Impulse Response and Granger Causality in **Dynamical Systems** with Autoencoder Nonlinear Vector Autoregressions

Kurt Izak Cabanilla
Thinking Machines Data Science

Kevin Thomas Go
Thinking Machines Data Science

**Abstract**

Sometimes knowing the future given the present is not enough. For sound policy making, predicting possible futures given different user defined scenarios can be more important. However, the workhorse for causality detection and impulse response, the Vector Autoregression (VAR), assumes linearity and has produced poor forecasts (Reis, 2018). Here, we introduce a vector autoencoder nonlinear autoregression neural network (VANAR) capable of both automatic time series feature extraction for its inputs and automatic functional form estimation. We compare the performance of VANAR and VAR across three tests: (1) forecasting skill, measured as n-step ahead forecast accuracy, (2) correct detection of Granger Causality between variables, and (3) impulse response tests on modeled trajectories subject to external shocks. These tests were performed on datasets with different underlying dynamics: a simulated nonlinear chaotic system, a simulated linear system, and an empirical system using Philippine macroeconomic data. Results show that VANAR significantly outperforms VAR in terms of the forecast and causality tests especially for the nonlinear and empirical macroeconomic systems. For the impulse response test, VANAR outperforms VAR in the linear system but both models fail to predict the shocked trajectories of the nonlinear chaotic system. VANAR was robust in its ability to model a wide variety of dynamics, from chaotic, high noise, and low data environments to complex macroeconomic systems, thus illustrating its potential usefulness in modeling more real world dynamical systems.

**Keywords:** Dynamical Systems, Deep Learning, Nonlinear Time Series Analysis
1.0 Introduction

Impulse response is simply the evolution of a dynamical system subject to a defined shock. Many dynamical systems such as those in neuroscience, ecology, and economics can be highly nonlinear. For example in macroeconomics, the entire economy is a complex network comprised of various industries and sectors, interest rates, exchange rates, inflation rates, labor, education, and household consumption among others (Sargent & Ljungqvist, 2018). These components have causal interactions with each other that can evolve. Introducing a shock such as a sudden increase in money supply from the central bank or fiscal investments from the government can drastically change the entire economy especially in the short run. How can we better know the impact of monetary policy over time? How much do we need to decrease the interest rate in order to improve economic output without too much inflation?

Classical approaches such as Vector Autoregression assume linearity and have produced unsatisfactory forecasts and simulations especially in macroeconomics (Reis, 2018). Furthermore, even standard nonlinear forms of VAR such as Threshold, Markov Switching, and Time Varying Parameters not only assume the type of nonlinearity but are also quite ad hoc in their specifications and hence require considerable tuning for different cases (Kilian & Lutkepohl, 2017).

Applications of machine learning on time series have primarily been on pure forecasting. There is little work on causality analysis and almost no work on impulse response estimation. Neural network approaches to VAR have been very sparse with the focus mostly on forecasting and only recently on causality (Fox et al., 2018). However, these neural VAR models only take in pure time series lags as input. They do not utilize other time series features which have been known to increase forecast accuracy in other models (Makridakis et al., 2014). While manually derived features are
feasible and powerful for univariate forecasting, in dynamical systems however, the number of important features can increase exponentially due to interaction terms. This makes feature engineering for multivariate time series models highly untenable, arbitrary, and time consuming.

To address these problems, we propose the vector autoencoder nonlinear autoregression (VANAR). It is a combination of two neural networks--an autoencoder for automatically extracting time series features for input processing, and a neural multilayer perceptron vector autoregression for automatically approximating the best functional form to fit the dynamical system.

We compare linear VAR and VANAR on three tests:

1. **A pure forecast test.** The causal inference and impulse response predictions of a model may not be robust or consistent if the model cannot make sufficiently accurate predictions relative to its competitors. This test will compare long horizon n-step ahead forecasts.
2. **Granger Causality test.** Which models consistently detect and identify the correct causal relationship?
3. **Impulse response test.** Which models most accurately simulate the dynamic effects of a shock through time?

The tests will be on three datasets: a two variable chaotic time series dataset simulated from a coupled logistic dynamical system, a three variable time series dataset generated from a stochastic linear system, and an empirical macroeconomic dataset comprising publicly available economic indicators from the Philippines. The mathematical details of the datasets are described in the results section.

Although we could test the accuracy of n-step ahead forecasts for non-simulated time series, we cannot do the same for impulse response precisely because we do not have access to true counterfactuals. In real life there is only a single actualization. Using simulated data generated from a given dynamical system, however, we can create the
true counterfactuals and thus gauge the accuracy of the impulse response functions. Nonetheless, we still do an impulse response analysis for the macroeconomic empirical dataset.

2.0 Problem Formulation: General Impulse Response

We define a general impulse response similar to (Kilian & Lutkepohl, 2017). For ease of exposition, we use only two variables, but it can be easily extended for $n$ variables.

Let the information set $\Omega_T = \{(x_1, y_1); (x_2, y_2); \ldots (x_T, y_T)\}$ be two time series of length $T$ and generated by some underlying discrete dynamical system $M$ and initial conditions $X_0$. $\Omega_T$ is essentially a sample trajectory $S(t \mid X_0)$. Let

$$\hat{x}_t = \hat{f}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p})$$

$$\hat{y}_t = \hat{g}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p})$$

be an estimated system of difference equations approximating $M$ given the information set, or $E[M \mid \Omega_T] = \hat{F}$, where $p$ is the max lag and $\hat{F} = (\hat{f}, \hat{g})'$.

This estimated dynamical system can be represented in vector form as

$$\hat{X}_t = \hat{F}(X_{t-1}, X_{t-2}, \ldots X_{t-p})$$

where $X_t = (x_t, y_t)'$ and $\hat{X}_t = (\hat{x}_t, \hat{y}_t)'$.

