Received: 28 November 2021 | Revised: 30 January 2022 | Accepted: 8 February 2022

DOI: 10.1002/aic.17642

RESEARCH ARTICLE

Process Systems Engineering

AIChE JOURNAL

Statistical machine-learning–based predictive control of uncertain nonlinear processes

Zhe Wu[1] | Aisha Alnajdi[2] | Quanquan Gu[3] | Panagiotis D. Christofides[2,4]

[1]Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore

[2]Department of Electrical and Computer Engineering, University of California, Los Angeles, California, USA

[3]Department of Computer Science, University of California, Los Angeles, California, USA

[4]Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California, USA

Correspondence
Panagiotis D. Christofides, Department of Chemical and Biomolecular Engineering and the Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, USA.
Email: pdc@seas.ucla.edu

Abstract

In this study, we present machine-learning–based predictive control schemes for nonlinear processes subject to disturbances, and establish closed-loop system stability properties using statistical machine learning theory. Specifically, we derive a generalization error bound via Rademacher complexity method for the recurrent neural networks (RNN) that are developed to capture the dynamics of the nominal system. Then, the RNN models are incorporated in Lyapunov-based model predictive controllers, under which we study closed-loop stability properties for the nonlinear systems subject to two types of disturbances: bounded disturbances and stochastic disturbances with unbounded variation. A chemical reactor example is used to demonstrate the implementation and evaluate the performance of the proposed approach.

KEYWORDS

generalization error, machine learning, model predictive control, recurrent neural networks, stochastic nonlinear systems

1 | INTRODUCTION

Machine learning has gained increasing attention in modeling nonlinear systems due to powerful learning strategies, the availability of big data sets, and the development of computing resources. While the training error of machine learning models could be sufficiently low with good-quality datasets and a careful tuning of model hyper-parameters, a fundamental challenge for the implementation of machine learning models in chemical process control is the generalization performance on unseen data. Generalization error bound provides an efficient way to measure the effectiveness of training and accuracy of machine learning models. The generalization error bound relies on various factors, including data sample size, bounds of weight matrices, and the number of neurons and layers. Many recent studies have been done to obtain the generalization error bounds for the implementation of neural networks in classification problems with single output.[1–4] Additionally, in Bartlett et al.,[5] a margin-based multi-class generalization bound was derived for the neural networks based on their margin-normalized spectral complexity. In Chen et al.[6] and Wu et al.,[7] the generalization error for RNNs was developed for multiclass classification problems, and regression problems of multi-input and multi-output (MIMO) nonlinear systems, respectively.

Additionally, model predictive controllers (MPC) using machine learning models have been studied in recent years, with successful applications to a number of chemical engineering problems.[8–11] As machine learning models can capture complex process dynamics, machine-learning–based MPCs have demonstrated their superior closed-loop performance when compared with the MPCs using (usually linear) data-driven models in traditional industrial process control systems. Most existing works on data-driven or learning-based MPC derived closed-loop stability properties based on the assumption that the generalization error is bounded. However, this assumption may not hold in practice. Additionally, machine learning models are typically approximations of the nominal system dynamics, and thus, how to deal with uncertainty in processes within machine-learning–based MPCs is an important issue that requires further study.

Motivated by the above considerations, we develop RNN-based MPC schemes for nonlinear systems with model uncertainty in this manuscript. While MPC of stochastic nonlinear systems has been extensively studied in literature, for example,[12–16] very few research works study machine-learning–based MPC for stochastic nonlinear systems. A recent work developed an artificial neural network for a stochastic multiscale chemical engineering system under uncertainty in MPC.[17] However, at this stage, machine-learning–based MPC of

stochastic nonlinear systems is still in its infancy. In this study, we perform a probabilistic closed-loop stability analysis for the nonlinear systems subject to two common types of disturbances (unknown-but-bounded disturbances and stochastic disturbances with unbounded variation) under RNN-MPC based on the generalization error bound derived for RNN models. The theoretical study provides a guidance showing how to improve machine learning models in a systematic way in order to achieve desired accuracy in both open-loop and closed-loop simulations. The rest of this article is organized as follows: in Section 2, the notations, the nonlinear systems, and the RNN formulation are presented. In Section 2.4, a generalization error bound is derived for RNNs through Rademacher complexity approach. In Section 3, closed-loop stability results are developed for the nonlinear systems subject to bounded, and unbounded, stochastic disturbances, respectively. Finally, in Section 4, we use a chemical reactor as an example to illustrate the relation between training sample size and the RNN generalization error as well as the probability of closed-loop system stability.

2 | PRELIMINARIES

2.1 | Notation

The transpose of $\mathbf{x}$ is denoted by $\mathbf{x}^T$. The Lie derivative is $L_f V(\mathbf{x}) := \frac{\partial V(\mathbf{x})}{\partial x} f(\mathbf{x})$. The operator $|\cdot|$ denotes the Euclidean norm of a vector. $|\cdot|_Q$ denotes the weighted Euclidean norm of a vector, where Q is a positive definite matrix. The Frobenius norm of A is denoted by $\|A\|_F$. Set subtraction is denoted by "\", that is, $A \setminus B := \{x \in \mathbf{R}^n \mid x \in A,\ x \notin B\}$. Given a set $\mathcal{D}$, the boundary of $\mathcal{D}$ is denoted by $\partial\mathcal{D}$, and the interior of $\mathcal{D}$ is denoted by $\mathcal{D}^o$. The first hit time (or the hitting time) of a set X is defined as the first time that the state trajectory hits the boundary of X, and is denoted by τ_X. Also, we define $\tau_X(t) = \min\{\tau_X, t\}$ and $\tau_{X,T}(t) = \min\{\tau_X, T, t\}$, where T is the operation time.

$\mathbf{R}_+$ represents nonnegative real numbers. A function $f(x)$ belongs to class C^k if for all $i = 1, 2, \ldots, k$, the ith derivative of f exists and is continuous. A function $f : \mathbf{R}^n \to \mathbf{R}^m$ is l-Lipschitz continuous, if for all $a,\ b \in \mathbf{R}^n$, $|f(a) - f(b)| \le l\,|a - b|$ holds, where $l \ge 0$ is a real constant. A continuous function $\alpha : [0, a) \to [0, \infty)$ belongs to a class $\mathcal{K}$ function if it is zero only when evaluated at zero, and is strictly increasing. $\mathbb{E}[X]$ is the expected value of a random variable X, and $\mathbb{P}(A)$ is the probability of the event A occurring.

2.2 | Class of systems

The following state-space model represents the class of continuous-time nonlinear systems considered in this work:

$$\dot{x} = F(x,\ u) := f(x) + g(x)u,\ x(t_0) = x_0, \tag{1}$$

where the n-dimensional state vector is denoted by $x \in \mathbf{R}^n$, and $u \in \mathbf{R}^k$ denotes the k-dimensional manipulated input vector bounded by $u \in U$. The set U defines the maximum u_{max} and the minimum value u_{min} for input vectors, that is, $U := \{u_{min} \le u \le u_{max}\} \subset \mathbf{R}^k$. The vector $f(\cdot)$ and the matrix functions $g(\cdot)$ are sufficiently smooth with dimensions $n \times 1$, and $n \times k$, respectively. We assume that $f(0) = 0$ without loss of generality, and therefore, the origin is a steady-state of Equation (1). Additionally, we assume that $t_0 = 0$ (i.e., the initial time is zero).

We assume that there exists a feedback controller $u = \Phi(x) \in U$ under which the origin can be rendered exponentially stable. The stabilizability assumption implies that there is a $\mathcal{C}^1$ Lyapunov function $V(x)$ such that for all x in D the following inequalities hold:

$$c_1|x|^2 \le V(x) \le c_2|x|^2, \tag{2a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x)) \le -c_3|x|^2, \tag{2b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \le c_4\,|x|\,, \tag{2c}$$

where D is an open neighborhood around the origin, and c_i, $i = 1, 2, 3, 4$ are positive constants. We follow the method in Wu et al.[11] to generate the data by carrying out extensive open-loop simulation for the system of Equation (1) with various inputs $u \in U$ and initial conditions x_0 to develop a set of time-series data for $x \in \Omega_\rho$, where Ω_ρ is a level set of Lyapunov function (i.e., $\Omega_\rho := \{x \in \mathbf{R}^n \mid V(x) \le \rho\}$, $\rho > 0$) utilized as the operating region. Then, we develop recurrent neural network (RNN) models for capturing system dynamics and predicting state evolution. Specifically, the RNN models predict future states $x(t)$, $t > t_k$ based on the current state measurements $x(t_k)$ at time $t = t_k$, and the manipulated inputs $u(t)$, $t > t_k$.

2.3 | Recurrent neural networks

In this section, we consider a general RNN model developed with m sequences of data $(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$, where $\mathbf{y}_{i,t} \in \mathbf{R}^{d_y}$ is the RNN output, and $\mathbf{x}_{i,t} \in \mathbf{R}^{d_x}$, $i = 1, \ldots, m$, and $t = 1, \ldots, T$ (T is the time length) is the RNN input, to capture the system dynamics of Equation (1). The RNN input/state/output vectors are written in boldface to differentiate the notations from those for the nonlinear system of Equation (1). Additionally, to simplify the discussion, we develop the RNN model of Equations (3) and (4) to predict future states for one sampling period (denoted by Δ) with internal steps $T = \frac{\Delta}{h_c}$, where h_c is the integration time step used by the Explicit Euler method to solve the continuous-time system of Equation (1), and Δ is the sampling period within which the control action $u(t)$ remains unchanged (i.e., for all $t = 1, \ldots, T$). The RNN model predicts one sampling period forward, including all the internal states every h_c time step. As a result, for $t = 1, \ldots, T$, the predicted states are the RNN output $\mathbf{y}_{i,t}$, and the inputs and the current state measurements are RNN input $\mathbf{x}_{i,t}$.

The time-series data is generated independently following the data distribution over $\mathbf{R}^{d_x \times T} \times \mathbf{R}^{d_y \times T}$. Then, we develop the dataset by

generating m data sequences of the same distribution. To simplify the discussion, a one-hidden-layer RNN (Figure 1) is considered. The RNN states in the hidden layers $\mathbf{h}_i \in \mathbf{R}^{d_h}$ are

$$\mathbf{h}_{i,t} = \sigma_h(U\mathbf{h}_{i,t-1} + W\mathbf{x}_{i,t}), \tag{3}$$

where the weight matrices $W \in \mathbf{R}^{d_h \times d_x}$ and $U \in \mathbf{R}^{d_h \times d_h}$ are associated with the input and hidden state vectors, respectively. The element-wise nonlinear activation function is denoted by σ_h (e.g., ReLU). The output layer $\mathbf{y}_{i,t}$ is calculated using the following equation:

$$\mathbf{y}_{i,t} = \sigma_y(V\mathbf{h}_{i,t}), \tag{4}$$

where the activation function σ_y and the weight matrix $V \in \mathbf{R}^{d_y \times d_h}$ are associated with the output layer.

We have the following standard assumptions on the RNN model and datasets.

Assumption 1. The RNN inputs are bounded, i.e., $|\mathbf{x}_{i,t}| \le B_X$, for all $i = 1,...,m$ and $t = 1,...,T$.

Assumption 2. The Frobenius norms of the weight matrices are bounded as follows:

$$\|W\|_F \le B_{W,F}, \|V\|_F \le B_{V,F}, \|U\|_F \le B_{U,F} \tag{5}$$

Assumption 3. All the datasets (i.e., training, validation, and testing) are drawn from the same distribution.

