Evidential Reasoning with Expert-Guided Machine Learning

Xueying Ding
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213

Gopaljee Atulya
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213

Alex Davis
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213
ald1@andrew.cmu.edu

Aarti Singh
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213
aartisingh@cmu.edu

Shane Fazzio
Lockheed Martin Corporation
Denver, Colorado, 80127
ronald.s.fazzio@lmco.com

Abstract

Evidential reasoning aims to infer hidden causes from observed effects. In the context of fault detection, it is possible to trace the cause of anomalies by combining evidential reasoning with physical knowledge. However, experts may not have full knowledge of the physical systems, and data may not be large enough to learn from scratch. We develop an online evidential reasoning algorithm that blends machine learning with expert-provided physical knowledge about causal structure and functional form. The expert first represents possible causes and effects in the structure of a Bayesian Network, then provides physics-informed priors about the model relating failure modes to effects, allowing inference in the absence of strong supervision. As data are sampled from the physical system, predictions are generated using a quasi-Bayesian mixture of the expert's judgment and a data-driven estimate. With simulated datasets, we evaluate the conditions under which the system converges to correct causal inferences under weak supervision, and small amounts of strong supervision from the expert. We find that the approach is able to make accurate inferences with little or no data unless the expert's physical model is very incorrect or the signal to noise ratio across error modes is small.

1 Introduction

Evidential reasoning aims to make inferences from observed effects to hidden causes [14], and is important in many applications, from medical diagnosis, to analyzing the root cause of a manufacturing machine's failure (a.k.a root cause analysis or RCA). In the manufacturing case, patterns in measurable quantities (e.g., sound, vibration, force) captured during operation will vary over time due to changes of operation, component degradation, corrosion, overheating, and other effects. A Bayesian Network (Bayes Net) can connect the observed sensor measurements (effects) to their hidden causes. A maintenance technician or engineer forms expectations about the patterns of observed variables given different failure modes, such as degradation after a specified number of cycles. The expert's task is to study the patterns (sensor readings) to determine whether the system is healthy, and if not, remedy the cause of the problem before failure occurs. Predictive maintenance

Preprint. Under review.

approaches that combine expert knowledge and machine learning to solve the evidential reasoning problem have the potential to reduce costs, improve situational awareness, and optimize machine operation [15, 2, 18]. Schedule based maintenance, on the other hand, is suboptimized as it risks both performing maintenance tasks when not needed, resulting in excess cost, and missing failure in progress, resulting in lost of productivity due to unplanned downtime.

Because machine failure is costly, algorithms would ideally be able to detect faults before they occur, with little supervision. Physical knowledge about expected patterns may be provided by a knowledgeable expert or collection of experts. In some cases, this knowledge may be represented as a system of ordinary differential equations (ODEs) or partial differential equations (PDEs). Recent research has used neural networks to leverage this kind of prior physical knowledge [23, 4, 16]. An effective algorithms should also be able to adapt to streaming data, reduce the burden of system operators, and be robust to operator mis-specification of the relationship between signal patterns and hidden causes. Leveraging expert knowledge about the potential causes and physical functional relationships underlying machine operation, we propose an approach that: 1) incorporates expert-provided physical knowledge into the modeling of machine failure modes, 2) uses an expert-informed Bayes Net to simplify the evidential reasoning inference problem, and 3) combines an expert's judgment with a data-driven model to solve the root cause analysis problem.

2 Related Work

Machine learning approaches to root-cause analysis may use supervised learning combined with physics-based feature engineering, such as using a signal's frequency spectrum in a vibration model. Many features in both the time and frequency domains have proved useful, including the energy ratio, side-band index, sideband level factor, root mean squared error, energy operator, skewness, kurtosis, crest, energy ratios of wavelet coefficients, ensemble empirical mode decomposition, wavelet kernel local fisher discriminant analysis, among others [9, 24]. Feng *et al.* [9] use a local connection network with normalized sparse auto-encoders, and Hong *et al.* [8] use Gaussian Process regression to model bearing degradation, where the rMSE of the GP fit on normal data and tested on new data is used to classify failure one day in advance.

Another important tool is the use of Bayes Nets. In more general settings, Kaufman *et al.* [10] propose a method for estimating the Markov blanket for some variables in large networks, which improves subsequent Bayesian inference using Gibbs sampling. Louizos et al. [17] use variational auto-encoders to represent confounders as latent variables to estimate causal effects.

Inclusion of experts in machine learning has gained interest in recent years. Abdollahi *et al.* [1] use expert guidance to optimize the selection of parameter spaces and hyper-parameters in the challenging task of 3D printing soft materials. Gennatas *et al.* [7] use expert decision rules as priors in a machine learning model. Xu *et al.* [26] use mis-specified expert functions in the form of pairwise comparisons for optimization.

3 Using expert-provided Bayes Nets for root cause analysis

We next describe an approach to use expert-provided Bayes Nets in the evidential reasoning task. Assume a physical system has S sensors $s_1, s_2, \dots s_S$. For any time interval $[0, t]$, we are interested in the quantity $p(\mathbf{m}|s_{1,0}, s_{2,0}, ...s_{1,t}, s_{2,t}, \dots s_{S,t})$ where $\mathbf{m}$ is a set of causes (normal or failure modes) with categorical input space $\mathcal{M}$, and $s_{i,t}$ denotes the measurements of sensor i at time t. If both the sensors and failure modes are observed, one can train physics-based neural network models $\{{\psi_i}^{\mathbf{m}}\}$ (see next section for ψ) to fit the sensor i's readings to a pattern, according to specific failure modes.