How will the system evolve given a sudden increase in one of the variables at a certain period? For instance, if $x$ is shocked at time $T$ by an impulse $\varepsilon$, then what is the new trajectory of both $x$ and $y$ over time? Using that example, we will have a new information set representing a counterfactual which we denote as:
\[ \Omega_{T,x}^{\varepsilon} = \{(x_1, y_1); (x_2, y_2); \ldots (x_T, y_T); \ldots (x_T + \varepsilon, y_T)\} . \]

Generally, \( \Omega_{T,x}^{\varepsilon} \) is defined as the **impulse information set** where we have \( T \) as the period of the shock, \( x \) as the shocked variable, and \( \varepsilon \) as the magnitude of the shock. We use this new information set as the input to the estimated dynamical system. We recursively produce outputs for all the variables in the system until the designed time path length \( H \) has been reached. The resulting set of values, or the counterfactual time path, is called the impulse response function. To obtain the predictions for \( T+1 \) we have:

\[
x_{T+1}^\hat{} = \hat{f}(x_T + \varepsilon, x_T, \ldots x_{T-p}, y_T, y_T, \ldots y_{T-m}) = \hat{f}(A_T \subseteq \Omega_{T,x}^{\varepsilon})
\]

\[
y_{T+1}^\hat{} = \hat{g}(x_T + \varepsilon, x_T, \ldots x_{T-p}, y_T, y_T, \ldots y_{T-m}) = \hat{g}(B_T \subseteq \Omega_{T,x}^{\varepsilon})
\]

We update the impulse information set for the next period, \( \Omega_{T+1,x}^{\varepsilon} \), with the predictions \( \hat{X}_t \) in order to track the dynamic effects of the initial shock:

\[
\Omega_{T+1}^{\varepsilon} = \{(x_1, y_1); (x_2, y_2); \ldots (x_T, y_T); \ldots (x_T + \varepsilon, y_T), (x_{T+1}^\hat{}, y_{T+1}^\hat{})\}
\]

Thus we have the new state for \( T+2 \):

\[
x_{T+2}^\hat{} = \hat{f}(x_{T+1}^\hat{}, x_T + \varepsilon, \ldots x_{T-p+1}, y_{T+1}^\hat{}, y_T, \ldots y_{T-m+1}) = \hat{f}(A_{T+1} \subseteq \Omega_{T+1,x}^{\varepsilon})
\]

\[
y_{T+2}^\hat{} = \hat{g}(x_{T+1}^\hat{}, x_T + \varepsilon, \ldots x_{T-p+1}, y_{T+1}^\hat{}, y_T, \ldots y_{T-m+1}) = \hat{g}(A_{T+1} \subseteq \Omega_{T+1,x}^{\varepsilon})
\]

And so on until \( \Omega_{T+H,x}^{\varepsilon} \) is recursively generated where \( H \) is the length of the time path after the shock period. \( \Omega_{T+H,x}^{\varepsilon} \) is defined as the **impulse information set recursively generated by** \( \hat{F} \).

The predicted values of the entire system after the period of the shock is then simply all the elements of \( \Omega_{T+H,x}^{\varepsilon} \) for \( t > T \). We define this formally.
Definition 1 (Impulse Path): Given an impulse information set $\Omega_{T+H,x}^e$ recursively generated by an estimation $\hat{F}$ of $\Omega_T$, the impulse path of $\hat{F}$ given a shock $\epsilon$ on variable $x$ at time $T$ is: $R_x^\epsilon(t) := (\hat{x}_t, \hat{y}_t) \in \Omega_{T+H}^e$ for $t > T$.

The impulse response is then simply the impulse path minus the impulse path given no shock:

Definition 2 (Impulse Response): Given an impulse information set $\Omega_{T+H,x}^e$ recursively generated by an estimation $\hat{F}$ of $\Omega_T$, the impulse response of $\hat{F}$ given a shock $\epsilon$ on variable $x$ at time $T$ is: $I_x^\epsilon(t) := R_x^\epsilon(t) - R_x^0(t)$ for $t > T$.

How do we find the estimated dynamical system $E[M \mid \Omega_T] = \hat{F}$ that will produce the most accurate n-step ahead forecasts, the correct causal inference, and most importantly, the most accurate impulse response?

2.1 Vector Autoregression

In a VAR with $p$ lags or a VAR-p, the estimated dynamic system has a linear functional form:

$$\hat{X}_t = \hat{F}(X_{t-1}, X_{t-2}, \ldots X_{t-p}) = \Phi_1 X_{t-1} + \Phi_2 X_{t-1} \ldots + \Phi_p X_{t-p} =$$

$$E[\Phi_1 X_{t-1} + \Phi_2 X_{t-1} \ldots + \Phi_p X_{t-p} + \xi_t]$$

Where $\Phi_i$ are $2x2$ (or $N \times N$ if there N variables in the system) coefficient matrices to be estimated and $\xi_t$ is a vector standard normal error term that has no serial autocorrelation: $E[\xi_t \xi_{t-1}] = 0$ and $E[\xi_t] = 0$.

Traditionally, impulse response functions for VARs are computed using structural shocks in the error term $\xi_t$, but this is an equivalent way to specify the general impulse
response shown above. We use ordinary least squares (OLS) to estimate VAR and the classical Akaike Information Criterion (AIC) to select the optimal lag $p$.

### 2.2 Vector Autoencoder Nonlinear Autoregression

A Vector Autoencoder Nonlinear Autoregression with $p$ lags, or a VANAR-$p$, model has two components, an autoencoder for the input processing, and a vector autoregressive neural network for the dynamical system estimation.

The autoencoder is comprised of two multilayer perceptrons neural networks (MLP): an encoder that compresses its input data to a lower dimension, and a decoder that takes the compressed output of the encoder and decompresses it to approximate the data in its original dimension. The objective of the encoder is to extract the most important features of its input data while the objective of the decoder is to reconstruct the original inputs based on those important features alone.

Given $\Omega_T$, the input of the encoder $E$ is essentially the same as that of the estimated dynamical system $E[M \mid \Omega_T]$. It takes in the vectors of the form $(X_{t-1}, X_{t-2}, \ldots X_{t-p})$ comprised of the $p$ lags of two variables and outputs a lower dimensional vector $\mathbb{R}^{\text{embedding dimension}}$. Given the two variable system, the dimension of this input vector is thus $p \times 2$, while the output is simply the user selected embedding dimension.

$$E : \mathbb{R}^{p \times 2} \rightarrow \mathbb{R}^{\text{embedding dimension}}$$

The decoder $D$ then takes in the output of the encoder and outputs back a vector as close as possible to the encoder input vector $X_{t-1}, X_{t-2}, \ldots X_{t-p}$:

$$D : E(\mathbb{R}^{p \times 2}) \rightarrow \mathbb{R}^{p \times 2}$$

$$\text{Autoencoder} = D \circ E : \mathbb{R}^{p \times 2} \rightarrow \mathbb{R}^{\text{embedding dimension}} \rightarrow \mathbb{R}^{p \times 2}$$

The output of the encoder $E(\mathbb{X}_{t-1}, \mathbb{X}_{t-2}, \ldots \mathbb{X}_{t-p})$ is essentially the vector of automatically extracted time series features. This is then concatenated to $(\mathbb{X}_{t-1}, \mathbb{X}_{t-2}, \ldots \mathbb{X}_{t-p})$ and then used as input to another multilayer perceptron $\hat{\mathbb{N}}$, a vector autoregressive artificial
neural network (VAR-ANN), which is basically just a VAR but instead each equation in the VAR is a multilayer perceptron; it is a nonlinear vector autoregression. \( \hat{N} \) without an autoencoder is simply \( \hat{N}: R^{p \times 2} \rightarrow R^2 \). With an activated autoencoder, \( \hat{N} \) is \( \hat{N}: R^{(p+embedding\ dimension) \times 2} \rightarrow R^2 \).

