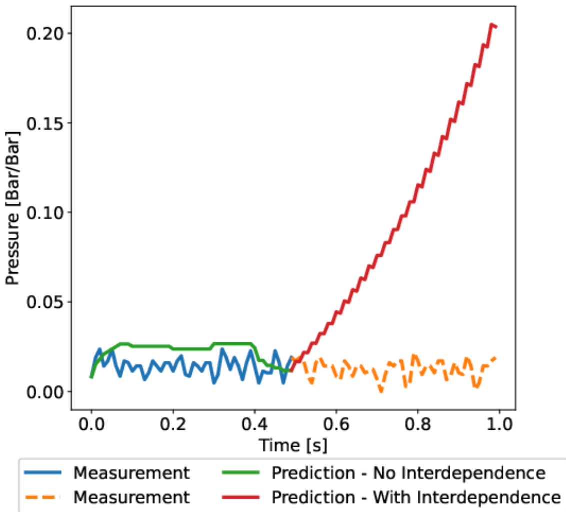

Figure 8: Measurements and predictions for the PS1 sensor before and after measurements from both PS1 and MPW sensor are replaced by predictions. The measurements from PS1 and MPW start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS1 and MPW.

prediction error. Since PS1 sensor data serves as input to the predictor of MPW sensor data and MPW sensor data, in turn, serves as input to the predictor of PS1 sensor data, this scenario stands for a loop interdependence configuration.

By evaluating the maximum gradient of both predictor's outputs, it is possible to find $\lambda_{PS1}^{MPW} = 1.03$ and $\lambda_{MPW}^{PS1} = 1.01$, which implies in $z = 1.02$, as stated in equation 16. Hence, the predictions of PS1's values are expected to become unstable.

Figures 8 and 9 show that the predictions get unstable, as predicted via stability analysis, which means that the prediction error grows exponentially as time evolves, an effect that is also captured by the estimated prediction error, which accurately captures the temporal evolution of the actual prediction error.

4.3.4 Scenario 4 - Feedback Stable. In this scenario, we assume that the data points from the PS6 sensor are replaced by their predictions and evaluate the effect of such replacement on the predictor's performance for its own sensor data in terms of absolute prediction error. As PS6 sensor data serves as input to its own predictor, this scenario represents a feedback configuration.

By evaluating the maximum gradient of the predictor's output, it is possible to find $\lambda_{PS6}^{r} = 0.36$, which implies in $z = 0.36$, as stated in equation 17. The predictions of PS6's values are expected to remain stable.

Figures 10 and 11 show that the prediction errors increase once the measurements are replaced by predictions, but remain stable, as predicted via the stability analysis. It is also clear that the estimation for the predicted error matches the observations for this case.

4.3.5 Scenario 5 - Feedback Unstable. In this scenario, we assume that the data points from the PS2 sensor are replaced by their predictions and evaluate the effect of such replacement on the performance of the predictor for its own sensor data in terms of absolute prediction error. Since PS2 sensor data serves as input to its

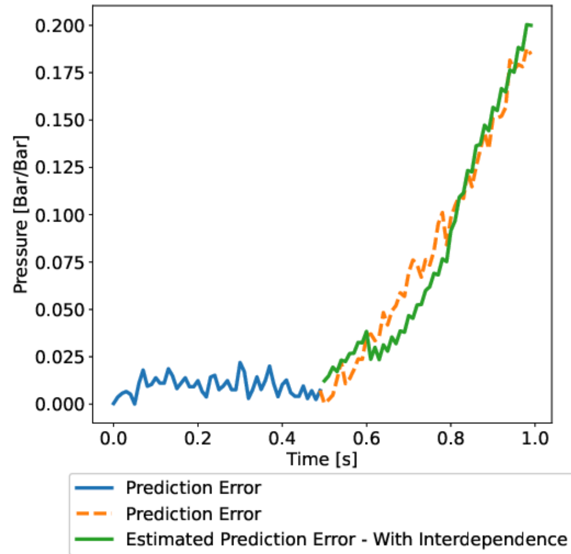

Figure 9: Prediction error and estimated prediction error for the PS1 sensor before and after measurements from the MPW sensor are replaced by predictions. The measurements from PS1 and MPW start to be replaced by predictions at t = 0.5s, leading to interdependence between the predictions for PS1 and MPW.