Bayes Nets capture the fact that it is unlikely that all failure modes affect all sensors. Causal relationships between the hidden states $\mathbf{m}$ and the observed variables s are initially given by experts as a Bayes Net $\mathcal{G}$. Using the Bayes Net we can factorize the joint distribution $p(\mathbf{m}, s_{1,0}, s_{2,0}, ...s_{1,t}, s_{2,t}, \dots s_{S,t})$ into a product of conditional distributions based on the parents (Pa) of each sensor: $p(\mathbf{m}, s_{1,0}, s_{2,0}, ...s_{1,t}, s_{2,t}, \dots s_{S,t}) = \prod_{i=1}^{S} p(s_{i,0}, s_{i,1}, ...s_{i,t}, |Pa(s_i) \subseteq \mathbf{m})) \cdot p(\mathbf{m})$.

With this factorization it is possible to do likelihood-based inference if robust likelihoods are available for the conditional distributions of $p(s_{i,0}, s_{i,1}, ...s_{i,t}, |Pa(s_i) \subseteq \mathbf{m}))$. In our model we impose the assumption that the distribution of sensors with respect to the fault vector $\mathbf{m}$ satisfies a Gaussian distribution, such that $p(s_{i,0}, s_{i,1}, ...s_{it}|\mathbf{m}) \sim \prod_{j=0}^{t} \mathcal{N}(\psi_i{}^{\mathbf{m}}(j), \sigma_i^2)$. A sequence is predicted based on the neural network $\psi_i{}^{\mathbf{m}}$ with input $[0, t]$ and compared with each $s_{i,j}, j \in [0, t]$ to see how similar two functions behave in terms of mean squared error. Our goal is to: 1) Estimate $\tilde{\mathbf{m}}$ by the conditional probabilities, and 2) update $\psi_i{}^{\mathbf{m}}$ based on streaming data.

The approach works as follows:

1. Construct Bayes Net $\mathcal{G}$ from the expert's engineering analysis of the system
2. For S sensors $s_{1,0}, s_{2,0}, ...s_{1,t}, s_{2,t}, \dots s_{S,t}$ and hidden state $\mathbf{m}$, factorize their joint distribution according to $\mathcal{G}$

$$p(\mathbf{m}, s_{1,0}, s_{2,0}...s_{1,t}, s_{2,t}, \dots s_{S,t}) \propto \prod_{i=1}^{S} p(s_{i,0}, s_{i,1}, ...s_{i,t}|Pa(s_i) \subseteq \mathbf{m})p(\mathbf{m}) \quad (1)$$

   where $Pa(s_i) \subseteq \mathbf{m}$ are the parents of sensor i in $\mathcal{G}$
3. Initialize $\psi_i{}^{\mathbf{m}}$ with an expert's assessment of the time series model for each sensor i and each combination of its parents
4. For each of the sensors and failure mode combinations, label the sensor data $\mathbf{s}_t$ observation with:

$$\tilde{\mathbf{m}} = \arg\max_{\mathbf{m} \in \mathcal{M}} \Big[\prod_{i=1}^{S} p(s_{i,0}, s_{i,1}, ...s_{i,t}|Pa(s_i) \subseteq \mathbf{m})p(\mathbf{m}) \Big] \quad (2)$$

$$= \arg\max_{\mathbf{m} \in \mathcal{M}} \Big[\prod_{i=1}^{S} \prod_{j=0}^{t} \mathcal{N}\big(s_{i,j}; \psi_i{}^{\mathbf{m}}(j), \sigma_i^2\big)p(\mathbf{m}) \Big] \quad (3)$$

5. Label the data vector $\mathbf{s}_t$ with the label $\tilde{\mathbf{m}}$, which indicates the failure modes that cause the machine to produce this data. (Optional strong expert supervision: Ask the expert to verify the accuracy of the label by inspecting the machine.)
6. With $n^{\tilde{\mathbf{m}}}$ as the number of labels for class $\tilde{\mathbf{m}}$, use learning rate $0 \leq \alpha(n^{\tilde{\mathbf{m}}}) \leq \alpha(n^{\tilde{\mathbf{m}}}+1) \leq 1$ to update $\psi_i{}^{\tilde{\mathbf{m}}}$ using backpropagation:

$$\psi_{i,\text{updated}}^{\tilde{\mathbf{m}}} = \alpha(n^{\tilde{\mathbf{m}}})\psi_i^{\tilde{\mathbf{m}}} + (1 - \alpha(n^{\tilde{\mathbf{m}}}))\psi_{i,\text{expert}}^{\tilde{\mathbf{m}}} \quad (4)$$

4 Integrating expert-provided physics into learning time-series functions

Machinery is governed by the classical mechanics and electrodynamics and consequently, the machine's sensor readings and signals will be governed by a systems of ODEs or PDEs that represent the underlying physics. The aim of predictive analytics is thusto solve for the parameters of the equations that govern the patterns of the data, under various anomaly scenarios. Our approach aids the experts by training the neural network ψ on a series of time-dependent readings for each sensor and failure mode.