It is important to note that \( \hat{N} \) need not necessarily be a neural network, it can be any other form of estimator whether linear or nonlinear such as a random forest regressor or whatnot. For the purposes of this paper, however, the nonlinear vector autoregression is a multilayer perceptron neural network.

For a VANAR-p with two hidden layers and \textit{deactivated autoencoder}, the first equation \( \hat{x}_t \) of \( \hat{N} \) is represented as:

\[
\hat{x}_t = \begin{bmatrix} X_1 & X_2 & \cdots & X_{t-p} \end{bmatrix} W_1 \cdot h_1 \left( W_2 \cdot \begin{bmatrix} x & b_0 \end{bmatrix} + b_1 \right) = \hat{f}(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, y_{t-1}, y_{t-2}, \ldots, y_{t-p})
\]

Where \( x = (X_{t-1}, X_{t-2}, \ldots X_{t-p})' \), \( W_1 \) denotes the weight matrix consisting of the values of \( w_{ij} \) with \( i \) as row and \( j \) as column coordinates, \( b_0 \) is a vector whose elements are all \( b_0 \), \( h^1 \) is a vector valued function where the sigmoid, or any other function (called activation functions) such as the tanh, is applied to the vector \( W_1 x + b_0 \), \( W_2 \) denotes the weight matrix consisting of the values of \( w_{kj} \), and so on until the vector \( h^2 \) is multiplied by the transpose of the weight vector \( W_3 \) to produce the scalar output \( \hat{x}_t \).

The weights of VANAR are then iteratively estimated using a gradient descent based optimization algorithm.

The estimated dynamical system for \( M \) given a VANAR-p with an activated autoencoder is the following:

\[
\hat{X}_t = \hat{F} \left( X_{t-1}, X_{t-2}, \ldots X_{t-p} \right) = \hat{N} \left( X_{t-1}, X_{t-2}, \ldots X_{t-p}, E(X_{t-1}, X_{t-2}, \ldots X_{t-p}) \right)
\]

\( \hat{F} \) is once more comprised of two different equations, one for each variable:
\[
\hat{x}_t = \hat{f}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p}, E(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p}))
\]
\[
\hat{y}_t = \hat{g}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p}, E(x_{t-1}, x_{t-2}, \ldots x_{t-p}, y_{t-1}, y_{t-2}, \ldots y_{t-p}))
\]

A VANAR-p with two hidden layers, the first equation \( \hat{x}_t \) in a VANAR-p can be represented as:

\[
\hat{x}_t = W_5^T h_2 (W_2 \cdot h_2 (W_1 x + b_0) + b_1)
\]

Where \( x = (X_{t-1}, X_{t-2}, \ldots X_{t-p}, E(X_{t-1}, X_{t-2}, \ldots X_{t-p}))' \)

The univariate version of VANAR is the Autoencoder Nonlinear Autoregression, or ANA. It is simply:

\[
\hat{x}_t = \hat{f}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, E(x_{t-1}, x_{t-2}, \ldots x_{t-p}))
\]

\[
\hat{y}_t = \hat{g}(x_{t-1}, x_{t-2}, \ldots x_{t-p}, E(x_{t-1}, x_{t-2}, \ldots x_{t-p}))
\]

### 2.2.1 VANAR Architecture and Lag Selection

The number of lags \( p \) is decided by an initial VAR estimation using the classical Akaike Information Criterion (AIC), thus VANAR has the same \( p \) as VAR. VANAR has two forms. The architecture of the first form is shallow but extremely wide. It only has two hidden layers but the number of neurons in each layer often ranges around 5000. The optimizer is the AdaGrad algorithm with a learning rate of 0.0001. The activation functions are RELU for the hidden layers and linear for the output layer. The second form has no hidden layers and a hyperbolic tangent output activation. It also uses AdaGrad but instead with a learning rate of 0.1. The form VANAR takes will depend on a validation set (a subset of the training set used for hyperparameter tuning). Once a given form has been selected, VANAR is retrained but this time for the entirety of the training set. Form 1 is trained with around 3000 epochs while Form 2, 20,000 epochs. The autoencoder only has one form and consists of 3 hidden layers each with RELU activations and a linear activation for the output layer. Note that the autoencoder can be
activated or deactivated depending on the length of the optimal lag and the validation set forecast accuracy. If the ideal lag is below 4, then feature extraction will not make much sense and thus is no longer considered. ANA has the same architecture and two forms as VANAR but instead is comprised of a single equation as it is univariate. The architecture diagram of VANAR Form 1 is shown below:

Theoretically, a VANAR-\( p \) generalizes a VAR-\( p \) since neural networks are generalizations of linear regressions due to the Universal Approximation Theorem (Hornik et al., 1989). But how do we test this empirically? To empirically verify that VANAR generalizes VAR, it should produce forecasts and impulse response functions that are superior to VAR for simulated time series generated from nonlinear dynamical systems and sufficiently close to VAR for simulated linear dynamical systems.

However, to define which variables can be shocked, you must first know which variables are causally related.
2.3 Granger Causality

Granger Causality states that variable $x$ Granger Causes variable $y$ if the history of $x$ improves the forecast accuracy of a model that only uses the history of $y$. This is done with linear autoregressive models.