Assumption 4. σ_h is a 1-Lipschitz continuous activation function, and is positive-homogeneous in the sense that $\sigma_h(\alpha z) = \alpha\sigma_h(z)$ holds for all $\alpha \ge 0$ and $z \in \mathbf{R}$.

Remark 1. Assumptions 1–4 are standard assumptions in machine learning theory. Specifically, Assumptions 1, 2 assume the boundedness of RNN inputs and weight matrices. This is consistent with the practical implementation of RNN training that only a finite class of RNN hypotheses are searched for the optimal solution. Assumption 3 is also a basic assumption that is widely adopted in machine learning modeling works. Note that we only require that the testing, validation and training data are drawn from the same distribution, and there is no specific requirement for the type of data distribution. This is a reasonable assumption from a practical application point of view as it implies that the training, validation and testing data sets are obtained from the same chemical process. The RNN models trained from the process operational data will be tested against data that come from the same target distribution. Assumption 4 requires positive-homogeneity of the activation function. For example, Rectified Linear Unit (ReLu), a popular nonlinear activation function in the machine learning domain, is a candidate of activation function that meets this assumption.

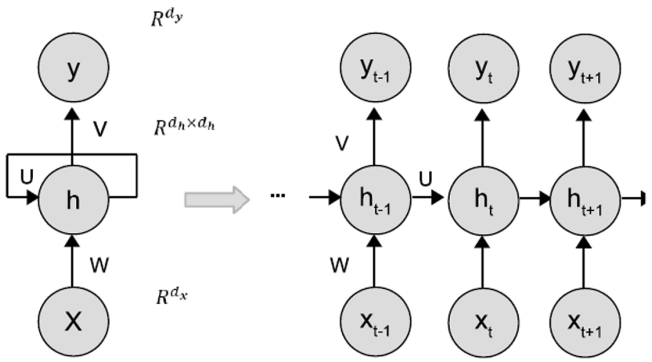

FIGURE 1 Recurrent neural network structure

2.4 | RNN generalization error

The RNN learning algorithms provide no information on the generalization performance for unseen testing data since they are evaluated on training data only. Therefore, the generalization error is used to measure the neural network's predictive capability for any data not utilized in training. Specifically, an upper bound is developed in this section for the RNN generalization error. Then we demonstrate that with high probability, this error is bounded if the development of RNN models meets a few requirements.

2.5 | Preliminaries

Let $\mathscr{H}$ be the hypothesis class of RNN functions $h(\cdot)$ that map a d_x-dimensional input $\mathbf{x} \in \mathbf{R}^{d_x}$ to a d_y-dimensional output $\mathbf{y} \in \mathbf{R}^{d_y}$. The predicted output of the RNN model and the loss function are denoted by $\mathbf{y}_t = h(x_t)$ and $L(\mathbf{y}_t, \overline{\mathbf{y}}_t)$, respectively, where $L(\mathbf{y}, \overline{\mathbf{y}})$ calculates the squared difference between the predicted output $\mathbf{y}$ and the true output $\overline{\mathbf{y}}$. We have the following error definitions for training RNN models.

Definition 1. Given a data distribution D, and a function h that predicts y (output) based on x (input), the **generalization error** or **expected loss/error** is:[18]

$$\mathbb{E}[L(h(x), y)] = \int_{X \times Y} L(h(x), y)\rho(x, y)dxdy \tag{6}$$

where the joint probability distribution for x and y is represented as $\rho(x, y)$, and the vector space for all possible outputs and inputs are denoted by Y and X, respectively.

Since in most cases ρ is an unknown distribution, we utilize empirical error as an approximation measure for the expected error. The empirical error is calculated as follows.

Definition 2. Given a dataset $S=(s_1,\ldots,s_m)$, $s_i=(x_i,y_i)$, with m data samples drawn from the data distribution D, the **empirical risk** or **error** is:[18]

$$\widehat{\mathbb{E}}_S[L(h(x),y)]=\frac{1}{m}\sum_{i=1}^{m}L(h(x_i),y_i) \tag{7}$$

Since the RNN data is generated within a compact set, the RNN predicted outputs $\mathbf{y}_t$ and the true outputs $\overline{\mathbf{y}}_t$ are assumed to be bounded by $r_t>0$, $t=1,\ldots,T$, that is, $|\mathbf{y}_t|, |\overline{\mathbf{y}}_t|\le r_t$. Thus, the following inequality from local Lipschitz continuity holds for the loss function of mean squared error (MSE) for all $|\mathbf{y}_t|\le r_t$, and $|\overline{\mathbf{y}}_t|\le r_t$.

$$|L(\mathbf{y}_2,\overline{\mathbf{y}})-L(\mathbf{y}_1,\overline{\mathbf{y}})|\le L_r|\mathbf{y}_2-\mathbf{y}_1|, \tag{8}$$

where the local Lipschitz constant is denoted by L_r.

2.6 | Rademacher complexity

Rademacher complexity is used to quantify the richness of a function class in computational learning theory. The definition of empirical Rademacher Complexity is presented below.

Definition 3. Given a set of data samples $S=(s_1,\ldots,s_m)$, and a hypothesis class $\mathscr{F}$ of real-valued functions, the definition of empirical Rademacher complexity of $\mathscr{F}$ is[18]:

$$\mathscr{R}_S(\mathscr{F})=\mathbb{E}_\epsilon\left[\sup_{f\in\mathscr{F}}\frac{1}{m}\sum_{i=1}^{m}\epsilon_i f(s_i)\right] \tag{9}$$

where $\epsilon=(\epsilon_1,\ldots,\epsilon_m)^T$, and ϵ_i are Rademacher random variables that are independent and identically distributed (i.i.d.) and satisfy $\mathbb{P}(\epsilon_i=-1)=\mathbb{P}(\epsilon_i=1)=0.5$.

The Rademacher complexity is used to derive the generalization error bound in the following lemma.

Lemma 1. (c.f. Theorem 3.3 in Mohri et al.[18]) Let $\mathscr{H}$ be the hypothesis class that maps $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\}\in\mathbf{R}^{d_x\times t}$ (i.e., the first t-time-step inputs) to $\boldsymbol{y}_t\in\mathbf{R}^{d_y}$ (i.e., the t-th output), and $\mathcal{G}_t$ be loss function set with $\mathscr{H}$.

$$\mathcal{G}_t=\{g_t:(\boldsymbol{x},\overline{\boldsymbol{y}})\to L(h(\boldsymbol{x}),\overline{\boldsymbol{y}}),h\in\mathscr{H}\} \tag{10}$$

where $\overline{\boldsymbol{y}}$ and $\boldsymbol{x}$ are the true output vector and the RNN input vector, respectively. Then, given a dataset consisting of m i.i.d. data samples, the inequality below holds in probability for all $g_t\in\mathcal{G}_t$ over the data samples $S=\left(\boldsymbol{x}_{i,t},\boldsymbol{y}_{i,t}\right)_{t=1}^{T}$, $i=1,\ldots,m$:

$$\mathbb{E}[g_t(\boldsymbol{x},\boldsymbol{y})]\le\frac{1}{m}\sum_{i=1}^{m}g_t(\boldsymbol{x}_i,\ \boldsymbol{y}_i)+2\mathscr{R}_S(\mathcal{G}_t)+3\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m}} \tag{11}$$

The RHS of Equation (11) represents the upper bound for the generalization error, which relies on various factors. Specifically, the first term of the RHS in Equation (11) represents the empirical risk, the second term represents the Rademacher complexity, and an error function of the samples size m and the confidence δ is represented in the last term. Note that the last and the first terms can be computed once a set of training data of size m and the confidence δ are given. Therefore, our goal is to derive the Rademacher complexity bound for $\mathscr{R}_S(\mathcal{G}_t)$.

2.7 | Generalization error bound

We first present a few lemmas to provide preliminary results following the proof technique in.[21]

Lemma 2. (c.f. Lemma 4 in Wu et al.[7]) Given a dataset $S=\left(\boldsymbol{x}_{i,t},\boldsymbol{y}_{i,t}\right)_{t=1}^{T}$, of m i.i.d. data samples, $i=1,\ldots,m$, and a real-valued function class $\mathscr{H}_k$ that corresponds to the k-th component of the class $\mathscr{H}$ of vector-valued functions, the scaled empirical Rademacher complexity $m\mathscr{R}_S(\mathscr{H}_k)=\mathbb{E}\left[\sup_{h\in\mathscr{H}_k}\sum_{i=1}^{m}\epsilon_i h(\boldsymbol{x}_i)\right]$ satisfies the following inequality:

$$\begin{aligned}m\mathscr{R}_S(\mathscr{H}_k)&=\frac{1}{\lambda}\log\exp\left(\lambda\mathbb{E}\left[\sup_{h\in\mathscr{H}_k}\sum_{i=1}^{m}\epsilon_i h(\boldsymbol{x}_i)\right]\right)\\&\le\frac{1}{\lambda}\log\left(\mathbb{E}\left[\sup_{h\in\mathscr{H}_k}\exp\left(\lambda\sum_{i=1}^{m}\epsilon_i h(\boldsymbol{x}_i)\right)\right]\right),\end{aligned} \tag{12}$$

where λ is an arbitrary positive real number.

Additionally, since the RNN models of Equations (3) and (4) are essentially complex nonlinear functions that are difficult to measure the learning capacity, the following lemma provides a useful tool to peel off RNN weights and nonlinear activation functions through layers.

Lemma 3. (c.f. Lemma 6 in Wu et al.[7]) Given any monotonically increasing and convex function $p:\mathbf{R}\to\mathbf{R}_+$, and a vector-valued RNN function class $\mathscr{H}$ with a positive-homogeneous, 1-Lipschitz, activation function $\sigma_h(\cdot)$, the inequality below holds:

$$\begin{aligned}&\mathbb{E}\left[\sup_{\|W\|_F\le B_{W,F},\|U\|_F\le B_{U,F},h\in\mathscr{H}}p\left(\left|\sum_{i=1}^{m}\epsilon_i\boldsymbol{h}_{i,t}\right|\right)\right]\\&\le 2\mathbb{E}\left[\sup_{h\in\mathscr{H}}p\left(B_{W,F}\left|\sum_{i=1}^{m}\epsilon_i\boldsymbol{x}_{i,t}\right|+B_{U,F}\left|\sum_{i=1}^{m}\epsilon_i\boldsymbol{h}_{i,t-1}\right|\right)\right]\end{aligned} \tag{13}$$

Based on Lemma 3, the following lemma derives a Rademacher complexity bound for the real-valued RNN function class $\mathscr{H}_k$ that corresponds to the k-th output of the class $\mathscr{H}$ of vector-valued functions.

Lemma 4. (c.f. Lemma 7 in Wu et al.[7]) Given a dataset $S = (\boldsymbol{x}_{i,t}, \boldsymbol{y}_{i,t})_{t=1}^{T}$, of m i.i.d. data samples, $i = 1,\ldots,m$, and a class of real-valued functions, $\mathscr{H}_{k,t}$, $k = 1,\ldots,d_y$, that corresponds to the k-th output at t-th time step, and satisfy Assumptions 1–4, the Rademacher complexity can be bounded using the following inequality:

$$\mathscr{R}_S(\mathscr{H}_{k,t}) \le \frac{MB_X\left(1+\sqrt{2\log(2)t}\right)}{\sqrt{m}} \tag{14}$$

where $M = \frac{1-(B_{U,F})^t}{1-B_{U,F}} B_{W,F} B_{V,F}$.

Finally, we consider the class of loss function for the vector-valued RNN models, and use the contraction inequality in Maurer[19] to further bound the RNN generalization error as follows.