We leverage recent advances in physics-based modeling to improve machinery models using neural networks that require little or no data for accurate estimation. Specifically, we use *physics-constrained loss functions* to approximate the solution to an ODE/PDE using a neural network [21, 22, 20]. The approach learns a hidden function $u(t, x)$ such that [20]:

$$\frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda] = 0 \quad (5)$$

where $\mathcal{N}[u; \lambda]$ is an arbitrary non-linear differential operator parameterized by λ and $x \in \mathbb{R}^D$. For example, the 1D Berger's equation is $\mathcal{N}[u; \lambda] = \lambda_1 u \frac{\partial u}{\partial x} - \lambda_2 \frac{\partial^2 u}{\partial^2 x}$. To enforce the physics-based constraint, the approach defines $f(t, x) = \frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda]$ and adds $\sum_{t=1}^{T} f(t, x)^2$ as a regularizer, where $\frac{\partial u}{\partial t}$ and $\mathcal{N}[u; \lambda]$ are calculated using auto-differentiation on the fitted neural network.

In our approach, we treat this differential equation information as imperfect prior knowledge provided by the expert. Model predictions are then based partly on a data-driven model, and partly on the expert's guess about the system's physical dynamics ψ_{expert}. To capture the data-driven component, we use a neural network with two hidden layers and sine activation functions [25] to predict physical quantities of interest at each timestep. Prior research [22] finds that such an architecture can learn to reproduce not only the positions in a fluid flow problem, but also the governing PDE equations. The use of sinusoidal activation functions in physics-informed neural networks was rigorously studied in the SIREN architecture to deal with the infinite local minima problem [25], with the ability to solve boundary value problems including Poisson, Helmholtz, and wave equations.

5 Summary of the role of the expert

The expert plays a key role in several stages of the proposed approach. Here we provide an overview of the expert's role in each part of the algorithm.

Bayes Net. The expert provides a Bayesian network $\mathcal{G}$ for the algorithm. For failure mode analysis, this often takes the form of a bipartite graph, where the first part of the graph consists of the failure modes F, and the second part includes the observable sensors S. The directed edges E link failure modes to some subset of the sensors. Experts should be able to construct such a graph based on their engineering knowledge of the system, particularly if they are experienced with the machine.

Assessment of the causal model. The expert provides an assessment of how the sensor observations behave given different configurations of failure modes. For example, in the case of a turning or cutting machine, there is usually a fundamental frequency at which the system vibrates, and chattering or other problems with the system add new frequencies. This additivity of signals can be used to simplify the combinatorial problem of specifying the conditional distribution of each sensor for all combinations of its parents. For the algorithm to be useful, the expert needs to be able to give precise enough estimates such that the difference between the signals across normal and failure modes is larger than the noise variance in the system.

Estimation of failure modes distribution. Although not used in our current approach, the expert can also provide base rate (unconditional) guesses about the probability of different failure modes. As the system is continuously fitting data-driven models and weakly classifying hidden states as failure modes at each timestep, it is possible for these estimates to also be data driven.

Strong supervision. We assume the expert can inspect the machine to determine whether the algorithm correctly classifies the failure modes. This might be visual inspection, diagnostic testing, or other relevant engineering analyses. Strong supervision is particularly important when the expert is inaccurate in the initial estimates (or is unable to provide estimates). If a previously unobserved malfunctioning cause is detected, the expert can add additional nodes (F, S) and edges (E) to refine the Bayesian network $\mathcal{G}$. The ideal scenario is that the expert can give strong supervision with the system; that is, the expert can detect all misclassified labels and provide correct feedback.

6 Synthetic experiments

We examine the ability of the proposed approach to recover time series functions and accurately classify error modes with little or no data, focusing on a) the amount of data required, b) noise in the data generating process, c) bias in the expert's judgments, and d) strong supervision. In a real machine deployed in a manufacturing environment, the Bayesian Network and functional relationships can be quite complex. For the purpose of initial demonstration, we use a simple Bayesian network with two sensors $S = \{s_1, s_2\}$ and three failure modes $F = \{f_1, f_2, f_3\}$. The directed edges depict the failure modes that are parents of each sensor, as shown by Figure 1. The probability of each failure mode occurring is characterized by independent Bernoulli random variables with probability p of being active during any time window $T = [T_1, T_2]$.

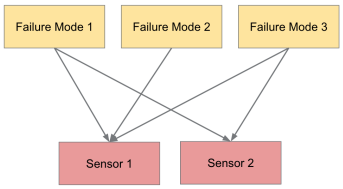


Figure 1: Bayesian network represented as bipartite graph with sensors $S = \{s_1, s_2\}$ and failure modes $F = \{f_1, f_2, f_3\}$.

Each sensor outputs either the normal operation signal (no failure modes are active) or a sum of the signals produced by the active failure modes. This means that each element $m \in \mathbf{m}$ is also $m \in F$. Let s_{iT} denote the signal observed by sensor i during the time interval T, then we have

$$s_{iT} = \begin{cases} \psi_i^0(T), & \text{if no fault} \\ \sum_{m \in F} \mathbb{1}(m = 1)\psi_i^m(T), & \text{if fault occurs} \end{cases} \tag{6}$$

where ψ_i^m is the function that produces signals when fault m occurs, ψ_i^0 produces the normal functioning signals.