We use the definition of Granger Causality in Hamilton (1996) but extend it to nonlinear functions:

**Definition 3.** Given a dynamical system estimate $\hat{F}$, out of sample values $y_t$, predicted values $\hat{y}_t$, and an error metric $L(\hat{y}_t, y_t)$ such as the Root Mean Squared Error (RMSE), a variable $x$ is said to not Granger Cause variable $y$ if the test set (out of sample) error

$$L \left( \hat{g} \left( x_{t-1}, x_{t-2}, \ldots, x_{t-p}, y_{t-1}, y_{t-2}, \ldots, y_{t-p} \right), y_t \right)$$

is greater than or equal to

$$L \left( \hat{g} \left( y_{t-1}, y_{t-2}, \ldots, y_{t-p} \right), y_t \right)$$

If otherwise, $x$ is said to Granger Cause $y$.

The test set forecasts are used rather than in sample predictions because train set overfitting can lead to misleading results.

It follows that the degree or strength of the Granger Causal variables can be quantified by how much they improve the error of a purely univariate model. For instance, in a system with three variables ($x$, $y$, and $z$), variable $x$ has a stronger Granger Causal effect on $y$ than variable $z$ if

$$L \left( \hat{g} \left( x_{t-1}, x_{t-2}, \ldots, x_{t-p}, y_{t-1}, y_{t-2}, \ldots, y_{t-p} \right), y_t \right)$$

is less than

$$L \left( \hat{g} \left( z, z_{t-2}, \ldots, z_{t-p}, y_{t-1}, y_{t-2}, \ldots, y_{t-p} \right), y_t \right)$$
Is less than

$$L(\hat{g}(y_{t-1}, y_{t-2}, \ldots y_{t-p}), y_t)$$

This nonlinear Granger Causality need not be restricted to bivariate or pairwise testing. Comparing the accuracy of an $n$ variable model with a univariate model can show how a collection of $n-1$ variables can have a causal effect on the $n$th variable, we call this $n-1$ Granger Causality. *For systems with more than two variables, we test using the n-1 Granger Causality.*

The version of neural Granger Causality developed by Fox et al. (2018) relied more on the coefficients of the estimated neural model, particularly the first set of weights before the first hidden layer. It used a hyperparameter $\lambda$ in the loss function to force these weights to be comprised mostly of zeroes and ones. This approach cannot be used for VANAR precisely because of the autoencoder component. Furthermore, the hyperparameter $\lambda$ makes this method more arbitrary and difficult to infer as different values of $\lambda$ can lead to either overly sparse weights or no sparsity at all. No sparsity at all could be misleading as the low weight values in the first hidden layer could still mean strong effects for the output because the other weights in the network could balance it out. On the other hand, overly sparse weights might eliminate too many variables and throw away rich information on the dynamics.

### 3.0 Data and Outline of Tests

The tests will be on three datasets: a two variable time series dataset simulated from a coupled logistic dynamical system, a three variable time series dataset generated from a stochastic linear system (VAR), and an empirical dataset using Philippine Macroeconomic data. There are three tests: an n-step ahead forecast test, a causality test, and an impulse response test. The error metric used is the Root Mean Squared Error (RMSE); a lower RMSE means better forecast accuracy.
The first test is the n-step ahead forecast comparison between VAR and VANAR, where RMSE of the two methods are compared. The long horizon forecast is the twenty step ahead forecast while the short horizon forecast is the ten step ahead forecast.

The second test is the Granger Causality Test. Given the forecast scores of VANAR and VAR from the first test, we now evaluate the forecast scores of their univariate forms--ANA and AR, respectively. We then compare the scores according to definition of Granger Causality from Section 2.3. The outputs of this test are binary: either the model correctly identifies the proper causal relationship or not.

The impulse response test is also a forecast comparison like the first test but instead with an alternate data input due to the impulse shock. The model forecasts are then the estimated impulse response paths given the shock. This is then compared with the simulated true impulse response paths given the underlying dynamical system.

All three tests will be done on all datasets. However, for the empirical dataset, only the forecast test can be properly evaluated since we do not know the underlying causal dynamical system (i.e., cannot check correctness of Granger causality test) and thus have no access to the real counterfactuals (i.e., cannot compare impulse responses between VAR- and VANAR-estimated trajectories and the “true” trajectory given a shock). Nonetheless, causal analysis as well as impulse response analysis will still be done for insights into the Philippine economy.
4.0 Results

4.1 System 1: Two-variable nonlinear system with mirage correlation

We simulated a two-variable chaotic system following Sugihara et al. (2012) Science:

\[ X(t) = X(t-1)[3.8 - 3.8X(t-1) - 0.02Y(t-1)] + \varepsilon(t) \text{ where } N(0, 0.01) \]  
\[ Y(t) = Y(t-1)[3.5 - 3.5Y(t-1) - 0.1X(t-1)] + \varepsilon(t) \text{ where } N(0, 0.01) \]  

For the first scenario, we simulate the entire system without any noise. This system produces the so-called 'mirage correlation', where two variables can appear to be correlated or anti-correlated at certain points in time, then lose this pattern at other points in time (Fig 2). However, when considering the entire time series of \( n = 1000 \) steps, the Spearman correlation between the two variables is only 0.09, despite them being coupled.

![Fig 2. Illustration of ‘mirage correlation’ in System 1. Depending on the slice of the time series being examined, the two variables in the system can appear anti-correlated](image)
greyed section), correlated (second greyed section), or no clear patterns (third greyed section). Spearman’s correlation of entire time series (n = 1000 steps) is 0.09.

For this system, we have four scenarios: the first is the Default Scenario which has no noise and has the exact functional form as (1) and (2), the second is the No Interaction scenario where the interaction coefficients of (1) and (2) are zero which implies that X has no effect on Y and vice versa, the third scenario is With Noise 1 which is (1) and (2) but with a white noise error term with a mean of zero and a standard deviation of 0.1, and the fourth is With Noise 2 which is the same as the previous scenario but instead with a standard deviation of 0.01. Note that when the noise is added, it is not part of the recursion but rather is added only once the default system has been fully simulated. Hence it only distorts the pattern and is considered as “observation error”.

Furthermore, for each scenario, we conduct the tests across three environments varying in the amount of data provided: a High Data Richness Environment where there are 850 observations available for training, a Medium Data Richness Environment where there are 350 observations available for training, and a Low Data Richness Environment where there are only 50 observations available for training.

The test set is the next 20 steps from the training set.