Theorem 1. (c.f. Theorem 1 in Wu et al.[7]) Given a dataset $S = (\boldsymbol{x}_{i,t}, \boldsymbol{y}_{i,t})_{t=1}^{T}$ with i.i.d. data samples, $i = 1,\ldots,m$, and the loss function class $\mathcal{G}_t$ associated with the RNN function class $\mathscr{H}_t$ that predicts outputs at the t-th time step, with probability at least $1-\delta$ over S, the following inequality holds for the RNN models with the activation functions and weight matrices that satisfy Assumptions 1–4.

$$\mathbb{E}[g_t(\boldsymbol{x},\boldsymbol{y})] \le \mathcal{O}\left(L_r d_y \frac{MB_X\left(1+\sqrt{2\log(2)t}\right)}{\sqrt{m}}\right) + 3\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m}} + \frac{1}{m}\sum_{i=1}^{m} g_t(\boldsymbol{x}_i,\boldsymbol{y}_i), \tag{15}$$

where M is given in Equation (14), B_X is the RNN input bound defined in Equation (1), L_r is the local Lipschitz constant defined in Equation (8), and d_y is the RNN output dimension.

Remark 2. Based on Theorem 1, to minimize the error between the actual process dynamics and the RNN model, we should first collect as much data as possible since the generalization error of Equation (15) decreases with increasing training samples m. Subsequently, we need to carefully tune the RNN hyper-parameters (e.g., weight bounds, and the number of layers and neurons) to achieve a desired training and validation performance. It should be noted that a complex RNN model can generally learn the training data well, but may not generalize well due to overfitting. Therefore, to reduce overfitting, a number of approaches such as regularization, dropout and early stopping methods can be adopted to improve the generalization performance of RNN models.

Remark 3. The RNN generalization error bound of Equation (15) implies that with a larger δ (i.e., a lower probability $1-\delta$), the RNN generalization error is less likely to be bounded by a tighter bound. Therefore, the choice of δ really depends on the level of trust of this bound. To derive a tighter bound with higher confidence, we essentially need to reduce the other two terms in Equation (15) by increasing the training sample size m and carefully training the RNN models to obtain a desired training error. It should be mentioned that the choice of δ affects the generalization error bound in the sense that a tighter bound implies less likelihood, but does not directly affect the generalization performance after the training is completed since δ is not involved in the RNN training process. In practice, we first require the RNN generalization error to be bounded by a constant or a function of state (e.g., $E_M \le \gamma(|x|)$ in the next section) such that MPC stability results hold. Based on this bound, we then tune the RNN hyperparameters and execute the training process to achieve a desired training performance and model complexity. Finally, we can approximate the probability $1-\delta$ by calculating how many testing samples meet the generalization error bound.

3 | PROBABILISTIC STABILITY ANALYSIS

The RNN models are incorporated within MPC in this section to provide the prediction of future states. We develop the MPC scheme using RNN models (RNN-MPC), and study the system stability properties for the nonlinear system of Equation (1) subject to disturbances. We demonstrate that under RNN-MPC, the state of the closed-loop system remains inside the stability region in probability for all times in the presence of process disturbances.

The single-hidden-layer RNN model is represented as a continuous-time nonlinear system for simplifying the analysis of its stability properties[11]:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T z, \tag{16}$$

where $u \in \mathbf{R}^k$ is the manipulated input, and $\hat{x} \in \mathbf{R}^n$ is the RNN state vector. A and Θ are the weight matrices, and z is a vector associated with the $\hat{x}$ and u. The readers are referred to Wu et al.[7] for the details of Equation (16). In the following sections, the Lyapunov-based MPC schemes using RNN models are designed to stabilize the nonlinear system in a probabilistic manner in the presence of process disturbances. Specifically, we consider two types of disturbances: unknown-but-bounded disturbances and stochastic disturbances with unbounded variation, and establish the probabilistic closed-loop stability results for the nonlinear systems under RNN-MPC.

3.1 | Nonlinear systems with bounded disturbances

The class of continuous-time nonlinear systems subject to bounded disturbances is described by the following ordinary differential equation:

$$\dot{x} = F(x,\ u,\ w) := f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{17}$$

where the notations are the same as those in Equation (1). The disturbance vector w is bounded by $W := \{w \in \mathbf{R}^q \mid |w| \le w_m|\ \ w_m \ge 0\}$. $h(x)$ is a sufficiently smooth matrix function of dimension $n \times q$. Based on the boundedness of x, u, and w, and the Lipschitz property of $F(x, u, w)$, there exist positive constants M_F, L_x, $L_x^{'}$, L_w, $L_w^{'}$ such that for all $w \in W$, $u \in U$, $x,x' \in \Omega_\rho$, the following inequalities hold:

$$|F(x',\ u,\ 0) - F(x,\ u,\ w)| \le L_w |w| + L_x |x - x'| \tag{18a}$$

$$\left|\frac{\partial V(x')}{\partial x} F(x',\ u,\ 0) - \frac{\partial V(x)}{\partial x} F(x,\ u,\ w)\right| \le L_w^{'} |w| + L_x^{'} |x - x'| \tag{18b}$$

$$|F(x,\ u,\ w)| \le M_F \tag{18c}$$

Under the assumption of exponential stabilization of the origin of the RNN model of Equation (16) for states in an open set $\widehat{D}$ around the origin by a feedback controller $u = \Phi_{nn}(x) \in U$, a $\mathcal{C}^1$ Lyapunov function $\widehat{V}(x)$ can be found such that for all states x in $\widehat{D}$, the following inequalities hold:

$$\left|\frac{\partial \widehat{V}(x)}{\partial x}\right| \le \widehat{c}_4 |x|, \tag{19a}$$

$$\frac{\partial \widehat{V}(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \le -\widehat{c}_3 |x|^2, \tag{19b}$$

$$\widehat{c}_1 |x|^2 \le \widehat{V}(x) \le \widehat{c}_2 |x|^2, \tag{19c}$$

where $\widehat{c}_i$, $i = 1, 2, 3, 4$ are positive constants. Similarly, we characterize the stability region for the RNN model of Equation (16) as a compact set embedded in $\widehat{D}$ as follows: $\Omega_{\widehat{\rho}} := \left\{x \in \widehat{D} \mid \widehat{V}(x) \le \widehat{\rho}\right\}$, where $\widehat{\rho} > 0$. The following proposition demonstrates that for the RNN model developed with a sufficiently small modeling error, the nominal system of Equation (17) with $w(t) \equiv 0$ can be stabilized under $u = \Phi_{nn}(x) \in U$ with high probability.

Proposition 1. Consider the nominal system of Equation (17) with $w(t) \equiv 0$ and an RNN model that is trained with m i.i.d. data samples and satisfies the conditions in Theorem 1. Under the stabilization assumption of Equation (19), if there exists a positive real number γ that satisfies $\gamma < \widehat{c}_3/\widehat{c}_4$, and constrains the modeling error, i.e., $|F_{nn}(x,\ u) - F(x,\ u,\ 0)| \le \gamma |x|$, for all $u \in U$ and $x \in \Omega_{\widehat{\rho}}$, then for all $x \in \Omega_{\widehat{\rho}}$, the origin of the nominal system is rendered exponentially stable with probability at least $1 - \delta$ under $u = \Phi_{nn}(x) \in U$.

Proof. Following the proof of Proposition 2 in Wu et al.,[11] the time derivative of $\widehat{V}$ is obtained as follows using Equations (19a) and (19b):

$$\begin{aligned}\dot{\widehat{V}} &= \frac{\partial \widehat{V}(x)}{\partial x} F(x,\ \Phi_{nn}(x),\ 0) \\ &\le\ \widehat{c}_4 |x| \cdot |F(x,\ \Phi_{nn}(x),\ 0) - F_{nn}(x,\ \Phi_{nn}(x))| - \widehat{c}_3 |x|^2,\end{aligned} \tag{20}$$

where the term $|F(x,\ \Phi_{nn}(x),\ 0) - F_{nn}(x,\ \Phi_{nn}(x))|$ is the modeling mismatch between the nominal system of Equation (17) and the RNN model. Using the generalization error results derived in Theorem 1, we have the following bound for the modeling error:

$$|F_{nn}(x,\ \Phi_{nn}(x)) - F(x,\ \Phi_{nn}(x),\ 0)| \le E_M, \tag{21}$$

where E_M represents the generalization error that is bounded by the RHS of Equation (15). Since Equation (15) depends on the training sample size, we can find the minimum data sample size $m_N(|x|,\ h_c,\ \delta)$ such that the modeling error is upper bounded by $E_M \le \gamma |x|$. By choosing the sample size $m \ge m_N(|x|,\ h_c,\ \delta)$, the following equation holds for $\dot{\widehat{V}}(x)$ $(x \ne 0)$, with probability at least $1 - \delta$.

$$\begin{aligned}\dot{\widehat{V}} &= \frac{\partial \widehat{V}}{\partial x}(F_{nn}(x,\ \Phi_{nn}(x)) + F(x,\ \Phi_{nn}(x),\ 0) - F_{nn}(x,\ \Phi_{nn}(x))) \\ &\le\ -\widehat{c}_3 |x|^2 + |F_{nn}(x,\ \Phi_{nn}(x)) - F(x,\ \Phi_{nn}(x),\ 0)| \cdot \widehat{c}_4 |x| \\ &\le\ -\widehat{c}_3 |x|^2 + \frac{\widehat{c}_3 |x|}{\widehat{c}_4} \cdot \widehat{c}_4 |x| \\ &\le\ -\widetilde{c}_3 |x|^2 \\ &< 0\end{aligned} \tag{22}$$

where $\widetilde{c}_3 = -\widehat{c}_3 + \widehat{c}_4 \gamma < 0$ for any $\gamma < \widehat{c}_3/\widehat{c}_4$. This implies that with a certain probability, $\dot{\widehat{V}}$ can be rendered negative (i.e., $\mathbb{P}\left[\dot{\widehat{V}} < 0\right] \ge 1 - \delta$), and therefore, the state of the nominal system of Equation (17) moves toward the origin under $u = \Phi_{nn}(x) \in U$ for all $x_0 \in \Omega_{\widehat{\rho}}$.

Remark 4. Note that the minimum data sample size $m_N(|x|, h_c, \delta)$ that satisfies $E_M \le \gamma |x|$ is a function of $|x|$, h_c, and δ. Specifically, by substituting the RHS of Equation (15) into E_M, it is straightforward to show that the solution to $E_M \le \gamma |x|$ is a function of the confidence level δ. Additionally, it is observed from $E_M \le \gamma |x|$ that the solution depends on the value of x in the way that a smaller x value (i.e., the states closer to the origin) leads to a tighter generalization error bound E_M, which requires more data to be used for training. In the extreme case where the state x is sufficiently close to the origin, a large number of data is needed to render E_M sufficiently low in order to meet the condition $E_M \le \gamma |x|$, which is computationally impracticable in general. Therefore, to reduce the computational time in training RNN models, we do not require this condition to hold in a small neighborhood around the origin, and we will demonstrate in Theorem 2 that closed-loop stability remains unaffected under RNN-based MPC despite this loose condition. Lastly, $m_N(\cdot)$ is also a

function of h_c since we calculate the modeling error of Equation (21) by approximating the derivatives using finite differences with a sufficiently small time step h_c.