Before data are collected from the machine, experts provide an assessment of the functional causal model for each sensor for all combinations of its parent failure modes, with a maximum of $2^{|F|}$ assessments. Ideally, given an arbitrary time window t, the estimate for sensor i follows $(Pr(|\hat{s}_{iT} - s_{iT}|) > \epsilon) < \delta)$. As we have more data collected and with correct expert's intervention, we want our estimated $\hat{s_i}$ to get closer to the true s_i.

6.1 Experiment 1: Time windows

First we consider the case where the expert provides nearly accurate estimates of the signal for each failure mode in the presence of small amounts of noise. We examine the amount of data required for the algorithm to accurately learn the sum of periodic functions. Our key variable manipulated in the experiment is the time window size, proportional to the number of observations collected after the failure occurs. Ideally this number will be small so that faults can be identified early and machine down-time will be small.

Data Generation. Our assumption is that the expert already knows how s_1 and s_2 behave during normal operation ($\sin(t)$ and $\cos(t)$, respectively). When a failure occurs, the two sensors s_1 and s_2 are generated with the same periodic functions, according to Table 1. This simplification does not affect the generality of the approach, because different neural networks are still required to learn the periodic functions for each combination of failure modes. We assume all combinations of errors occur with equal probability. Sensor measurements are collected for 1 observation per 0.2 seconds, with T samples are collected for a time window of $T \times 0.2$ seconds. The experiment varies the length of time T for collection of sensor data. Our primary outcomes are the accuracy of failure mode classification and resulting average MSE's between the modeled functions and data.

Table 1: Parameters for Data Generation in Experiment 1

Fault Mode	Expert's Assumption	Actual Equation
1	$2\cos(5t)$	$2.2\cos(5t) + \epsilon_t \sim N(0, 0.01)$
2	$-2.4\cos(1.9t)$	$-0.7\cos(2.2t) + \epsilon_t \sim N(0, 0.01)$
3	$0.6\cos(0.5t)$	$0.7\cos(0.5t) + \epsilon_t \sim N(0, 0.01)$

Training. We train independent SIREN models for each combination of sensors and failure modes. For each of the SIREN models, we initialize 2 internal layers of 256 nodes with sine activation and 1 output sine layer. We also use the expert's strong supervision when the model misclassifies the labels. We repeat this experiment for 25 iterations(assuming the machine is breaking down 25 times) for each of the failure modes.

Results. Table 2 reports the accuracy of root cause classifications and Figure 4 shows the MSE for s_1. We find that a small time window causes the classifier to slowly converge, as shown in Figure 4. The accuracy improves as the time window expands, but the small time window does not affect the classification accuracy too much, reflecting the expert's initial accuracy. Function recovery is possible with around 10 to 15 iterations, regardless of the time window sizes.

Table 2: Accuracy of model classification for failure modes, varying the window size .

Window Size	F1	F2	F3	F1,F2	F1,F3	F2,F3	F1,F2,F3
WS = 5	1.00	0.88	0.96	0.96	0.88	0.96	0.72
WS = 15	0.96	0.96	1.00	0.84	0.92	1.00	0.92
WS = 25	1.0	1.00	1.00	1.00	1.00	0.96	0.92
WS = 35	0.96	0.96	1.00	0.92	1.00	1.00	1.00
WS = 45	1.00	1.00	1.00	1.00	1.00	0.96	0.96

6.2 Experiment 2: Noise variance

Here we examine how noise in the sensor measurements affects function recovery and failure mode classification. The approach is the same as Experiment 1, but with a fixed window size of $T = 25$, and the actual equation has a range of different noise variances $\sigma_i^2 \in \{0.1, 0.2, 0.4, 0.6, 0.8, 1\}$.

Results. The failure mode classification accuracy is shown in Table 3, with MSEs for fitting s_1 in Figure 6. The classification accuracy drops as the sensor observations become more noisy. For larger variances, the MSEs vacillate instead of converging to their lower bound, meaning that the model fails to recover the underlying mean function. Note that this happens with a σ_i^2 of approximately 0.4, where accuracy drops significantly across SIREN models.

Table 3: Accuracy of model classification for failure modes with different noise variances.

Noise Variance	F1	F2	F3	F1,F2	F1,F3	F2,F3	F1,F2,F3
$\sigma_i^2 = 0.1$	1.00	1.00	1.00	0.96	0.92	0.96	0.96
$\sigma_i^2 = 0.2$	0.92	1.00	1.00	0.84	1.00	0.96	0.92
$\sigma_i^2 = 0.4$	0.96	0.96	0.8	0.64	0.88	0.88	0.40
$\sigma_i^2 = 0.8$	0.32	0.40	0.16	0.52	0.52	0.16	0.44
$\sigma_i^2 = 1.0$	0.64	0.08	0.20	0.28	0.24	0.12	0.24

6.3 Experiment 3: Expert accuracy and supervision

There are two critical cases where the model's performance depends on expert judgment: 1) when the experts do not have full knowledge of the underlying physical equations for the failure modes, and 2) when the experts are unavailable for providing strong supervision to correct misclassifications. We examine the model's performance in each scenario. To illustrate model performance, we focus only on the classification of failure mode F_2.