4.1.1 N-steps Ahead Forecast Test

For all scenarios and for all environments, VANAR outperforms VAR for the long horizon forecast test (20 steps ahead) and significantly outperforms VAR in the short horizon test, often being twice as accurate for the short horizon. Detailed tables are in the appendix. For all tests for System 1, VANAR form 1 was used. An active autoencoder VANAR rather than a deactivated autoencoder was used for the vast majority of scenarios and environments since it yielded improved accuracy.
4.1.2 Causality Test

For all scenarios and for all environments except one, VANAR detects the correct causality, while VAR fails in several scenarios and environments (Table 1).

**Table 1.** Causality tests of VANAR and VAR across different scenarios and data richness environments. If causality is incorrectly detected, details on which variables detection failed is given in parentheses.

| Scenario      | Method | High Data Richness | Medium Data Richness | Low Data Richness |
|---------------|--------|---------------------|----------------------|-------------------|
| Default       | VANAR  | Correct             | Correct              | Correct           |
|               | VAR    | Incorrect (both)    | Correct              | Incorrect (fails to detect y on x) |
| No Interaction| VANAR  | Correct             | Correct              | Correct           |
|               | VAR    | Correct             | Correct              | Correct           |
| Noise 1       | VANAR  | Incorrect (fails to detect x on y) | Correct | Correct |
|               | VAR    | Incorrect (both)    | Correct              | Incorrect (fails to detect x on y) |
| Noise 2       | VANAR  | Correct             | Correct              | Correct           |
|               | VAR    | Incorrect (both)    | Correct              | Incorrect (fails to detect y on x) |
4.1.3 Impulse Response Test

The impulse response test is done only in the default system scenario and the rich data environment. A shock of 0.1 is added to variable \( y \) at time step 850 and we forecast the impulse response of \( x \) for twenty periods given the shock. Results show that the impulse response functions of both VAR and VANAR remain very close to zero at all times while the true impulse response remains close to zero only for around the first ten time steps and diverges drastically from there. This demonstrates the sensitivity to initial conditions that are characteristic of chaotic systems (Banks et al., 1992). Furthermore, it implies that given a chaotic system and a single trajectory generated from initial conditions, models estimated on that trajectory tend to be dynamically stable. This can perhaps be solved by adding a noise term inside the true system itself and generating a random trajectory given the same initial conditions. Models estimated on that random trajectory might learn how the system reacts to shocks. Theoretical justifications for this phenomena are a subject of future research.

**Fig 3.** Impulse response forecasts of VAR, VANAR, and System 1 for variable \( x \) in response to a shock of 0.1 added to variable \( y \) at time step 850.
4.2 System 2: Three variable Linear Stochastic System

We simulated a stable 3-variable 5-lag VAR system where the interaction terms were generated by drawing from a uniform distribution ranging from values [-0.7, 0.7]. Initial conditions were drawn randomly from a uniform distribution of [-1, 1], and the system was simulated for 120 time steps. Stability was ensured since all eigenvalues of the simulated system’s companion matrix fell within the unit circle. We also applied process noise $e$ at each point in the dataset, where $e \sim N(0, 0.1)$. Since this is a simulated VAR system, a VAR model should generally have the best fit. Mathematically, System 2 is written as:

$$X_t = \Phi_1 X_{t-1} + \Phi_2 X_{t-1} + \ldots + \Phi_5 X_{t-5} + e_t$$  \hspace{1cm} (3)

where $X_{t-1}, X_{t-2}, \ldots X_{t-5}$ are each [3 x 1] vectors representing the state variables at each time-lag, $\Phi_1, \Phi_2, \ldots \Phi_5$ are the interaction matrices representing the effects of all variables on each other at each time-lag, and $e_t$ is the process noise.
**Fig 4.** Full time series of the simulated data in System 2 with process noise $e$, where $e \sim N(0,0.1)$.

### 4.2.1 N-steps Ahead Forecast Test

The first 100 steps were used for training while the last 20 were for testing. Despite the underlying system itself being a VAR, VANAR 2nd form is superior to VAR for all except the third variable for the twenty step ahead long horizon test. For the ten step ahead short term horizon, VANAR’s performance is considerably worse than VAR although still capable of capturing the general trend of the system. Thus even though VANAR is nonlinear, it was able to sufficiently estimate the overall trend of the linear system.

**Table 2.** 20 steps and 10 steps ahead forecast accuracy of VANAR and VAR for System 2 variables $y_1$, $y_2$, and $y_3$.

| 20 steps ahead for variable n | VANAR-19 Form2, activated autoencoder | ANA       | VAR-19 | AR-19 |
|------------------------------|---------------------------------------|-----------|--------|-------|
| y1                           | 0.6015036                             | 0.9778138 | 0.778  | 0.974 |
| y2                           | 0.5770663                             | 1.1487226 | 0.682  | 0.874 |
| y3                           | 0.6981658                             | 0.8962612 | 0.670  | 0.783 |

| 10 steps ahead for variable n | VANAR-19 Form2, activated autoencoder | ANA       | VAR-19 | AR-19 |
|------------------------------|---------------------------------------|-----------|--------|-------|
| y1                           | 0.6210068                             | 0.6013922 | 0.379  | 0.775 |
| y2                           | 0.5641976                             | 0.8029597 | 0.332  | 0.552 |
| y3                           | 0.6320933                             | 0.7737009 | 0.162  | 0.717 |
4.2.2 Granger Causality Test

Given the table of forecast error metrics above, we can test for Granger Causality. In the linear stochastic system, VANAR and VAR both detect the correct causal relationship for all variables. Thus, for causal inference, VANAR can approximate linear systems as well as a linear model.

Table 3. 20 steps ahead n-1 Granger Causality Test

| 20-steps ahead Granger Causality Test | VANAR (all four variables as input) | VAR |
|--------------------------------------|-------------------------------------|-----|
| y2 and y3 causing y1                 | Correct                             | Correct |
| y1 and y3 causing y2                 | Correct                             | Correct |
| y1 and y2 causing y3                 | Correct                             | Correct |

4.2.2 Impulse Response Test

We shocked variable y2 by an impulse of 0.1 at time period 100 and generated the impulse response path for variable y1. Note that in generating the true impulse response path of equation (3) no random process noise was used except for the initial conditions used to generate the path, thus we have the deterministic effect of a shock. The models were nonetheless still trained on equation (3) with process noise was until period 100 where the shock occurs. The denoising of the true impulse response was done in order to see if the models captured the true effect of a shock. For the impulse response test, VANAR forecasts were much more robust. VANAR impulse response paths across different estimation runs consistently had a lower RMSE than the VAR impulse response path. VAR has an RMSE of 0.029 while VANAR has an RMSE of 0.020. VANAR’s impulse response path has a steady fluctuation slightly above zero.
Fig 5. Impulse response forecasts of VAR, VANAR, and the linear stochastic System 2. VANAR has a steady oscillating impulse response path but nonetheless best approximates the true impulse response.