Proposition 2. Consider the RNN model $\dot{\widehat{x}} = F_{nn}(\widehat{x},u)$ of Equation (16) developed satisfying Assumptions 1–4 and the nonlinear system $\dot{x} = F(x,u,w)$ of Equation (17) with bounded disturbances $|w| \le w_m$. There exists a positive constant κ and a class $\mathcal{K}$ function $f_w(\cdot)$ such that for all $x, \widehat{x} \in \Omega_{\widehat{\rho}}$, with probability at least $1-\delta$, the following inequalities hold with $x_0 = \widehat{x}_0 \in \Omega_{\widehat{\rho}}$ (i.e., the same initial condition):

$$|x(t) - \widehat{x}(t)| \le f_w(t) := \frac{L_w w_m + E_M}{L_x}\left(e^{L_x t} - 1\right) \tag{23a}$$

$$\widehat{V}(x) \le \kappa |x - \widehat{x}|^2 + \frac{\widehat{c}_4\sqrt{\widehat{\rho}}}{\sqrt{\widehat{c}_1}}|x - \widehat{x}| + \widehat{V}(\widehat{x}) \tag{23b}$$

Proof. The results are derived following the proof technique used in Proposition 3 in Wu et al.[11] Note that in Wu et al.,[11] the modeling error is assumed to be bounded by a constant number almost surely (i.e., with probability 1). However, the modeling error $|F(\widehat{x},\, u,\, 0) - F_{nn}(x,\, u)|$ in this work is bounded by E_M with probability at least $1-\delta$ since we use the generalization error to represent the model mismatch for any states in the stability region including those which are not used in training. The key steps for the proof are presented below. We first define an error vector $e(t) = x(t) - \widehat{x}(t)$ that represents the modeling error between the RNN model state $\dot{\widehat{x}} = F_{nn}(\widehat{x},\, u)$ and the actual nonlinear system state $\dot{x} = F(x,\, u,\, w)$ subject to bounded disturbances. The time derivative of error vector is bounded for all $\widehat{x},\ x \in \Omega_{\widehat{\rho}}$, $w(t) \in W$, and $u \in U$ as follows:

$$\begin{aligned} |\dot{e}| &= |F(x,\, u,\, w) - F_{nn}(\widehat{x},\, u)| \\ &\le |F(x,\, u,\, w) - F(\widehat{x},\, u,\, 0)| + |F(\widehat{x},\, u,\, 0) - F_{nn}(\widehat{x},\, u)|, \\ &\le L_x |e(t)| + L_w w_m + E_M \end{aligned} \tag{24}$$

where the last inequality is obtained using Equation (18a) and the modeling error constraint of Equation (21). Since the initial error $e(0)$ is zero for the same initial condition $x_0 = \widehat{x}_0$, the evolution of error vector is bounded as follows:

$$|e(t)| \le \frac{L_w w_m + E_M}{L_x}\left(e^{L_x t} - 1\right) \tag{25}$$

Additionally, using Taylor series expansion and ignoring higher-order terms, we have

$$\begin{aligned} \widehat{V}(x) &\le \widehat{V}(\widehat{x}) + \frac{\partial \widehat{V}(\widehat{x})}{\partial x}|x - \widehat{x}| + \kappa |x - \widehat{x}|^2 \\ &\le \widehat{V}(\widehat{x}) + \frac{\widehat{c}_4\sqrt{\widehat{\rho}}}{\sqrt{\widehat{c}_1}}|x - \widehat{x}| + \kappa |x - \widehat{x}|^2 \end{aligned}, \tag{26}$$

where the last inequality is obtained using Equations (19a), (19c), and κ is a positive real number. This completes the proof of Proposition 2.

Subsequently, the RNN models are utilized within MPC to provide the prediction of future state evolution. The optimization problem of RNN-MPC is presented as follows[11,20]:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_{\mathrm{MPC}}(\widetilde{x}(t),\, u(t))dt \tag{27a}$$

$$\text{s.t. } \dot{\widetilde{x}}(t) = F_{nn}(\widetilde{x}(t),\, u(t)) \tag{27b}$$

$$u(t) \in U,\ \forall t \in [t_k,\, t_{k+N}) \tag{27c}$$

$$\widetilde{x}(t_k) = x(t_k) \tag{27d}$$

$$\dot{\widehat{V}}(x(t_k),\, u) \le \dot{\widehat{V}}(x(t_k),\, \Phi_{nn}(x(t_k))),$$

$$\text{if } x(t_k) \in \Omega_{\widehat{\rho}} \setminus \Omega_{\rho_{nn}} \tag{27e}$$

$$\widehat{V}(\widetilde{x}(t)) \le \rho_{nn},\ \forall t \in [t_k,\, t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_{nn}}, \tag{27f}$$

where L_{MPC} denotes the objective function of MPC that attains its minimum value at the origin. $\widetilde{x}$ is the state predicted by the RNN model $F_{nn}(x,u)$. The RNN-MPC of Equation (27) is applied in a sample-and-hold fashion in which control actions remain unchanged within each sampling period, with control actions optimized over the prediction horizon from t_k to t_{k+N} as a piece-wise function in $S(\Delta)$. The goal of RNN-MPC is to stabilize the nonlinear system at the steady-state in the way that the state is maintained in the stability region $\Omega_{\widehat{\rho}}$ at all times, and is driven into a small terminal set around the origin ultimately. Equation (27b) is the prediction model, and Equation (27c) is the input constraint. The feedback measurement of x at each sampling time is utilized as the initial state for solving Equation (27b). The two Lyapunov-based constraints of Equations (27e) and (27f) ensure stability for the closed-loop system under RNN-MPC. The following theorem establishes the closed-loop stability properties for the uncertain system of Equation (17) subject to bounded disturbances $w \in W$ under RNN-MPC.

Theorem 2. Consider the nonlinear system of Equation 17 under the RNN-MPC of Equation (27) with the controller $\Phi_{nn}(x)$ that meets Equation (19). Let $\Delta > 0$, $\widehat{\rho} > \rho_{\min} > \rho_{nn} > \rho_s$ and $\epsilon_w > 0$ satisfy Equations (28) and (29).

$$-\frac{\widetilde{c}_3}{\widehat{c}_2}\rho_s + L_x' M_F \Delta + L'_w w_m \le -\epsilon_w \tag{28}$$

and

$$\rho_{nn} := \max\left\{ \widehat{V}(\widehat{x}(t+\Delta)) \mid u \in U,\ \widehat{x}(t) \in \Omega_{\rho_s} \right\} \tag{29a}$$

$$\rho_{\min} \ge \rho_{nn} + \frac{\widehat{c}_4 \sqrt{\widehat{\rho}}}{\sqrt{\widehat{c}_1}} f_w(\Delta) + \kappa(f_w(\Delta))^2, \tag{29b}$$

where $f_w(t) = \frac{E_M + L_w w_m}{L_x}\left(e^{L_x t} - 1\right)$. Then, by choosing the sample size m to be greater than the minimum sample size $m_N(|x|, h_c, \delta)$ that satisfies $E_M \le \gamma |x|$, for any initial state $x_0 \in \Omega_{\hat{\rho}}$ and for each sampling time step, system stability is achieved for the disturbed system of Equation (17) with $w \in W$ with probability at least $1-\delta$, under the RNN-MPC of Equation (27) in the sense that $x(t)$ ultimately converges to $\Omega_{\rho_{\min}}$, and is maintained in $\Omega_{\widehat{\rho}}$ at all times, that is, $x(t) \in \Omega_{\widehat{\rho}}, \forall t \ge 0$.

Proof. The proof follows the proofs of Proposition 3 and Theorem 2 in Wu et al.[7] for the nominal system of Equation (17) with $w(t) \equiv 0$, and we present a proof sketch here to help readers understand the key steps. The main difference is that the system of Equation (17) is subject to sufficiently small bounded disturbances $|w(t)| \le w_m$ in this work, which needs to be accounted for in the controller design and stability analysis, while in Wu et al.,[7] the results were developed for the nominal system of Equation (17) with $w(t) \equiv 0$. We first obtain the time derivative of $\widehat{V}$ for any $x(t_k) \in \Omega_{\widehat{\rho}} \setminus \Omega_{\rho_s}$ under the controller $u = \Phi_{nn}(x) \in U$:

$$\begin{aligned}
\dot{\widehat{V}}(x(t)) &= \frac{\partial \widehat{V}(x(t))}{\partial x} F(x(t),\ \Phi_{nn}(x(t_k)),\ w) \\
&= \frac{\partial \widehat{V}(x(t_k))}{\partial x} F(x(t_k),\ \Phi_{nn}(x(t_k)),\ 0) \\
&\quad + \frac{\partial \widehat{V}(x(t))}{\partial x} F(x(t),\ \Phi_{nn}(x(t_k)),\ w) \\
&\quad - \frac{\partial \widehat{V}(x(t_k))}{\partial x} F(x(t_k),\ \Phi_{nn}(x(t_k)),\ 0)
\end{aligned} \tag{30}$$

Using the results in Proposition 1 and Equation (18b), we can further bound Equation (30) for $x(t) \in \Omega_{\widehat{\rho}} \setminus \Omega_{\rho_s}$, $u \in U$, and $w \in W$ as follows:

$$\begin{aligned}
\dot{\widehat{V}}(x(t)) &\le \frac{\partial \widehat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) - \frac{\widetilde{c}_3}{\widehat{c}_2}\rho_s \\
&\quad - \frac{\partial \widehat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\
&\le -\frac{\widetilde{c}_3}{\widehat{c}_2}\rho_s + L_x' M_F \Delta + L_w' w_m
\end{aligned} \tag{31}$$

Therefore, if Equation (28) is satisfied, with probability at least $1-\delta$, $\dot{\widehat{V}}(x(t))$ is rendered negative for any $x(t_k) \in \Omega_{\widehat{\rho}} \setminus \Omega_{\rho_s}$, which implies the convergence of state toward the terminal set, as well as the boundedness of state within $\Omega_{\widehat{\rho}}$ in probability. Additionally, $\Omega_{\rho_{\min}}$ and $\Omega_{\rho_{nn}}$ of Equation (29) are the two small sets around the origin which contain Ω_{ρ_s} as their subsets (i.e., $\rho_{\min} > \rho_{nn} > \rho_s$). Specifically, we do not require the modeling error to be bounded by E_M within Ω_{ρ_s}, and thus, Equation (31) does not hold for the state in Ω_{ρ_s}. This loose condition of modeling error within Ω_{ρ_s} significantly reduces the computational complexity for training RNN models with the data sufficiently close to the origin. The set $\Omega_{\rho_{nn}}$ is characterized to ensure that for any state within Ω_{ρ_s}, the predicted state remains inside $\Omega_{\rho_{nn}}$ under any control actions within the bounds. As a result, the state of the actual nonlinear system of Equation (17) subject to disturbances $w(t) \in W$ is bounded in the set $\Omega_{\rho_{\min}}$ that is characterized accounting for the modeling error and disturbances.