Expert's prior knowledge. We vary two components of the expert's assessment: 1) the amplitude of the sine or cosine functions and the frequency of the sine or cosine functions. We use a training strategy similar to the previous experiments, see Table 4. Scale parameters θ_1 are generated from $\{-10, -5, -2, 0, 2, 5, 10\}$ and frequency parameters θ_2 are generated from $\{0.1, 1, 1.5, 2, 2.5, 3, 3.5\}$. When we alter θ_1, θ_2 is fixed to be the same as the frequency parameter in actual equation (2.2). When we alter θ_2, θ_1 is taken to be the same as the scale parameter in the actual equation (-0.7). We use 30 iterations instead of 25, to fully observe $F2$. We assume the experts give correct classification labels when the model misclassifies.

Results. We show the performance of the model under different prior knowledge, with its accuracy in detecting $F2$ and the MSE convergence for $F2$. For simple periodic functions, if enough data are given to the SIRENs for each of the iterations and an expert corrects misclassified labels, the model is able to give relatively accurate predictions. In both cases, the MSE between the true function and predicted function converges in Figure 5. We further compare the accuracy results without expert supervision (labels), provided in Table 5.

Table 4: Parameters for Data Generation in Experiment 3.

Fault Mode	Expert's Assumption	Actual Equation
1	$2\cos(5t)$	$2.2\cos(5t) + \epsilon_t \sim N(0, 0.01)$
2	$\theta_1 \cos(\theta_2 t)$	$-0.7\cos(2.2t) + \epsilon_t \sim N(0, 0.01)$
3	$0.6\cos(0.5t)$	$0.7\cos(0.5t) + \epsilon_t \sim N(0, 0.01)$

Table 5: The accuracy of the model under various parameters, comparing strong and no supervision.

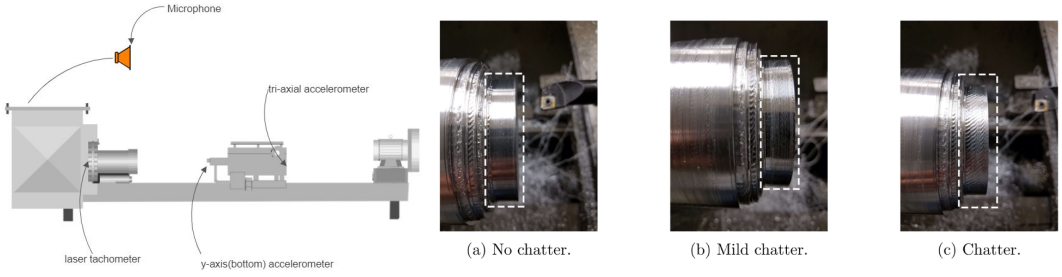
Parameters	**Expert Labels**	**No Labels**
$\theta_1 = -5, \theta_2 = 2.2$	1.0	0.86
$\theta_1 = -2, \theta_2 = 2.2$	1.0	1.0
$\theta_1 = 0, \theta_2 = 2.2$	1.0	1.0
$\theta_1 = 2, \theta_2 = 2.2$	1.0	0.0
$\theta_1 = 5, \theta_2 = 2.2$	1.0	0.0
Accuracy	**Expert Labels**	**No Labels**
$\theta_2 = 1.0, \theta_1 = -0.7$	1.0	0.0
$\theta_2 = 1.5, \theta_1 = -0.7$	1.0	0.0
$\theta_2 = 2.0, \theta_1 = -0.7$	1.0	1.0
$\theta_2 = 2.5, \theta_1 = -0.7$	1.0	1.0
$\theta_2 = 3.5, \theta_1 = -0.7$	1.0	0.0

Supervision. We conduct the same experiments as shown in Table 4, varying the scale θ_1 and frequency θ_2 with two common ranges of parameters, $\theta_1 \in \{-5, -2, 0, 2, 5\}$, and $\theta_2 \in \{1, 1.5, 2, 2.5, 3\}$. However, here we assume that the expert cannot provide any labels.

Results. The MSE for convergence is shown in Figure 7 and the accuracy in comparison with strong supervision is shown in Table 5. Without the expert's correction of the wrongly classified labels, the model does not properly modify the internal physical neural network associated with the failure mode, leading to non-convergence of the MSE. Our experiment shows that the expert's strong supervision is important, especially in the event of limited data, to initialize the physical models.

7 Turning data and chatter diagnosis

In this section we apply our approach to a real machining dataset by Khasawneh *et. al.* [11] who examine tool chattering during a turning task under different settings for workpiece and cutting edge. The sensor signals are simultaneously collected at a sampling frequency of 160 kHz. The processed data are low-pass filtered using a Butterworth filter of order 100, and then subsampled to 10 kHz. We use the processed and tagged x-axis signals from the triaxial accelerometer. In addition to desired normal operation (stable, s), there are three non-stable modes: intermediate chatter (precursor to failure) (i), chatter (failure mode) (c), and unknown (failures other than chattering) (u), see Figure 2. In prior work, classifying the chattering data was accomplished by extracted features from signals based on Topological Data Analysis (TDA) and nonlinear time series analysis [12] (accuracy score of 0.97), or the usage of wavelet packet transform (WPT) and ensemble empirical mode decomposition (EEMD) [27] (accuracy score of 0.94).