4.3 System 3: Philippine Macroeconomy

We now use VANAR and VAR to model the Philippine economy using the following macroeconomic variables: *GDP annual growth rate, inflation rate, employment rate, interest rate, lending rate, M1 money supply (log transformed), fiscal expenditure (log transformed), remittances (log transformed), and industrial production.* Despite some of the variables being nonstationary, we do not difference any of them in order to keep the full information of the dynamics necessary for the impulse response analysis. The data is quarterly ranging from 1992 to all of 2018. Monthly variables are aggregated to quarterly frequency with a simple mean.

The test set is comprised of the last four quarters which is the entire year of 2018. We first compare the forecast accuracy of the models with respect to GDP growth and then
we analyze the dynamic effects of expansionary fiscal and monetary policy shocks on the economy.

**Fig 5.** Philippine GDP Year on Year Growth, quarterly frequency.

### 4.3.1 N-steps Ahead Forecast Test

VANAR significantly outperforms VAR in the four steps ahead forecast test. VAR has an RMSE of 1.662 while VANAR has an RMSE of 0.441.
Fig 6. Philippine GDP Year on Year Growth forecasts of VANAR and VAR

Table 4. Four steps ahead forecast accuracy of VANAR and VAR for quarterly Philippine GDP Growth

| 4 steps ahead forecast | VANAR-9. Form1, Activated autoencoder | VAR-9 |
|------------------------|----------------------------------------|-------|
| GDP Growth RMSE        | 0.441                                  | 1.662 |

4.3.2 Impulse Response Analysis

We examine the dynamic effects of expansionary monetary policy by applying a positive shock of 1 to the M1 Money Supply variable during the 4th quarter of 2017. The impulse response is generated for up to three years ahead.
Fig 7. Impulse Response of GDP Growth from an Expansionary Monetary Policy Shock

The impulse response fundamentally represents the difference between predicted GDP growth with a shock and predicted GDP growth without a shock. The first period in the graph above is the origin point of the shock and thus has no effect yet on the economy. The impulse response undulates with a mean near zero, giving some empirical confirmation to the neutrality of money theory. Monetary policy shocks seem to have no real effect on the economy in the long run. There is a short run positive effect however.

Similarly, we now examine the effects of applying a positive shock of 1 to the Fiscal Expenditures variable. Thus, instead of expansionary monetary policy, we have an increase in government spending and investments, otherwise known as expansionary fiscal policy.
Unlike, expansionary monetary policy which is neutral in the long run, expansionary fiscal policy has significant real effects in both the short run and the long run.

4.3.3 Univariate One Step Ahead Forecast Test

Lastly, to show how VANAR holds up to classical and state of the art univariate forecasting methods, we forecast Philippine monthly inflation and monthly tourist arrivals using VANAR’s univariate version, ANA, and compare it with the classical automatic Seasonalized Autoregressive Integrated Moving Average (SARIMA), the state of the art TBATS (Exponential Smoothing State Space model with Box-Cox Transformation, ARMA errors, Trend and Seasonal components), and the automatic single hidden layer autoregressive multiplayer perceptron (MLP). Instead of an n-step ahead forecast, this is a one step ahead forecast. Training set for inflation is from February 1990 to December 2017. The test set is the entirety of 2018 plus January of 2019, a total of thirteen observations. For tourist arrivals, the training set is from January
1991 to December 2016. The test set is the entirety of 2017 and 2018, a total of twenty four observations.

![Philippine Monthly Inflation Rate](image)

**Fig 9.** Philippine Monthly Inflation Rate time series from February 1990 to January 2019.

The error metric used here is the Root Mean Squared Scaled Error which is just the RMSE of the model divided by the RMSE of a naive forecast where a naive forecast is using the value at the current period as the forecast for the next period. Thus an RMSSE greater than one signifies a model forecast worse than a naive forecast while an RMSSE of less than one signifies true gains in forecasting accuracy. For inflation rate, ANA is the best performing model with the automatic SARIMA close behind.

**Table 5.** Philippine monthly Inflation Rate one step ahead forecast RMSSE for SARIMA, TBATS, MLP, and ANA.

| Model               | SARIMA | TBATS | MLP      | ANA-3 Form1, Deactivated Autoencoder |
|---------------------|--------|-------|----------|-------------------------------------|
| RMSSE               | 0.744  | 0.808 | 0.954    | 0.704                               |
**Fig 10.** Philippine Inflation Rate test set forecasts for ANA.

**Fig 11.** Philippine Monthly Tourist Arrivals time series from January 1991 to December 2018.
For the Tourist Arrivals time series, ANA is once more the best performing model and with a considerable margin over the other models.

**Table 7.** Philippine monthly Tourist Arrivals one step ahead forecast RMSSE for SARIMA, TBATS, MLP, and ANA.

| Model | SARIMA | TBATS | MLP | ANA |
|-------|--------|-------|-----|-----|
| RMSSE | 0.935  | 0.892 | 0.726 | 0.567 |

**Fig 12.** Philippine Tourist Arrivals test set forecasts for ANA.

## 5.0 Conclusion

Dynamical systems are present in almost every aspect of reality, from the interactions of the cells in your body to the movements of entire economies. Forecasting these systems are an important and foundational step towards a more complete causal and
counterfactual modeling of these systems. Dynamical system modeling has so far been dominated by either meticulously developed theoretical mathematical models or linear stochastic estimators such as VAR. The former is cumbersome to construct and may contain a great deal of assumptions while the latter may be too simple. We proposed a general framework for modeling dynamical systems, causal inference, and estimating impulse response. An estimation model, linear or nonlinear, such as a random forest or a neural network can do forecasting, causal inference, and impulse response as long as it fulfills the necessary autoregressive input and output form of the proposed framework. The vector autoencoder nonlinear autoregression neural network (VANAR) is one such model that was able to provide superior forecasts over VAR for the simulated nonlinear system and the empirical macroeconomic system. For causal inference, it consistently outperformed VAR and detected the correct causality even in noisy and low data richness conditions. Applications of the general dynamical system framework and VANAR to neuroscience, ecology, medicine, and more macroeconomics remains to be explored. Given its robust performance across different datasets and different time series tasks, VANAR is a strong candidate for use in more real world dynamical systems for policymakers and researchers alike.