3.2 | Nonlinear systems with stochastic disturbances

In addition to the robustness treatment of the process disturbances as bounded uncertain variables, another approach to dealing with model uncertainty is to develop controllers that achieve stability in probability for the closed-loop system by modeling the disturbance terms in a probabilistic manner and taking the distribution information of disturbances into account. Specifically, we consider the nonlinear system with stochastic disturbances in the form of the following stochastic differential equation (SDE):

$$dx(t) = f(x(t))dt + g(x(t))u(t)dt + h(x(t))dw(t), \tag{32}$$

where the notations follows those in Equation (1). The disturbance vector $w(t)$ is represented by a standard Wiener process. The steady-state of the nominal system with $w(t) \equiv 0$ is assumed to be at the origin, that is, $(x_s^*,\ u_s^*) = (0,\ 0)$. In Equation (1), $f(x(t)) + g(x(t))u(t)$ and $h(x(t))$ are the deterministic drift and the diffusion matrix, respectively. $h(0)$ is assumed to be zero such that $h(x(t))dw(t)$ (i.e., the disturbance term) vanishes at the origin. Similarly, we assume that $L_g V(x)$, $L_f V(x)$, and $h(x)^T \frac{\partial^2 V(x)}{\partial x^2} h(x)$ are locally Lipschitz. For the system of Equation (32), if for any $\varepsilon > 0$, the following conditions hold, then the origin is asymptotically stable in probability.

$$\begin{aligned}
\lim_{x(0)\to 0} \mathbb{P}\left(\lim_{t\to\infty} |x(t)| = 0 \right) &= 1, \\
\lim_{x(0)\to 0} \mathbb{P}\left(\sup_{t \ge 0} |x(t)| > \varepsilon \right) &= 0
\end{aligned} \tag{33}$$

It is assumed that a stochastic feedback controller $u = \Phi_s(x) \in U$ exists such that for all $x \in D_2 \subset \mathbf{R}^n$ (D_2 is an open neighborhood of the origin), exponential stabilization of the origin of the RNN model of Equation (16) is achieved in probability. The stabilizability assumption implies that a C^2, positive definite stochastic Lyapunov function V exists and meets the following conditions:

$$\mathscr{L}\mathrm{V}(\mathrm{x})=\frac{\partial \mathrm{V}(\mathrm{x})}{\partial \mathrm{x}}\mathrm{F}_{nn}(\mathrm{x},\ \Phi_{\mathrm{s}}(\mathrm{x}))+\frac{1}{2}Tr\left\{\mathrm{h}^{\mathrm{T}}\frac{\partial^2 \mathrm{V}}{\partial \mathrm{x}^2}\mathrm{h}\right\}\le -\alpha_1|\mathrm{x}|^2 \tag{34}$$

$$h(x)^T\frac{\partial^2 V}{\partial x^2}h(x)\ge 0, \tag{35}$$

where $\mathscr{L}\mathrm{V}(\mathrm{x})$ represents the infinitesimal generator of the system of Equation (32), and α_1 is a positive real number. $\phi_d=\{x\in\mathbf{R}^n \mid \mathscr{L}\mathrm{V}+\kappa V(\mathrm{x})\le 0,\ \kappa>0,\ \mathrm{u}=\Phi_s(\mathrm{x})\in \mathrm{U}\}$ is characterized as a set of initial conditions from which the exponential stabilization of the origin of the RNN model of Equation (16) can be achieved in probability using the controller $u=\Phi_s(x)\in U$. Subsequently, a level set of $V(x)$ inside ϕ_d, that is, $\Omega_\rho:=\{x\in\phi_d \mid V(x)\le\rho\}$, $\rho>0$, is chosen as the operating region.

Based on $u=\Phi_s(x)\in U$, the following Lyapunov-based MPC scheme is designed to stabilize the stochastic nonlinear system of Equation (32), where the notations follow those in Equation (27).

$$\min_{u\in S(\Delta)}\int_{t_k}^{t_{k+N}} L_{\mathrm{MPC}}(\widetilde{x}(t),\ u(t))dt \tag{36a}$$

$$\text{s.t. } \dot{\widetilde{x}}(t)=F_{nn}(\widetilde{x}(t),\ u(t)) \tag{36b}$$

$$u(t)\in U,\quad \forall t\in[t_k,\ t_{k+N}) \tag{36c}$$

$$\widetilde{x}(t_k)=x(t_k) \tag{36d}$$

$$\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t_k}),\ \mathrm{u}(\mathrm{t_k}))\le\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t_k}),\ \Phi_{\mathrm{s}}(\mathrm{x}(\mathrm{t_k}))),\text{if } \mathrm{x}(\mathrm{t_k})\in\Omega_\rho\setminus\Omega^{\mathrm{o}}_{\rho_{nn}} \tag{36e}$$

$$V(\widetilde{x}(t))<\rho_{nn},\ \forall t\in[t_k,\ t_{k+N}),\text{if } x(t_k)\in\Omega^{\mathrm{o}}_{\rho_{nn}} \tag{36f}$$

Theorem 3 establishes the probabilistic stability properties for the uncertain system of Equation (32) under the RNN-MPC of Equation (36).

Theorem 3. Consider the system of Equation (32) under the MPC of Equation (36) using RNN models that meet the Assumptions 1–4. By letting $m\ge m_N(|x|,\ h_c,\ \delta)$, given any initial condition $x(0)\in\Omega_\rho$, probability $\lambda\in(0,1]$, and positive real numbers satisfying $\rho>\rho_{\min}>\rho_{nn}$ and $\rho_c\in[\rho_{nn},\rho]$, there exists a sampling period $\Delta>0$ and probabilities $\beta,\gamma\in[0,1]$ such that the following inequalities hold for $t\in[t_k,t_{k+1})$, $t_{k+1}:=t_k+\Delta$.

$$\mathbb{P}\left(\sup_{t\in[t_k,t_{k+1})} V(x(t))<\rho_{\min}\right)\ge(1-\delta)(1-\beta)(1-\lambda),\ \forall x(t_k)\in\Omega_{\rho_{nn}} \tag{37}$$

$$\mathbb{P}\left(\tau_{\mathbf{R}^n\setminus\Omega^{\mathrm{o}}_{\rho_{nn}}}<\tau_{\Omega_\rho}\right)\ge(1-\delta)(1-\gamma)(1-\lambda),\ \forall x(t_k)\in\Omega_{\rho_c}\setminus\Omega^{\mathrm{o}}_{\rho_{nn}} \tag{38}$$

where

$$\frac{\sup_{x\in\partial\Omega_{\rho_{nn}}}V(x)}{\inf_{x\in\mathbf{R}^n\setminus\Omega_{\rho_{\min}}}V(x)}\le\beta \tag{39a}$$

$$\sup_{x\in\Omega_{\rho_c}\setminus\Omega^{\mathrm{o}}_{\rho_{nn}}}\frac{V(x)}{\rho}\le\gamma \tag{39b}$$

Proof. The two probabilities in Equations (37) and (38) can be interpreted as follows. Equation (37) gives the probability that the future state remains inside $\Omega_{\rho_{\min}}$ (i.e., the terminal set around the origin) during one sampling period for any initial state inside $\Omega_{\rho_{nn}}$ ($\Omega_{\rho_{nn}}$ is a subset of $\Omega_{\rho_{\min}}$ as defined in Equation 29). Equation (38) is the probability that the state hits the boundary of $\Omega_{\rho_{nn}}$ before it leaves the stability region Ω_ρ for any initial states in $\Omega_\rho\setminus\Omega_{\rho_{nn}}$. The proof consists of three parts. In the first part, we demonstrate that the infinitesimal generator $\mathscr{L}\mathrm{V}$ for the stochastic system of Equation (32) is rendered negative (Equation 34) under the controller $u=\Phi_s(x)\in U$ for all $x\in\Omega_\rho$ with a certain probability, which is a key step in the derivation of Equations (37) and (38). Specifically, Equation (34) holds almost surely for the RNN model since the stability region Ω_ρ is characterized using the RNN model and the controller $u=\Phi_s(x)\in U$; however, due to the existence of generalization error, Equation (34) holds for the nonlinear system of Equation (32) in a probabilistic manner (i.e., with probability at least $1-\delta$), which leads to the probability of $1-\delta$ on the RHS of Equations (37) and (38). Furthermore, the final probability of $\mathscr{L}\mathrm{V}$ being negative also needs to account for the impact of stochastic, unbounded disturbance. In the second part, we prove the probabilities in Equations (37) and (38), and demonstrate that the probability terms (i.e., $1-\beta$, $1-\gamma$, and $1-\lambda$) depend on the size of the sets Ω_ρ, $\Omega_{\rho_{nn}}$, $\Omega_{\rho_{\min}}$ and the sampling period. Finally, in the third part, we prove probabilistic closed-loop stability for the stochastic nonlinear system of Equation (32) under the RNN-MPC of Equation (36).

Part 1 : We first prove that there exists a sufficiently small sampling period Δ such that $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$ can be rendered negative for all the states $x(t_k)\in\Omega_\rho\setminus\Omega^{\mathrm{o}}_{\rho_s}$ in probability. Following the proof of Theorem 1 in Wu et al.,[16] we define an event that the disturbance $w(t)$ is bounded within one sampling period as follows: $A_B:=\{w\in\mathbf{R}^q \mid \sup_{t\in[t_k,t_k+\Delta)}|w(t)|\le B\}$. Then, there exists a sufficiently small ball B such that $P(A_B)=1-\lambda$ holds for any probability $\lambda=(0,1]$ under the disturbance $w(t)$ following standard Brownian motion. As a result, for $w\in A_B$, the probability $\mathbb{P}(A_W)\ge 1-\lambda$ holds for the event $A_W:=\{\sup_{t\in[t_k,t_k+\Delta)}|x(t)-x(t_k)|\le k_1(\Delta)^r\}$, $k_1>0$, $r<\frac{1}{2}$, which states that the state evolution is bounded within one sampling period in the presence of a bounded disturbance.[21] Subsequently, we prove that $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$ is negative for any $x(t_k)\in\Omega_\rho\setminus\Omega^{\mathrm{o}}_{\rho_s}$ accounting for the modeling error between RNN model and the nominal

system of Equation (32). Specifically, by letting the training sample size $m \geq m_N(|x|,\ h_c,\ \delta)$ such that the modeling error is constrained by $\gamma|x|$ for any $x(t_k) \in \Omega_\rho \setminus \Omega^o_{\rho_s}$, the following equation holds under $u = \Phi_s(x) \in U$, with probability no less than $1-\delta$.

$$\begin{aligned}\mathscr{L}V(x(t)) &= \frac{\partial V}{\partial x}(F_{nn}(x,\Phi_s(x)) + F(x,\Phi_s(x),0) - F_{nn}(x,\Phi_s(x))) \\ &\quad + \frac{1}{2}Tr\left\{h(x)^T \frac{\partial^2 V(x)}{\partial x^2} h(x)\right\} \\ &\leq -\alpha_1|x|^2 + |F_{nn}(x,\Phi_s(x)) - F(x,\Phi_s(x),0)| \cdot \widehat{c}_4|x| \\ &\leq -\alpha_1|x|^2 + \gamma|x| \cdot \widehat{c}_4|x| \leq -\widetilde{\alpha}_1|x|^2 < 0,\end{aligned} \tag{40}$$

where $\widetilde{\alpha}_1 = -\alpha_1 + \widehat{c}_4\gamma < 0$ for any $\gamma < \alpha_1/\widehat{c}_4$. Equation (40) shows that under the stochastic stabilizing controller $u = \Phi_s(x)$ designed for the RNN model, the infinitesimal generator $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$ for the stochastic nonlinear system of Equation (32) is rendered negative for $x \in \Omega_\rho \setminus \Omega^o_{\rho_s}$ with probability at least $1-\delta$, provided that the training sample size is chosen appropriately to ensure a sufficiently small and bounded modeling error. As a result, we can find a positive real number κ such that $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t})) \leq -\kappa V(\mathrm{x})$ holds for any $x \in \Omega_\rho \setminus \Omega^o_{\rho_s}$ with probability at least $1-\delta$.Subsequently, we prove under sample-and-hold implementation (i.e., $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$), there exists a sufficiently small sampling period Δ such that $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$ can be rendered negative within one sampling period. Specifically, since $\mathscr{L}\mathrm{V}(\mathrm{x}) = \mathrm{L_f V}(\mathrm{x}) + \mathrm{L_g V}(\mathrm{x}) + \frac{1}{2}Tr\left\{h(x)^T \frac{\partial^2 V(x)}{\partial x^2} h(x)\right\}$, and $L_fV(x)$, $L_gV(x)$, $h(x)^T \frac{\partial^2 V(x)}{\partial x^2} h(x)$ are locally Lipschitz, there exist positive real numbers $k_3,\ k_4,\ k_5$ such that the following equations hold.