(a) No chatter. (b) Mild chatter. (c) Chatter.

Figure 2: The experimental setup showing the workpiece, the cutting tool, and the attached accelerometers (left). The surface finish corresponding to different chatter labels[27] (right).

While the methods in [12] and [27] train and test the model on the entire dataset, we simulate a zero-shot environment with few or no data in the beginning and train our classification model sequentially. We have also made use of different levels of EEMD[5] to denoise the input chatter data. The chattering data consists of variable length time-series data-points which are split into non-overlapping chunks with time window of 500 observations each for consistency. The model observes one chunk at each instance which is classified and subsequently used to update the model. For SIREN, we use 3 hidden layers of 256 nodes, and an output linear layer. For LSTM, we apply 2 hidden layers of 256 nodes, and an output linear layer. The process proceeds for 1000 iterations, with expert's supervision once every 10 iterations, as well as each of the first 10 iterations. For expert's prior guesses, we provide approximate equations based on the judgement of a single chunk(500 data) for each scenario, see Table 6.

Table 6: Expert's Prior Guesses for Chattering Data

Fault Mode	**Expert's Assumption**
stable(s)	$0.004\sin(2400 \cdot t) + 0.08+ \sim N(0, 0.0001)$
unknown(u)	$0.005\sin(6000 \cdot t) + 0.015+ \sim N(0, 0.0001)$
intermediate(i)	$0.002\sin(14000) + 0.004+ \sim N(0, 0.0001)$
chatter(c)	$0.015\sin(3000 \cdot t) + 0.001+ \sim N(0, 0.0001)$

7.1 Result

The average classification accuracy for all scenarios in the chattering dataset is shown by Table 7 for SIREN and LSTM models respectively. The rows indicate the original data, and the denoised data decomposed by $1, 2, 3, 4$ intrinsic mode functions(IMFs) during each EEMD decomposition. Figure 3 shows the accuracy over the number of iterations for the SIREN and LSTM models on the original data.

Table 7: The accuracy of SIREN- and LSTM-based BN, under different IMFs of denoised data

Data with different IMFs	**SIREN**	**LSTM**
Original	0.63	0.62
1 IMF	0.58	0.61
2 IMFs	0.86	0.76
3 IMFs	0.28	0.32
4 IMFs	0.21	0.20

Both SIREN and LSTM models achieve best accuracy for denoised data at 2 IMFs (0.86 and 0.76 respectively). As expected, the models perform better with increasing data updates. Both models underperform due to increased data complexity (data generated from complex mechanical systems) and noise present in the chattering dataset. Our model performs less accurately than state-of-the-art models introduced by Khasawneh *et. al.* [12] and Yesilli *et. al.* [27], which rely on specific physical knowledge or feature extractions that are tailored by experts for the chattering processes. Moreover, their classification results are reported for entire variable length time sequences. Our model does not rely heavily on prior physical knowledge (underlying PDEs, which are often unknown in practice) and feature extraction. We believe this setup allows for better generalization to more than just chattering signals, for example, acoustic signals and imagery data.

8 Conclusion

We provide an initial test of an online evidential reasoning machine learning model under various window sizes at each time of query, different levels of noise, inaccurate prior knowledge, and expert supervision. Overall, our results suggest that 1) function recovery and accuracy are high when the expert proposes a reasonably accurate model of the physical process and is able to supervise the model when it makes errors, even when the number of observations are small; 2) function recovery and accuracy may degrade when the expert is either very wrong or there is too much noise in the system; 3) a small amount of strong supervision from the expert can realign the algorithm in the presence of inaccurate initial estimates or noise.

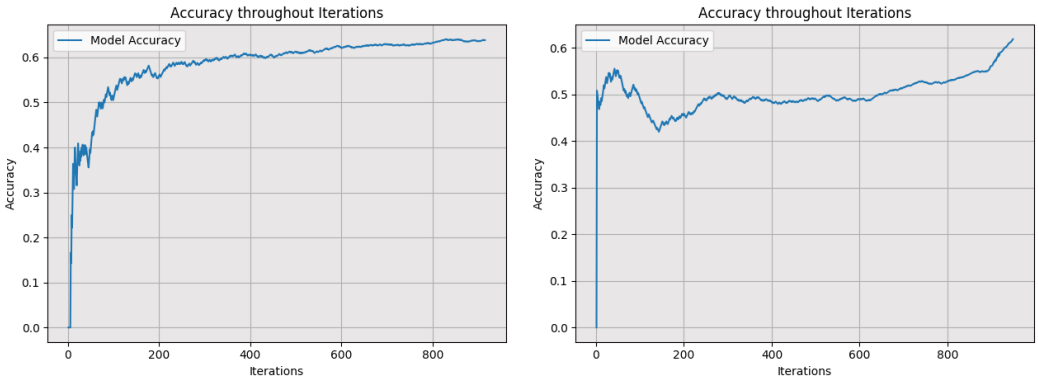


Figure 3: Left: Accuracy of first 1000 iterations for BN training with SIREN. Right: Accuracy of first 1000 iterations for BN training with LSTM.