REFERENCES

Banks et al. (2018). On Devaney's Definition of Chaos. The American Mathematical Monthly, 99(4):

Fox, E., Tank, A., Covert, I., Foti, N., & Shojaie, A. (2018). Neural granger causality for nonlinear time series. arXiv:1802.05842

Hornik, K., Stinchcombe, N., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2: 359-366
Kilian, L., & Lütkepohl, H. (2017). Structural Vector Autoregressive Analysis (Themes in Modern Econometrics). Cambridge: Cambridge University Press. doi:10.1017/9781108164818

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: concerns and ways forward. PLOS ONE 13(3): e0194889. https://doi.org/10.1371/journal.pone.0194889

Reis, R. (2018). Is something really work with macroeconomics? Oxford Review of Economic Policy, Volume 34, Issue 1-2, 5 January 2018, Pages 132–155, https://doi.org/10.1093/oxrep/grx053

Sargent, T., & Ljungqvist, L. (2018) Recursive macroeconomic theory, 4th edition. MIT Press: Cambridge, MA

Sugihara, G., May, R., Ye, H., Hsieh, C., Deyle, E., Fogarty, M., & Munch, S. (2012). Detecting causality in complex ecosystems. Science (New York, N.Y.). 338. 10.1126/science.1227079.
APPENDIX

System 1: Two-variable nonlinear system with mirage correlation

We simulated a two-variable chaotic system following Sugihara et al. (2012) Science:

\[ X(t) = X(t-1)[3.8 - 3.8 X(t-1) - 0.02 Y(t-1)] + \varepsilon(t) \text{ where } N(0, 0.01) \]  

\[ Y(t) = Y(t-1)[3.5 - 3.5 Y(t-1) - 0.1 X(t-1)] + \varepsilon(t) \text{ where } N(0, 0.01) \]

For this system, we have four scenarios: the first is the Default Scenario which has no noise and has the exact functional form as (1) and (2), the second is the No Interaction scenario where the interaction coefficients of (1) and (2) are zero which implies that X has no effect on Y and vice versa, the third scenario is With Noise 1 which is (1) and (2) but with a white noise error term with a mean of zero and a standard deviation of 0.1, and the fourth is With Noise 2 which is the same as the previous scenario but instead with a standard deviation of 0.01. Note that when the noise is added, it is not part of the recursion but rather is added only once the default system has been fully simulated. Hence it only distorts the pattern and is considered as “observation error”.

Furthermore, for each scenario, we conduct the tests across three environments varying in the amount of data provided: a High Data Richness Environment where there are 850 observations available for training, a Medium Data Richness Environment where
there are 350 observations available for training, and a **Low Data Richness Environment** where there are only 50 observations available for training.

The test set is the next 20 steps from the training set.

**Default Scenario**

**High Data Richness Environment**
- Training data = 850 steps

| n steps ahead for variable X | VANAR-14. Form1, Activated autoencoder. | ANA | VAR-14 | AR-14 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.1356056                              | 0.1768853 | 0.257 | 0.230 |
| 10                          | 0.08673972                              | 0.1174932 | 0.293 | 0.258 |

| n steps ahead for variable Y | VANAR-14. Form1, Activated autoencoder. | ANA | VAR-14 | AR-14 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.01805058                              | 0.04122844 | 0.064 | 0.054 |
| 10                          | 0.00580254                              | 0.04659928 | 0.060 | 0.037 |

**Medium Data Richness Environment**
- Training data = 250 steps

| n steps ahead | VANAR-9. | ANA | VAR-9 | AR-9 |
|---------------|----------|-----|-------|------|
| 20            |          |     |       |      |
| 10            |          |     |       |      |
### for variable X

|          | VANAR-9. Form1, Activated autoencoder. | ANA   | VAR-9 | AR-9 |
|----------|----------------------------------------|-------|-------|------|
|          |                                        |       |       |      |
| 20       | 0.1196829                              | 0.130279 | 0.206 | 0.242 |
| 10       | 0.1060848                              | 0.1207651 | 0.202 | 0.238 |

### n steps ahead for variable Y

|          | VANAR-9. Form1, Activated autoencoder. | ANA   | VAR-9 | AR-9 |
|----------|----------------------------------------|-------|-------|------|
|          |                                        |       |       |      |
| 20       | 0.01846403                             | 0.02132268 | 0.020 | 0.018 |
| 10       | 0.01318353                             | 0.01881621 | 0.016 | 0.016 |

### Low Data Richness Environment
- Training data = 50 steps

### n steps ahead for variable X

|          | VANAR-14. Form1, Activated autoencoder. | ANA   | VAR-4 | AR-4 |
|----------|----------------------------------------|-------|-------|------|
|          |                                        |       |       |      |
| 20       | 0.2471158                              | 0.3300185 | 0.265 | 0.245 |
| 10       | 0.2415083                              | 0.12016 | 0.245 | 0.216 |

### n steps ahead for variable Y

|          | VANAR-14. Form1, Activated autoencoder. | ANA   | VAR-4 | AR-4 |
|----------|----------------------------------------|-------|-------|------|
|          |                                        |       |       |      |
| 20       | 0.03056499                             | 0.0373002 | 0.031 | 0.084 |
No Interaction Scenario

We repeat the methods in the Default Scenario but with a different dataset: we simulate System 1 where the two variables have no interaction (i.e., $Y_{(t-1)}$ no longer appears in the formula for $X_{(t)}$, and vice-versa).