$$\begin{aligned}&|L_fV(x(t)) - L_fV(x(t_k))| \leq k_3|x(t) - x(t_k)| \\ &\left|L_gV(x(t))u(t_k) - L_gV(x(t_k))u(t_k)\right| \leq k_4|x(t) - x(t_k)| \\ &\left|\frac{1}{2}Tr\left\{h(x(t))^T \frac{\partial^2 V(x(t))}{\partial x^2} h(x(t))\right\}\right. \\ &\left.-\frac{1}{2}Tr\left\{h(x(t_k))^T \frac{\partial^2 V(x(t_k))}{\partial x^2} h(x(t_k))\right\}\right| \\ &\leq k_5|x(t) - x(t_k)|\end{aligned} \tag{41}$$

Using the results from Equation (40), we obtain the following inequality for $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$, $\forall x \in \Omega_\rho \setminus \Omega^o_{\rho_s}$ under $u(t) = \Phi_s(x(t_k)) \in U$, $\forall t \in [t_k, t_{k+1})$:

$$\begin{aligned}\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t})) &= \mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t_k})) + (\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t})) - \mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t_k}))) \\ &\leq -\kappa\rho_s + (k_3 + k_4 + k_5)|x(t) - x(t_k)|\end{aligned} \tag{42}$$

Therefore, for any $w \in A_B$ with $\mathbb{P}(A_W) \geq 1-\lambda$, and $\Delta < \left(\frac{\kappa\rho_s - \epsilon}{k_1(k_3+k_4+k_5)}\right)^{\left(\frac{1}{r}\right)}$, we have $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t})) < -\epsilon$, for all $t \in [t_k, t_{k+1})$. We define the event that $\mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t}))$ is rendered negative within one sampling period as $A_V := \left\{\sup_{t \in [t_k, t_{k+1})} \mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{t})) < -\epsilon\right\}$, and the final probability of A_V occurring is derived as $\mathbb{P}(A_V) \geq (1-\lambda)(1-\delta)$, given that the modeling error is sufficiently small, and the disturbance is bounded. This completes the proof of *Part* 1.

Part 2 : Subsequently, we prove the main results of the probabilities of Equations (37) and (38). We first prove Equation (37) which states that for any initial state inside $\Omega_{\rho_{nn}}$, the future state remains inside the terminal set $\Omega_{\rho_{\min}}$ in one sampling period. To simply the notations, the conditional expectations and the probabilities given that the event A_V occurs are denoted as $\mathbb{E}^*(\cdot)$ and $\mathbb{P}^*(\cdot)$, respectively. We consider the extreme scenario where the initial state is on the boundary of $\Omega_{\rho_{nn}}$, and show Equation (37) holds in this case. Specifically, using Dynkin's formula, we obtain the expected value of $V(x)$ as follows[15,16,22]:

$$\mathbb{E}^*(V(x(\tau_{T,\mathcal{Z}}(t)))) = V(x(t_k)) + \mathbb{E}^*\left(\int_{t_k}^{t_k+\tau_{T,\mathcal{Z}}(t)} \mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{s}))ds\right), \tag{43}$$

where $\mathcal{Z} = \Omega_{\rho_{\min}} \setminus \Omega^o_{\rho_{nn}}$, $t \in [t_k,\ t_{k+1})$, and $T = \infty$. As defined in Section "Notations", we have $\tau_{T,\mathcal{Z}}(t) = \min\{\tau_{\mathcal{Z}}, T, t\}$, where $\tau_{\mathcal{Z}}$ is the hitting time of the set $\mathcal{Z}$. Then, for any $x(t_k) \in \partial\Omega_{\rho_{nn}}$, we have the following inequality using the proof technique in Mahmood and Mhaskar[15] and Wu et al.[16]

$$\begin{aligned}\mathbb{E}^*(V(x(\tau_{T,\mathcal{Z}}(t)))) &= \int_{V \geq \lambda} V(x(\tau_{T,\mathcal{Z}}(t)))d\mathbb{P}^* + \int_{V < \lambda} V(x(\tau_{T,\mathcal{Z}}(t)))d\mathbb{P}^* \\ &\geq \widetilde{\lambda}\mathbb{P}^*\left(V(x(\tau_{T,\mathcal{Z}}(t))) \geq \widetilde{\lambda}\right)\end{aligned} \tag{44}$$

Let $\widetilde{\lambda} = \inf_{x \in \mathbf{R}^n \setminus \Omega_{\rho_{\min}}} V(x)$. We have the following inequality for $x(t_k) \in \partial\Omega_{\rho_{nn}}$:

$$\begin{aligned}\mathbb{P}^*(V(x(t))) \geq \rho_{min}, \text{for some } t \in [t_k,\ t_{k+1})) &\leq \frac{\mathbb{E}^*(V(x(\tau_{T,\mathcal{Z}}(t))))}{\widetilde{\lambda}} \\ &= \frac{V(x(t_k)) + \mathbb{E}^*\left(\int_{t_k}^{t_k+\tau_{T,\mathcal{Z}}(t)} \mathscr{L}\mathrm{V}(\mathrm{x}(\mathrm{s}))ds\right)}{\inf_{x \in \mathbf{R}^n \setminus \Omega_{\rho_{\min}}} V(x)} \\ &\leq \frac{V(x(t_k))}{\inf_{x \in \mathbf{R}^n \setminus \Omega_{\rho_{\min}}} V(x)}\end{aligned} \tag{45}$$

The last inequality is obtained using the fact derived in *Part* 1 that $\mathscr{L}\mathrm{V}$ is rendered negative with probability at least $(1-\lambda)(1-\delta)$. By taking the complementary events, we obtain the probability $\inf_{x(t_k) \in \partial\Omega_{\rho_{nn}}} \mathbb{P}^*(V(x(t)) < \rho_{\min},\quad \forall t \in [t_k,\ t_{k+1})) \geq (1-\beta)$, conditioned on the occurrence of event A_V, where β is defined in Equation (39a). Since we have $\mathbb{P}(A_V) \geq (1-\lambda)(1-\delta)$ from *Part* 1, the probability of Equation (37) is obtained using the properties of conditional probability.

Next, we consider the initial state $x(t_k) \in \Omega_\rho \setminus \Omega_{\rho_{nn}^o}$, and prove the probability of Equation (38). Specifically, we assume the initial state is on the boundary of Ω_{ρ_c}, $\rho_c \in [\rho_{nn}, \rho]$, which is a set between Ω_ρ and $\Omega_{\rho_{nn}}$, and show that the state will reach the boundary of $\Omega_{\rho_{nn}}$ before leaving Ω_ρ with a certain probability. Let $A_T := \left\{ \tau_{\mathbf{R}^n \setminus \Omega^o_{\rho_{nn}}} > \tau_{\Omega_\rho} \right\}$ denotes the complementary event that the state first hits the boundary of Ω_ρ instead of $\Omega_{\rho_{nn}}$. The following inequality is obtained since the event A_T belongs to the event $\left\{ \frac{V\left(x\left(\tau_{\Omega_\rho \setminus \Omega^o_{\rho_{nn}}}\right)\right)}{\rho} \geq 1 \right\}$.

$$\mathbb{P}^*\left(\tau_{\mathbf{R}^n \setminus \Omega^o_{\rho_{nn}}} > \tau_{\Omega_\rho}\right) \leq \mathbb{P}^*\left(\frac{V\left(x\left(\tau_{\Omega_\rho \setminus \Omega^o_{\rho_{nn}}}\right)\right)}{\rho} \geq 1\right) \leq \frac{V(x(t_k))}{\rho} \tag{46}$$

The last inequality is derived using the results in Equation (45). Therefore, given a positive real number γ satisfying Equation (39b), we have $\mathbb{P}^*\left(\tau_{\mathbf{R}^n \setminus \Omega^o_{\rho_{nn}}} < \tau_{\Omega_\rho}\right) \geq (1-\gamma)$ by taking the complementary event of A_T. The final probability of Equation (38) is derived accounting for the conditional probability of A_V.

Part 3: Finally, consider the stochastic nonlinear system of Equation (32) under the RNN-MPC of Equation (36). When $x(t_k) \in \Omega_\rho \setminus \Omega_{\rho_{nn}^o}$, the constraint of Equation (36e) is activated to optimize control actions such that $\mathscr{L}V$ is no greater than the one using the stabilizing controller $u = \Phi_s(x) \in U$, and thus, is also rendered negative. Therefore, using the results in *Part* 2 which prove that with a certain probability the state will reach the boundary of $\Omega_{\rho_{nn}}$ before leaving Ω_ρ, it follows that the probability under MPC is no worse than the probability of Equation (38). Once the state enters $\Omega_{\rho_{nn}}$, the constraint of Equation (36f) is activated to maintain the predicted states within $\Omega_{\rho_{nn}}$. In this case, Equation (37) gives the probability that the state of Equation (32) remains inside $\Omega_{\rho_{\min}}$ for the next sampling period. Therefore, for the stochastic nonlinear system of Equation (32) under the RNN-MPC of Equation 36, we derive the probability of closed-loop stability in the sense that the closed-loop state is bounded in Ω_ρ, and is ultimately bounded in $\Omega_{\rho_{nn}}$. This completes the proof of Theorem 3.

Remark 5. Equations (37) and (38) in Theorem 3 give the probabilities of closed-loop stability for each sampling period. While the RNN predictions are invoked recursively within MPC to predict for the entire prediction horizon, the probabilities of closed-loop stability remain unaffected since only the first control action is applied for the next sampling period. The RNN-MPC is implemented in a receding horizon manner by recursively solving the optimization problem of Equation (36) with new state measurements received at each sampling time. Therefore, at each time step, Equations (37) and (38) can be used to estimate the probability of system being stable under RNN-MPC.

Remark 6. It should be noted that the probabilities of Equations (37) and (38) represent only the lower bounds for closed-loop stability under RNN-MPC. The actual probability of closed-loop stability could be higher due to a number of reasons: (1) the RNN model is well trained and the modeling error does not reach the upper bound for every time step, (2) in general, the stochastic disturbances in industrial chemical plants fall within a bounded region in most of the time, which can be handled through the robustness of MPC, and (3) the optimality of MPC improves closed-loop performance in terms of fast convergence to the steady-state under the constraint of Equation (36f), which leads to better probability results than those derived under the controller $u = \Phi_s(x)$ in Theorem 3.

Remark 7. Theorem 3 demonstrates that in addition to the RNN structure in terms of width and depth, and the training sample size that affect the neural network generalization performance (Theorem 1), the closed-loop stability for the stochastic system of Equation (32) under RNN-MPC also depends on the sampling time and the size of multiple sets embedded in the stability region Ω_ρ. Therefore, all the factors above should be accounted for to improve the overall probability of stability for the nonlinear systems subject to stochastic disturbances.

Remark 8. In the presence of stochastic disturbances with unbounded variation, one of the benefits of using the RNN-MPC of Equation (36) is that the operating region can be characterized less conservatively by utilizing the probability distribution of disturbances. It was demonstrated in Wu et al.[16] that compared to the RNN-MPC of Equation (36), the closed-loop operating regions were overly conservative using the robust controller design that handles disturbances in a bounded manner, which leads to reduced economic benefits in the context of economic MPC. Therefore, the probabilities in Theorem 3 demonstrate the relationship between closed-loop stability and operating region size, which could provide a guidance to characterize the operating region when implementing machine learning models in real chemical processes subject to various process disturbances.