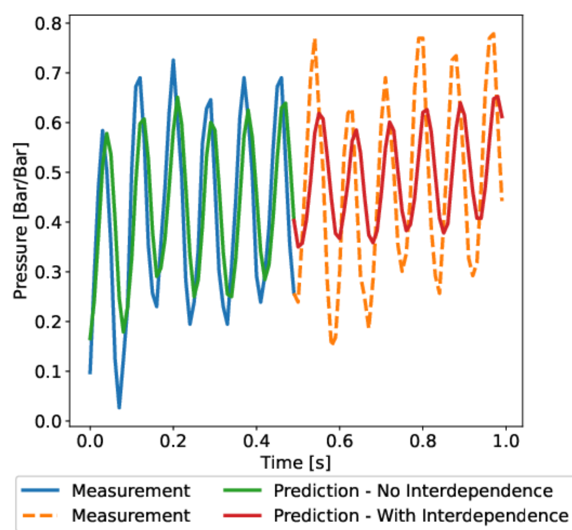

Figure 10: Measurements and predictions for the PS6 sensor before and after its measurements are replaced by predictions. The measurements from PS6 to be replaced by predictions at t = 0.5s, leading to a recurrent interdependence between the predictions for PS6.

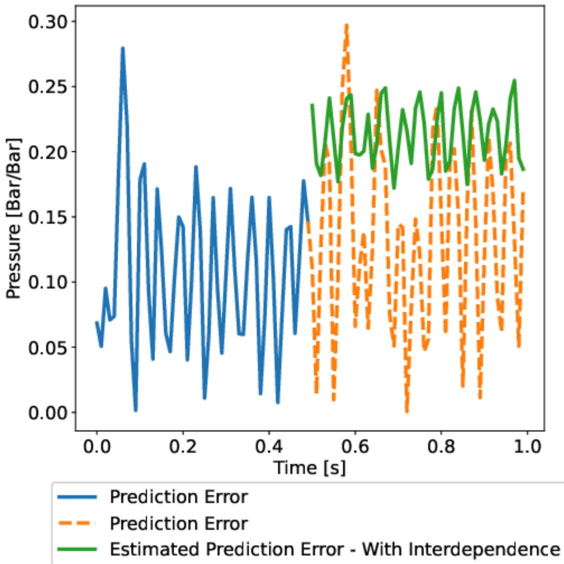

Figure 11: Prediction error and estimated prediction error for the PS6 sensor before and after its measurements are replaced by predictions. The measurements from PS6 to be replaced by predictions at t = 0.5s, leading to a recurrent interdependence between the predictions for PS6.

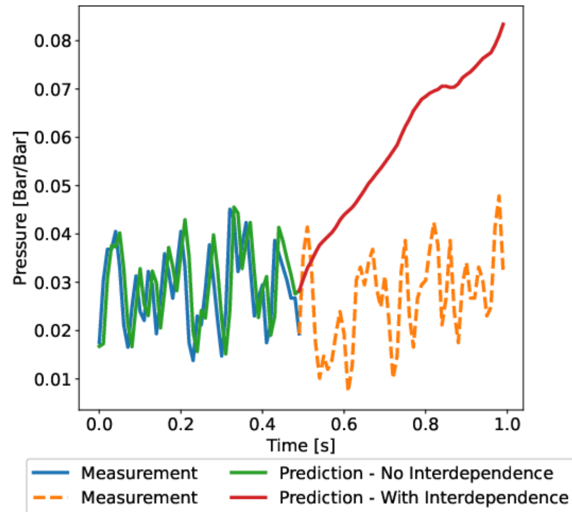

Figure 12: Measurements and predictions for the PS2 sensor before and after its measurements are replaced by predictions. The measurements from PS2 to be replaced by predictions at t = 0.5s, leading to a recurrent interdependence between the predictions for PS2.

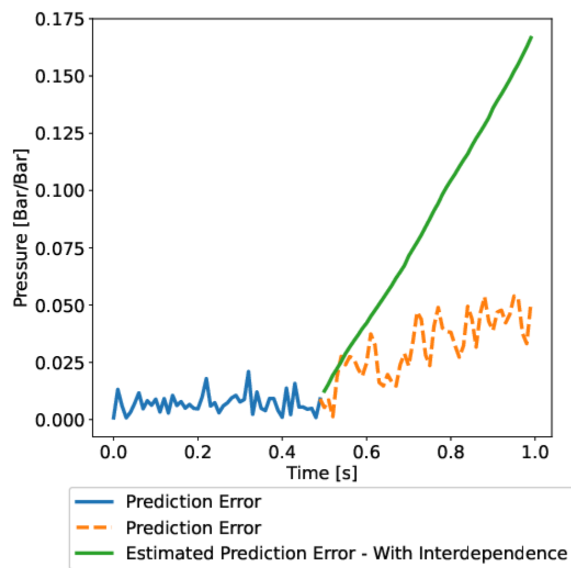

Figure 13: Prediction error and estimated prediction error for the PS2 sensor before and after its measurements are replaced by predictions. The measurements from PS2 to be replaced by predictions at t = 0.5s, leading to a recurrent interdependence between the predictions for PS2.

own predictor, this scenario represents a feedback interdependence configuration.