High Data Richness Environment
- Training data = 850 steps

| n steps ahead for variable X | VANAR-65 Form1, Deactivated autoencoder. | ANA | VAR-65 | AR-65 |
|-----------------------------|----------------------------------------|------|--------|-------|
| 20                          | 0.2206835                              | 0.2058791 | 0.233 | 0.232 |
| 10                          | 0.1757822                              | 0.1546297 | 0.201 | 0.189 |

| n steps ahead for variable Y | VANAR-65 Form1, Deactivated autoencoder. | ANA       | VAR-65   | AR-65    |
|-----------------------------|----------------------------------------|-----------|----------|----------|
| 20                          | 0.01711407                             | 1.744163e-05 | 0.090 | < 0.001 |
| 10                          | 0.01885756                             | 1.447229e-05 | 0.090 | < 0.001 |

Medium Data Richness Environment
- Training data = 250 steps
### Low Data Richness Environment

- Training data = 50 steps

#### n steps ahead for variable X

|                  | VANAR-24 Form1, Deactivated autoencoder. | ANA | VAR-24 | AR-24 |
|------------------|-----------------------------------------|-----|--------|-------|
| 20               | 0.2662899                               | 0.2304825 | 0.300 | 0.243 |
| 10               | 0.233162                                | 0.1979543 | 0.358 | 0.252 |

#### n steps ahead for variable Y

|                  | VANAR-24 Form1, Deactivated autoencoder. | ANA | VAR-24 | AR-24 |
|------------------|-----------------------------------------|-----|--------|-------|
| 20               | 0.01255132                               | 0.0003848478 | 0.090 | <0.001 |
| 10               | 0.01101204                               | 0.0002955276 | 0.090 | <0.001 |

#### n steps ahead for variable Y

|                  | VANAR-5 Form1, Deactivated autoencoder. | ANA | VAR-5 | AR-5 |
|------------------|----------------------------------------|-----|-------|------|
| 20               | 0.2703183                               | 0.2588373 | 0.272 | 0.245 |
| 10               | 0.2548257                               | 0.2326701 | 0.276 | 0.225 |

#### n steps ahead for variable Y

|                  | VANAR-5 Form1, Deactivated autoencoder. | ANA | VAR-5 | AR-5 |
|------------------|----------------------------------------|-----|-------|------|
| 20               | 0.1077617                               | 0.003335119 | 0.095 | 0.008 |
With Observation Noise 1 Scenario

We repeat the methods in the Default Scenario but with a different dataset: we simulate System 1 but with observation noise. Noise is introduced at each time step as a normally-distributed random variable with mean = 0 and sd = 0.1.

SD = 0.1

High Data Richness Environment
- Training data = 850 steps

| n steps ahead for variable X | VANAR-25. Form1, Activated autoencoder | ANA | VAR-25 | AR-25 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.2450847                              | 0.2475296 | 0.310 | 0.260 |
| 10                          | 0.27111667                              | 0.2804906 | 0.365 | 0.295 |

| n steps ahead for variable Y | VANAR-25. Form1, Activated autoencoder | ANA | VAR-25 | AR-25 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.09659208                             | 0.04982226 | 0.140 | 0.104 |
| 10                          | 0.1021324                              | 0.02934832 | 0.173 | 0.113 |

Medium Data Richness Environment
- Training data = 250 steps

| n steps ahead | VANAR-9. | ANA | VAR-9 | AR-9 |
|---------------|----------|-----|-------|------|

for variable X
Form1, Activated autoencoder.

|   |     |     |     |     |
|---|-----|-----|-----|-----|
| 20 | 0.04789752 | 0.2531016 | 0.261 | 0.262 |
| 10 | 0.03368176  | 0.07449114 | 0.292 | 0.271 |

n steps ahead for variable Y
VANAR-9. Form1, Activated autoencoder.

|   |     |     |     |     |
|---|-----|-----|-----|-----|
| 20 | 0.01743703 | 0.04430856 | 0.131 | 0.104 |
| 10 | 0.01130574  | 0.03508665 | 0.125 | 0.098 |

Low Data Richness Environment
- Training data = 50 steps

n steps ahead for variable X
VANAR-5. Form1, Activated autoencoder.

|   |     |     |     |     |
|---|-----|-----|-----|-----|
| 20 | 0.2593454  | 0.225887 | 0.303 | 0.310 |
| 10 | 0.2452869  | 0.2311914 | 0.212 | 0.263 |

n steps ahead for variable Y
VANAR-5. Form1, Activated autoencoder.

|   |     |     |     |     |
|---|-----|-----|-----|-----|
| 20 | 0.1071115  | 0.1549625 | 0.162 | 0.129 |
| 10 | 0.1141862  | 0.1758735 | 0.180 | 0.155 |
With Observation Noise 2 Scenario

We repeat the methods in the Observation Noise 1 Scenario but with the noise term having a standard deviation of 0.01.

SD = 0.01

High Data Richness Environment
- Training data = 850 steps

| n steps ahead for variable X | VANAR-14. Form1, Activated autoencoder. | ANA | VAR-14 | AR-14 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.1770601                              | 0.3436085 | 0.257  | 0.231 |
| 10                          | 0.08852866                              | 0.2923459 | 0.298  | 0.260 |

| n steps ahead for variable Y | VANAR-14. Form1, Activated autoencoder. | ANA | VAR-14 | AR-14 |
|-----------------------------|----------------------------------------|-----|--------|-------|
| 20                          | 0.0193065                              | 0.03161021 | 0.053  | 0.042 |
| 10                          | 0.009916833                             | 0.03103254 | 0.063  | 0.045 |

Medium Data Richness Environment
- Training data = 250 steps
### n steps ahead for variable X

|        | VANAR-9. Form1, Activated autoencoder. | ANA     | VAR-9 | AR-9 |
|--------|---------------------------------------|---------|-------|------|
| 20     | 0.2541653                             | 0.4106635 | 0.210 | 0.246 |
| 10     | 0.1450822                             | 0.3488136 | 0.200 | 0.239 |

### n steps ahead for variable Y

|        | VANAR-9. Form1, Activated autoencoder. | ANA     | VAR-9 | AR-9 |
|--------|---------------------------------------|---------|-------|------|
| 20     | 0.02026843                            | 0.02010566 | 0.022 | 0.022 |
| 10     | 0.009910785                           | 0.01341163 | 0.018 | 0.016 |

### Low Data Richness Environment

- Training data = 50 steps

|        | VANAR-5. Form1, Activated autoencoder. | ANA     | VAR-5 | AR-5 |
|--------|---------------------------------------|---------|-------|------|
| 20     | 0.2132341                             | 0.2160254 | 0.266 | 0.243 |
| 10     | 0.2275326                             | 0.1893417 | 0.249 | 0.209 |

|        | VANAR-5. Form1, Activated autoencoder. | ANA     | VAR-5 | AR-5 |
|--------|---------------------------------------|---------|-------|------|
| 20     | 0.02770695                            | 0.04308755 | 0.030 | 0.077 |
|   |              |              |      |      |
|---|-------------|-------------|------|------|
| 10| 0.02502873  | 0.05695597  | 0.035| 0.081|