4 | APPLICATION TO A CHEMICAL PROCESS EXAMPLE

To demonstrate the efficacy of machine-learning–based MPC and study the impact of data sample size on RNN generalization

performance and system stability in the presence of bounded disturbances and stochastic disturbances, we present a simulation example using the chemical process example from Wu et al.[20] Specifically, we consider a continuous stirred tank reactor (CSTR) that is non-isothermal and well-mixed with reactant A transformed into product B ($A \rightarrow B$) in an exothermic, irreversible second-order reaction. A heating jacket is equipped to remove/supply heat at a rate Q. We first consider the case of bounded disturbances, and present the process dynamical model by the following energy and material balance equations:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 + w_1 \tag{47a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} + w_2 \tag{47b}$$

where T denotes the temperature in the reactor, and C_A represents the concentration of reactant A. F is the volumetric flow rate, T_0 is the feed temperature, and C_{A0} is the feed concentration of reactant A. V and Q are the volume of the reacting substance in the reactor and the heat input rate, respectively. $w^T = [w_1\ w_2]$ are bounded disturbances of Gaussian distribution with variance $\sigma_1 = 2.5\ \text{kmol/m}^3$, $\sigma_2 = 70\,K$, and bounds $|w_1| \le 2.5\ \text{kmol/m}^3$, $|w_2| \le 70\,\text{K}$. The definition of all the other parameters and their values are reported in Wu et al.[20] A schematic of the CSTR with an irreversible, second-order reaction can be found in Wu and Christofides.[23]

In the presence of stochastic disturbances, the nonlinear system can be represented in the following form:

$$dC_A = \frac{F}{V}(C_{A0} - C_A)dt - k_0 e^{\frac{-E}{RT}} C_A^2 dt + \bar{\sigma}_1 (C_A - C_{As}) d\bar{w}_1 \tag{48a}$$

$$dT = \frac{F}{V}(T_0 - T)dt + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 dt + \frac{Q}{\rho_L C_p V} dt + \bar{\sigma}_2 (T - T_s) d\bar{w}_2, \tag{48b}$$

where $\bar{w}_1$, $\bar{w}_2$ are standard Wiener processes that satisfy $\bar{w}(0) = 0$ and $\bar{w}(t) - \bar{w}(s) \sim \sqrt{t-s}\,\mathcal{N}(0,1)$ ($\mathcal{N}(0,1)$ is a normal distribution with zero mean and unit variance). To simulate the Wiener process, we discretize the Wiener process with the integration time step h_c as follows: $d\bar{w}_i \sim \sqrt{h_c}\,\mathcal{N}(0,\ 1)$, $i = 1,\ 2$, and thus, the realization of the Wiener process can be obtained through $\bar{w}_i(t + h_c) = \bar{w}_i(t) + d\bar{w}_i$, $\forall t \ge 0$. The coefficients $C_A - C_{As}$ and $T - T_s$ are added to ensure that the disturbances vanish at the steady-state. The variances $\bar{\sigma}_1 = 2.5\ \text{kmol/m}^3$, $\bar{\sigma}_2 = 70\,\text{K}$ are used in the simulation of Equation (48). The RNN-MPC is designed to stabilize the reactor at $(T_s,\ C_{As}) = (402\,\text{K},\ 1.95\ \text{kmol/m}^3)$, which is an unstable steady-state under the given input values $(Q_s,\ C_{A0s}) = (0\,\text{kJ/h},\ 4\ \text{kmol/m}^3)$. The heat supply/removal rate and the inlet concentration of species A are the two manipulated inputs. All the states and inputs of the process are represented in their deviation variable forms, i.e., $\Delta T = T - T_s$, $\Delta C_A = C_A - C_{As}$, $\Delta C_{A0} = C_{A0} - C_{A0_s}$, and $\Delta Q = Q - Q_s$. Additionally, the upper bounds of the manipulated inputs are $|\Delta Q| \le 5 \times 10^5\ \text{kJ/h}$ and $|\Delta C_{A0}| \le 3.5\ \text{kmol/m}^3$.

We conduct extensive simulation by discretizing the targeted region (i.e., the stability region $\Omega_{\hat{\rho}}$ in this work) with sufficiently small intervals in state-space and choosing every possible initial conditions within this discretized region. Similarly, the input values are also chosen from $u \in U$ with a sufficiently small interval. In this way, sufficient variety is ensured in our dataset, and the RNN model developed using this dataset is able to capture the process dynamics well for the entire operating region. The RNN training process follows the standard training method in Wu et al.[20] In this study, the RNN models are trained using different data sample size (while other parameters and settings remain unchanged), and the generalization performance is evaluated using the testing data. The RNN models are built with 50 neurons in a single hidden layer. The MSE is used as the loss function. PyIpopt, which is a python connector to the IPOPT software package, is used to solve the MPC optimization problem,[24] and Keras is used to build and train RNN models.[25] Figure 2 shows the relationship between the RNN generalization performances and the training sample size, from which it is demonstrated that with less data used for training, the errors for testing

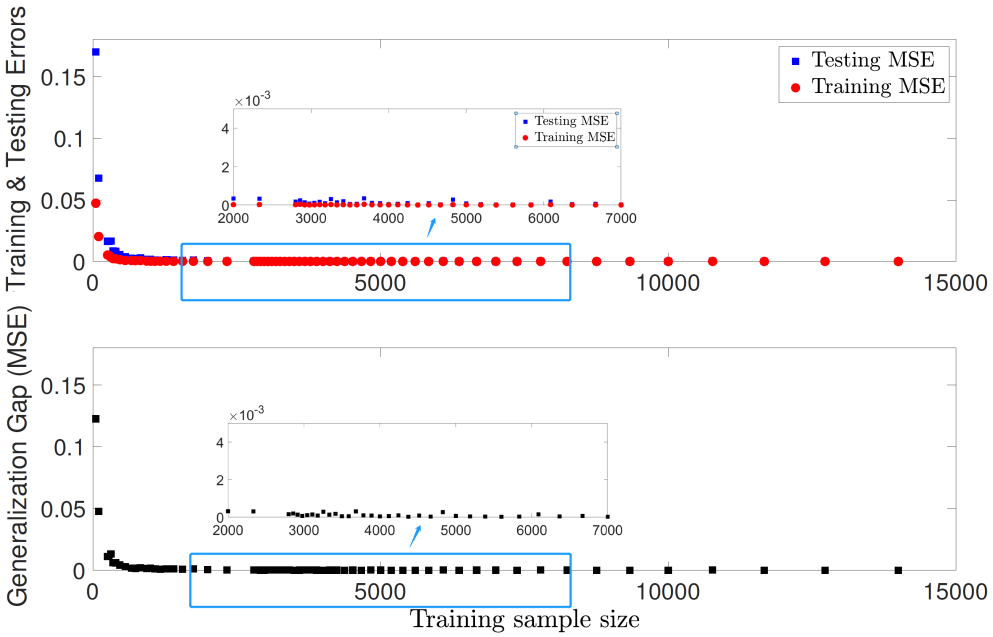

FIGURE 2 Generalization performance for the RNN models utilizing various sample sizes. MSE, mean squared error; RNN, recurrent neural networks

and training both increase.[7] Additionally, we calculate the generalization gap using $\mathbb{E}[g_t(\mathbf{x}, \mathbf{y})] - \frac{1}{m}\sum_{i=1}^{m} g_t(\mathbf{x}_i, \mathbf{y}_i)$. The increase of the generalization gap in Figure 2 implies a worse generalization performance for models with less training data. It is demonstrated in Figure 2 that a desired generalization performance has been achieved for sample sizes greater than 2000, and overfitting does not occur with the increase of training samples.

Subsequently, we simulate the closed-loop system with bounded disturbances and Wiener process disturbances with unbounded variation. Figure 3 shows the probability results obtained through the simulation of various initial conditions (48 initial conditions) within $\Omega_{\hat{\rho}}$ using the six RNN models trained earlier with different sample sizes. The probabilities in Figure 3 are calculated using the following rule: given an initial condition, the closed-loop system under RNN-MPC is unstable if the state trajectory leaves $\Omega_{\hat{\rho}}$ at any time step or escapes from $\Omega_{\rho_{\min}}$ after it enters $\Omega_{\rho_{\min}}$ due to disturbances and/or modeling error. It is shown in Figure 3 that the probability of closed-loop stability increases with more data used for training, which is consistent with the results shown in the open-loop simulation study (Figure 2). Additionally, it is observed that the probabilities of closed-loop stability under bounded disturbances reach 0.7 for training sample size greater than 1×10^4, and the probabilities under unbounded, stochastic disturbances reach 0.6 with the same number of training sample size. Overall, it is shown that the MPC under bounded disturbances achieves higher probabilistic of closed-loop stability than that under unbounded, stochastic disturbances with the same variances. This is consistent with the disturbance realizations shown in Figure 4, from which it is demonstrated that the Gaussian disturbance on temperature T is bounded within the $\pm\sigma_2$ region (top figure), while the Wiener process disturbance is unbounded, and has a greater impact on system stability due to its wider range. Additionally, it should be noted that the probability results in Theorem 2 (for bounded disturbances) and Theorem 3 (for stochastic disturbances) only provide a lower bound for the probability of closed-loop stability. Therefore, we cannot directly obtain the value of the parameter δ from simulation results.

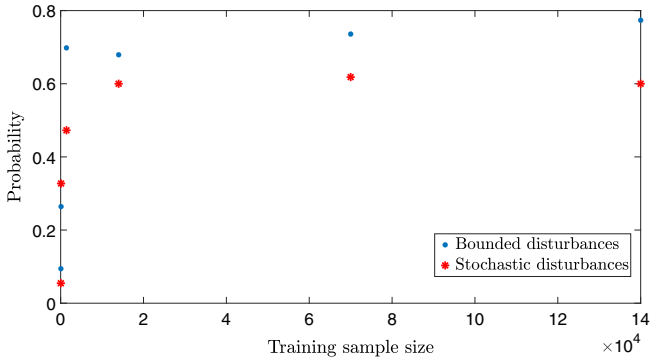

FIGURE 3 Probability of closed-loop stability under bounded disturbances (blue circles) and stochastic, unbounded disturbances (red asterisks), respectively, using the RNN-MPC trained with various sample sizes. MPC, model predictive controllers; RNN, recurrent neural networks

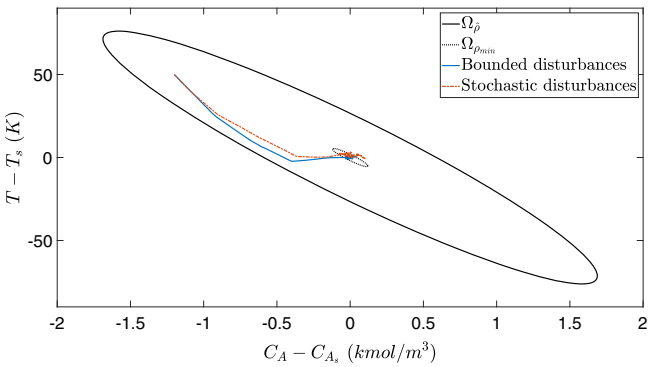

FIGURE 5 Closed-loop state trajectories under MPC with bounded disturbances (blue, solid line) and stochastic, unbounded disturbances (red, dashed line) for the same initial condition $(-1.2,\ 50)$. MPC, model predictive controllers