By evaluating the maximum gradient of the predictor's output, it is possible to find $\lambda^r_{PS2} = 1.01$, which implies in $z = 1.01$, as stated in equation 17. The predictions of PS2's values are expected to go unstable.

Figures 12 and 13 show that the prediction errors increase indefinitely once the measurements are replaced by predictions, characterizing, as expected, an unstable system. In this case, the estimated prediction error is not an accurate representation of the actual prediction error, but it still represents an upper bound for it and captures the unstable behavior of the predictor in this situation.

5 FINAL REMARKS

Multivariate predictors are usually able to produce more accurate predictions than other strategies. However, the accuracy of such predictors can decrease significantly whenever one or more of its inputs are actually outputs of other predictors. This effect can be even more intense in cases where there are cyclic dependencies between different predictors, as in the loop and feedback configurations described in Section 3.

In this work, we proposed a method to evaluate whether or not the dependencies between different predictors, that is, when inputs of some predictors are outputs of others, will lead to an unbounded growth of the prediction error due to the dynamics of error propagation, using techniques based on the stability analysis of discrete linear systems. This method can be used to assess the performance of systems that make use of several predictors qualitatively, in terms of prediction error stability, as well as quantitatively, in terms of the sensibility of the performance of a predictor with respect to the correctness of its inputs. Besides the stability analysis, the proposed method also provides an estimate for the upper bound of a predictor's error based on the error propagation along the ANN model.

The results presented in the previous section illustrate the effectiveness of the proposed method. In the first scenario, in which there was no cyclic dependence between predictors, the predictor's error remains bounded, as expected, and the proposed algorithm for estimating an upper bound for the prediction error is shown to produce accurate estimates. In cases like the one presented in the first scenario, an evaluation of the maximum absolute value of the predictor's gradient components, with respect to its inputs, a necessary step for the stability analysis, may still be useful to characterize the sensibility of the predictor to variations of its inputs.

The effectiveness of the method is corroborated by the results for scenarios 2, 3, 4, and 5, in which the predictors present different forms of cyclic dependency. The parameters of the predictors, acquired right after the training stage, were used to evaluate the roots of the characteristic equation for the error propagation dynamics, and the result of such process accurately predicted whether or not there would be unbounded growth of prediction error. For both stable and unstable configurations, the strategy used to estimate an upper bound fulfilled its purpose, being very accurate for scenarios 2, 3, and 4, but providing a poor estimate for the prediction error in scenario 5, yet still an upper bound.

When considering the possible impacts of different levels of variability of the input data on the results, one can assume different levels of variability of input data would present impacts on the predictors. The performance of ANN-based predictors is highly dependent on the training process and, in consequence, the data. In terms of the impacts on the presented method, in case the variability is caused by random noise, it is expected that the sensibility of the well-trained predictor's output to each specific variable would be smaller. Since the mutual information between the predictor's output and each input would be smaller, it is expected that the predictor would be more robust in terms of having a greater margin of stability, but less accurate. Furthermore, in case of higher variability in the random noise sense, the expected prediction error would be higher. Then, our estimate for the upper bound on prediction error would also be higher.

Other than that, we can also discuss the sensitivity of the presented method to different datasets. For this discussion, it is important to notice that, as previously mentioned, the ANN-based predictor performance is highly dependent on the input data. However, while the different datasets may incur a faster or slower decrease in the prediction error (according to the properties of the generated predictors), the dataset should not affect the validity of the mathematical analysis presented in the paper as long as the infinite differentiability property is still valid for the activation function of the trained predictors.

Given the presented results, the proposed solution represents two major contributions. The first is a method to estimate the uncertainty of a prediction, in terms of an upper bound for the predictor's error, even in the absence of measurement for the prediction to be compared against. The second is an analytical method to evaluate, at design time, the conditions in which a set of predictors with cyclic dependencies will generate predictions with bounded errors. This is a valuable contribution since, in many CPSs, the instability of a set of predictors may lead to problems in the decision-making process and even system failure.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) project INCANTO PrINT - Finance Code 001, and partially funded by ANP - Agência Nacional de Petróleo, Gás Natural e Biocombustíveis, Brasil, Petrobras project OBNZip.