9 Broader impact and future work

Modern machine learning models used for research have increased in complexity and size, leading for example to significant energy consumption and resource use. The implications of such have not been fully explored and are still in nascent stages. García-Martín *et al.* [6] describe methods to estimate energy consumption of machine learning models. At the same time many emerging applications suffer from constrained or sparse data sets that prevent supervised learning models from being fully effective. Our future work aims to continue study of the effects of expert-guidance in learning systems to make use of all possible knowledge of the system being modeled, hopefully leading to both reduced model complexity and improved model inferencing.

Another area of interest is joint learning of experts and machines. Interesting questions include "Can machines guide experts in learning?" and "Can machines teach experts new physics?". Feedback loops through which machines can communicate to experts with information and suggestions can improve learning and teaching. In the current framework, this would be advising the experts on the effectiveness of the priors provided or the causal relations.

Finally, inclusion of neural differential equations (ODEs, PDEs) is an area where experts may be able to provide stronger functional form assumptions to the network. Chen *et al.* [3] represent the transformations of inputs through a neural network as a continuous ordinary differential equation rather than discrete layers. Maulik *et al.* [19] apply the approach to the viscous Berger's equation. Recently, Kidger *et al.* [13] use controlled differential equations to allow neural differential equation time series models with missing data and irregularly spaced temporal observations. It is an interesting direction to explore the utility of experts in suggesting such differential equation based regularizers for neural networks.

References

[1] Sara Abdollahi, Alexander Davis, John H Miller, and Adam W Feinberg. Expert-guided optimization for 3d printing of soft and liquid materials. *PloS one*, 13(4):e0194890, 2018.

[2] Tarem Ahmed, Boris Oreshkin, and Mark Coates. Machine learning approaches to network anomaly detection. In *Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques*, pages 1–6. USENIX Association, 2007.

[3] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[4] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pages 3140–3150, 2019.

[5] Said Gaci. A new ensemble empirical mode decomposition (eemd) denoising method for seismic signals. *Energy Procedia*, 97:84 – 91, 2016. European Geosciences Union General Assembly 2016, EGU Division Energy, Resources the Environment (ERE).

[6] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88, 2019.

[7] E. D. Gennatas, J. H. Friedman, L. H. Ungar, R. Pirracchio, E. Eaton, L. Reichman, Y. Interian, C. B. Simone, A. Auerbach, E. Delgado, M. J. Van der Laan, T. D. Solberg, and G. Valdes. Expert-augmented machine learning, 2019.

[8] Sheng Hong, Zheng Zhou, Chen Lu, Baoqing Wang, and Tingdi Zhao. Bearing remaining life prediction using gaussian process regression with composite kernel functions. *Journal of Vibroengineering*, 17(2):695–704, 2015.

[9] Feng Jia, Yaguo Lei, Liang Guo, Jing Lin, and Saibo Xing. A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. *Neurocomputing*, 272:619–628, 2018.

[10] Dinu Kaufmann, Sonali Parbhoo, Aleksander Wieczorek, Sebastian Keller, David Adametz, and Volker Roth. Bayesian markov blanket estimation. In *Artificial Intelligence and Statistics*, pages 333–341, 2016.

[11] Andreas; Yesilli Melih Khasawneh, Firas; Otto. Turning dataset for chatter diagnosis using machine learning. *Mendeley Data*, V1(doi: 10.17632/hvm4wh3jzx.1), 2019.

[12] Firas A. Khasawneh, Elizabeth Munch, and Jose A. Perea. Chatter classification in turning using machine learning and topological data analysis. *IFAC-PapersOnLine*, 51(14):195–200, 2018.

[13] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926*, 2020.

[14] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[15] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1939–1947, New York, NY, USA, 2015. Association for Computing Machinery.

[16] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[17] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.

[18] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection, 2016.

[19] Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, and Daniel Livescu. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena*, 405:132368, 2020.

[20] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[21] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[22] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.

[23] Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, pages 1–13, 2019.

[24] Syahril Ramadhan Saufi, Zair Asrar Bin Ahmad, Mohd Salman Leong, and Meng Hee Lim. Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review. *IEEE Access*, 7:122644–122662, 2019.

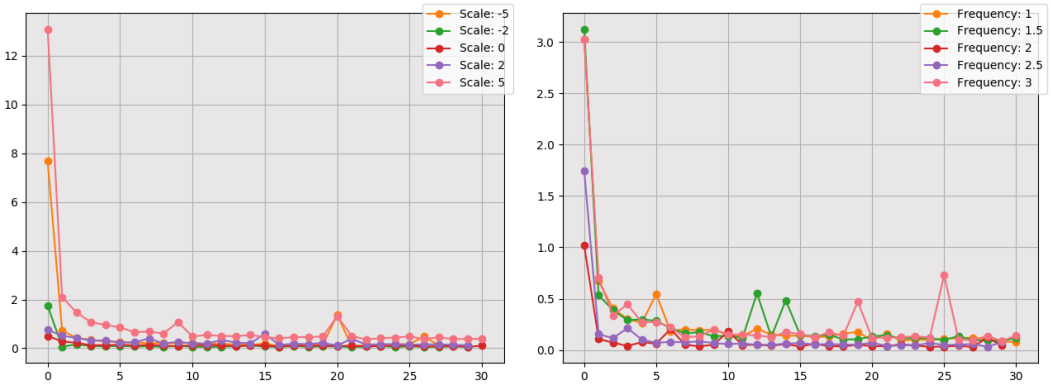
[25] Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020.