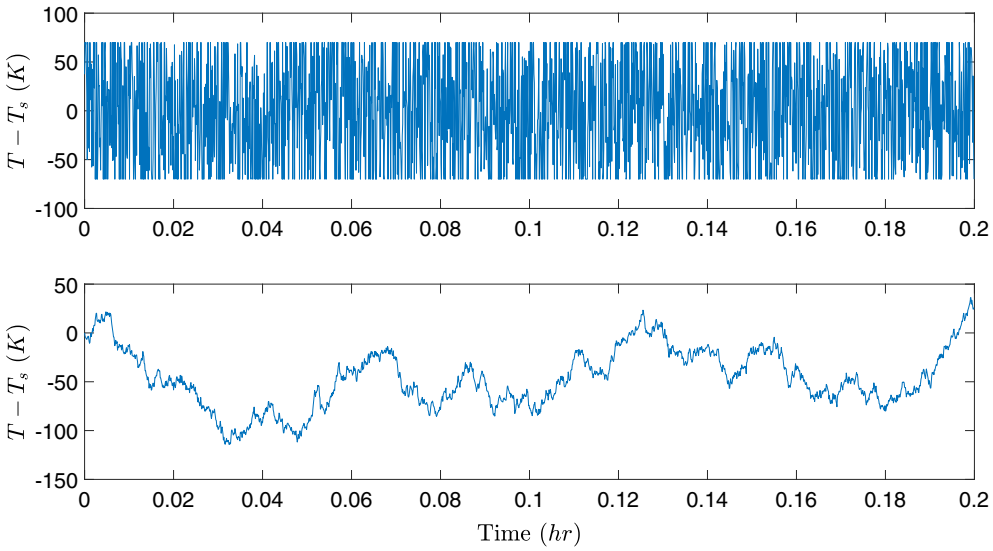

FIGURE 4 Bounded, Gaussian disturbance (top figure), and unbounded, Wiener process disturbance (bottom figure) on temperature T

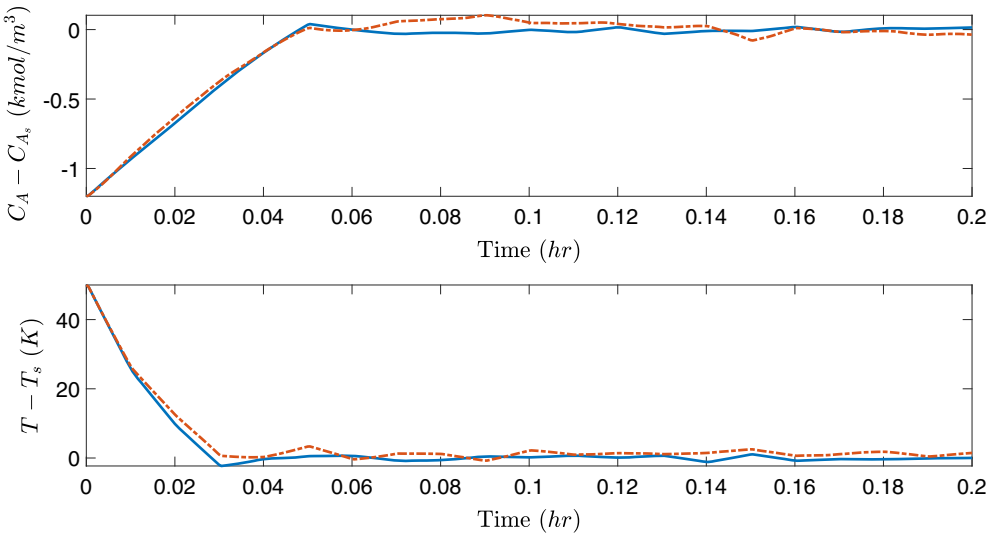

FIGURE 6 Closed-loop state profiles for the CSTR with bounded disturbances (blue, solid line) and stochastic, unbounded disturbances (red, dashed line) under RNN-MPC for the same initial condition $(-1.2,\ 50)$. CSTR, continuous stirred tank reactor; MPC, model predictive controllers; RNN, recurrent neural networks

The state-space trajectory and the state profiles for the initial state $x_0 = (-1.2,\ 50)$ with the model trained with 7000 training data are shown in Figures 5 and 6. In Figure 5, it is seen that the dynamic trajectory remains inside the stability region $\Omega_{\widehat{\rho}}$ all the time, and ultimately converges to the small ball $\Omega_{\rho_{\min}}$ in the presence of bounded disturbances (blue, solid line); however, in the presence of stochastic disturbances, the state trajectory leaves $\Omega_{\rho_{\min}}$ during its oscillation around the steady-state, and thus is considered unstable in this case. In Figure 6, it is shown that for this particular initial condition, the closed-loop states (i.e., reactor temperature T and reactant concentration C_A) are stabilized at the origin after around 0.06 h for both disturbances, with slight variation around the steady-state afterwards due to disturbances. The case study demonstrates the relation between RNN training sample size and its generalization performance as well as the probability of closed-loop system stability, which supports the results derived in Theorems 1 and 2.

Remark 9. Note that the RNN generalization performance depends on various factors as demonstrated in Theorem 1. While we only showed the impact of the sample size of training data on RNN generalization performance in this section due to space limitations, the generalization error is also affected by RNN width/depth, and input time length. Interested readers are referred to[21] for the simulation studies of open-loop RNN generalization performance and system stability analysis that account for all the above factors for the nominal system without any disturbances.

Remark 10. The probability results in Figures 2 and 3 are consistent with the theoretical results derived in Theorems 1–3 in a qualitative manner, i.e., the RNN generalization performance and probability of closed-loop stability improve with increasing training samples. However, it is difficult to carry out a quantitative analysis since the analytically calculated minimum sample size is based on the upper bound of RNN generalization error, while in practice the RNN prediction error will not reach its upper bound for every time step. Therefore, the analytically derived probability results could be conservative compared to the simulation results in Figures 2 and 3. Additionally, the probability of closed-loop stability also depends on the Lipschitz constants and parameters such as $\widehat{c}_i$, $i = 1, 2, 3, 4$ that relate to process dynamics and are generally difficult to compute. Therefore, in this work, we only provide a qualitative analysis for the relationship between the probabilistic closed-loop stability and the training sample size.

Remark 11. Computational efficiency could be a challenge for the implementation of RNN models to large-scale chemical processes. To reduce the computational costs for developing RNN models and solving RNN-MPC, reduced-order modeling techniques such as feature selection and autoencoder can be utilized to build reduced-order RNN models for large-scale chemical processes. Additionally, a priori process knowledge can also be used to reduce the complexity of RNN models and improve its computational efficiency and accuracy.

5 | CONCLUSION

We developed machine-learning–based predictive control schemes for nonlinear systems subject to stochastic disturbances with unbounded variation and bounded disturbances, respectively. We first derived a generalization error bound for the RNN models developed for the nominal system using the Rademacher complexity method from statistical learning theory. Then, we established system stability

results for the uncertain system with unknown disturbances in a bounded manner. With regards to the uncertain system with stochastic disturbances under RNN-MPC, we accounted for the distribution information of disturbances, and derived the probabilistic closed-loop stability properties. Through the simulation of a chemical reactor example, we demonstrated that the training data sample size affects the RNN generalization performance, and closed-loop stability for the MPC using RNN models.

ACKNOWLEDGMENTS

Financial support from the NUS Start-up grant, the U.S. National Science Foundation and the Department of Energy is gratefully acknowledged.

AUTHOR CONTRIBUTIONS

Zhe Wu: Conceptualization (equal); formal analysis (equal); methodology (equal); writing – original draft (equal). **Aisha Alnajdi:** Formal analysis (equal); methodology (equal); software (equal). **Quanquan Gu:** Supervision (supporting); writing – review and editing (equal). **Panagiotis D. Christofides:** Supervision (equal); writing – review and editing (equal).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Panagiotis D. Christofides https://orcid.org/0000-0002-8772-4348

REFERENCES

1. Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *ArXiv*. [Preprint] Available from: arXiv:1905.13210, 2019.
2. N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. *Proceedings of the Conference on Learning Theory*, 297–299. PMLR 2018.
3. J. Hanson, M. Raginsky, and E. Sontag. Learning recurrent neural net models of nonlinear systems. *ArXiv* [Preprint] Available from: arXiv: 2011.09573, 2020.
4. D. Zou and Q. Gu. An improved analysis of training over-parameterized deep neural networks. *ArXiv* [Preprint] Available from: arXiv:1906.04688, 2019.
5. P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *ArXiv*. [Preprint] Available from: arXiv:1706.08498, 2017.
6. M. Chen, X. Li, and T. Zhao. On generalization bounds of a family of recurrent neural networks. *ArXiv*. [Preprint] Available from: arXiv: 1910.12947, 2019.
7. Wu Z, Rincon D, Gu Q, Christofides PD. Statistical machine learning in model predictive control of nonlinear processes. *Mathematics*. 2021;9:1912.
8. Kimaev G, Ricardez-Sandoval LA. Nonlinear model predictive control of a multiscale thin film deposition process using artificial neural networks. *Chem Eng Sci*. 2019;207:1230-1245.
9. Limon D, Calliess J, Maciejowski JM. Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*. 2017;50:7769-7776.
10. M. Mittal, M. Gallieri, A. Quaglino, S.S.M. Salehian, and J. Koutník. Neural Lyapunov model predictive control. *ArXiv* [Preprint] Available from: arXiv:2002.10451, 2020.
11. Wu Z, Tran A, Rincon D, Christofides PD. Machine learning-based predictive control of nonlinear processes. Part I: theory. *AIChE J*. 2019;65:e16729.
12. Deng H, Krstić M. Output-feedback stabilization of stochastic nonlinear systems driven by noise of unknown covariance. *Systems Contr Lett*. 2000;39:173-182.
13. Deng H, Krstic M, Williams RJ. Stabilization of stochastic nonlinear systems driven by noise of unknown covariance. *IEEE Transact Automatic Control*. 2001;46:1237-1253.
14. Homer T, Mhaskar P. Output-feedback lyapunov-based predictive control of stochastic nonlinear systems. *IEEE Transact Automatic Control*. 2017;63:571-577.
15. Mahmood M, Mhaskar P. Lyapunov-based model predictive control of stochastic nonlinear systems. *Automatica*. 2012;48:2271-2276.
16. Wu Z, Zhang J, Zhang Z, et al. Economic model predictive control of stochastic nonlinear systems. *AIChE J*. 2018;64:3312-3322.
17. Kimaev G, Ricardez-Sandoval LA. Artificial neural networks for dynamic optimization of stochastic multiscale systems subject to uncertainty. *Chem Eng Res Design*. 2020;161:11-25.
18. Mohri M, Rostamizadeh A, Talwalkar A. *Foundations of Machine Learning*. MIT Press; 2018.
19. A. Maurer. A vector-contraction inequality for rademacher complexities. *Proceedings of the International Conference on Algorithmic Learning Theory*, 3–17. Springer, 2016.
20. Wu Z, Tran A, Rincon D, Christofides PD. Machine learning-based predictive control of nonlinear processes. Part II: computational implementation. *AIChE J*. 2019;65:e16734.
21. Khasminskii R. *Stochastic Stability of Differential Equations*. Vol 66. Springer Science & Business Media; 2011.
22. Øksendal B. *Stochastic Differential Equations*. Springer; 2003:65-84.
23. Wu Z, Christofides PD. *Process Operational Safety and Cybersecurity: A Feedback Control Approach*. Springer Nature; 2021.
24. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Programm*. 2006;106:25-57.
25. F. Chollet et al. Keras. https://keras.io 2015. Available at: https://github.com/fchollet/keras.

How to cite this article: Wu Z, Alnajdi A, Gu Q, Christofides PD. Statistical machine-learning–based predictive control of uncertain nonlinear processes. *AIChE J*. 2022;68(5): e17642. doi:10.1002/aic.17642