REFERENCES

[1] Angelo Alessandri, Marco Baglietto, Giorgio Battistelli, and Victor Zavala. 2010. Advances in moving horizon estimation for nonlinear systems. In *49th IEEE Conf. on Decision and Control (CDC)*. IEEE. https://doi.org/10.1109/cdc.2010.5718126

[2] Martin Arlitt, Manish Marwah, Gowtham Bellala, Amip Shah, Jeff Healey, and Ben Vandiver. 2015. IoTAbench. In *Proc. of the 6th ACM/SPEC Int'l Conf. on Performance Engr.* ACM. https://doi.org/10.1145/2668930.2688055

[3] Chengyu Cao and Naira Hovakimyan. 2007. Novel ${\cal L}_{1}$ Neural Network Adaptive Control Architecture With Guaranteed Transient Performance. *IEEE Trans. on Neural Networks* 18, 4 (July 2007), 1160–1171. https://doi.org/10.1109/tnn.2007.899197

[4] C. L. Philip Chen, Yan-Jun Liu, and Guo-Xing Wen. 2014. Fuzzy Neural Network-Based Adaptive Control for a Class of Uncertain Nonlinear Stochastic Systems. *IEEE Trans. on Cybernetics* 44, 5 (May 2014), 583–593. https://doi.org/10.1109/tcyb.2013.2262935

[5] Adel A. Ghobbar and Chris H. Friend. 2003. Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model. *Computers & Operations Research* 30, 14 (Dec. 2003), 2097–2114. https://doi.org/10.1016/s0305-0548(02)00125-9

[6] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. 2015. Condition monitoring of a complex hydraulic system using multivariate statistics. *IEEE Int'l Instrumentation and Measurement Technology Conf. (I2MTC) Proc.* (2015), 210–215.

[7] Benjamin Karg and Sergio Lucia. 2021. Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning. *Computers & Chemical Engr.* 148 (May 2021), 107266. https://doi.org/10.1016/j.compchemeng.2021.107266

[8] Qinyuan Liu and Zidong Wang. 2021. Moving-Horizon Estimation for Linear Dynamic Networks With Binary Encoding Schemes. *IEEE Trans. on Automatic Control* 66, 4 (April 2021), 1763–1770. https://doi.org/10.1109/tac.2020.2996579

[9] Vittoriano Muttillo, Paolo Giammatteo, and Vincenzo Stoico. 2021. Statement-Level Timing Estimation for Embedded System Design Using Machine Learning Techniques. In *Proc. of the ACM/SPEC Int'l Conf. on Performance Engr.* ACM. https://doi.org/10.1145/3427921.3450258

[10] C. Nicolis, Rui A. P. Perdigao, and S. Vannitsem. 2009. Dynamics of Prediction Errors under the Combined Effect of Initial Condition and Model Errors. *Journal of the Atmospheric Sciences* 66, 3 (March 2009), 766–778. https://doi.org/10.1175/2008jas2781.1

[11] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. 1996. *Signals & Systems (2nd Ed.).* Prentice-Hall, Inc., USA.

[12] Björn F. Postema, Tobias Van Damme, Claudio De Persis, Pietro Tesi, and Boudewijn R. Haverkort. 2018. Combining Energy Saving Techniques in Data Centres using Model-Based Analysis. In *Companion of the 2018 ACM/SPEC Int'l Conf. on Performance Engr.* ACM. https://doi.org/10.1145/3185768.3186310

[13] Michael B. Richman, Theodore B. Trafalis, and Indra Adrianto. 2009. *Missing Data Imputation Through Machine Learning Algorithms.* Springer Netherlands, Dordrecht, 153–169. https://doi.org/10.1007/978-1-4020-9119-3_7

[14] Roberto M. Scheffel and Antnio A. Frhlich. 2018. WSN Data Confidence Attribution Using Predictors. In *2018 Eighth Latin-American Symp. on Dependable Computing (LADC)*. IEEE. https://doi.org/10.1109/ladc.2018.00025

[15] Roberto M Scheffel and Antônio A Fröhlich. 2019. Increasing sensor reliability through confidence attribution. *Journal of the Brazilian Computer Soc.* 25, 1 (2019), 1–20.

[16] Guido Schillaci, Alejandra Ciria, and Bruno Lara. 2020. Tracking Emotions: Intrinsic Motivation Grounded on Multi - Level Prediction Error Dynamics. In *2020 Joint IEEE 10th Int'l Conf. on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE. https://doi.org/10.1109/icdl-epirob48136.2020.9278106

[17] Connie U. Smith. 2020. Software Performance Antipatterns in Cyber-Physical Systems. In *Proc. of the ACM/SPEC Int'l Conf. on Performance Engr.* ACM. https://doi.org/10.1145/3358960.3379138

[18] Chunsheng Yang, Yanni Zou, Jie Liu, and Kyle R Mulligan. 2020. Predictive Model Evaluation for PHM. *Int'l Journal of Prognostics and Health Management* 5, 2 (Nov. 2020). https://doi.org/10.36001/ijphm.2014.v5i2.2238