[26] Yichong Xu, Aparna Joshi, Aarti Singh, and Artur Dubrawski. Zeroth order non-convex optimization with dueling-choice bandits. In *Conference on Uncertainty in Artificial Intelligence*, pages 899–908. PMLR, 2020.

[27] Melih C. Yesilli, Firas A. Khasawneh, and Andreas Otto. On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP Journal of Manufacturing Science and Technology*, 28:118 – 135, 2020.

10 Appendix and addtional plots

Figure 4: Mean square error over 25 iterations by error mode for different time windows(SIREN).



(a) Change the expert's guess on scales.

(b) Change the expert's guess on frequencies.

Figure 5: MSE convergence between true and predicted functions(SIREN)

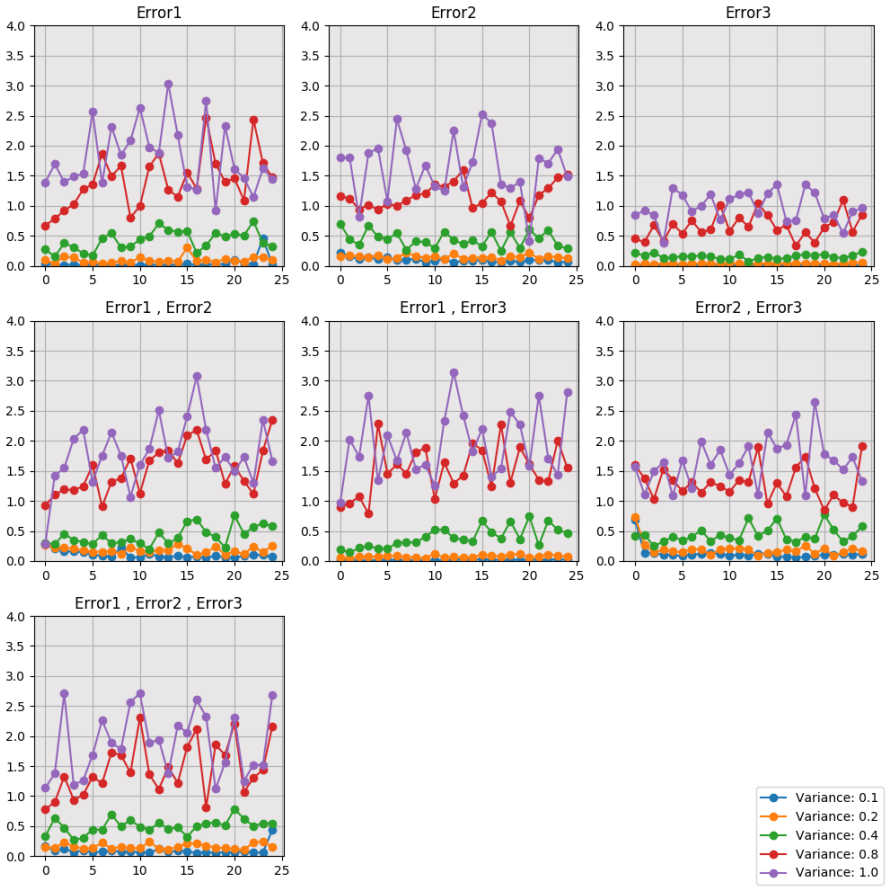
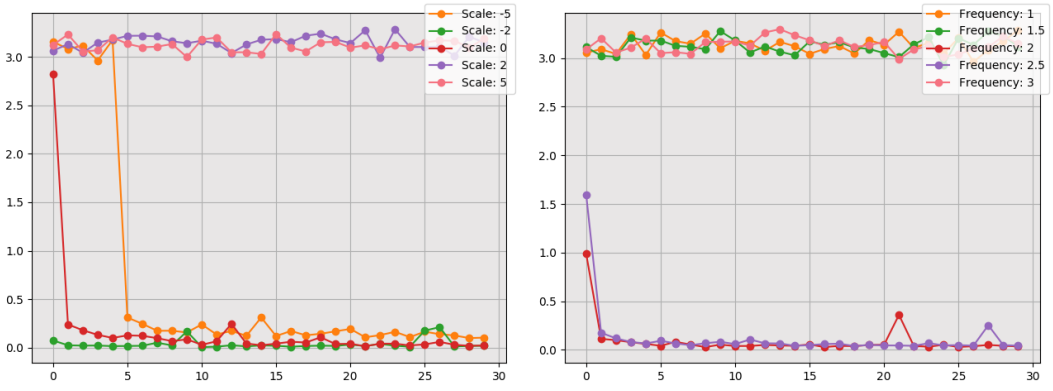


Figure 6: Mean square error over 25 iterations by error mode for different variance of data(SIREN).



(a) Change the expert's guess on scales.

(b) Change the expert's guess on frequencies.

Figure 7: With weak expert's intervention, MSE convergence between true and predicted functions(SIREN)