Machine Learning Based Porcine Respiratory and Reproductive Syndrome Forecasting

By

MOHAMMADSADEGH SHAMSABARDEH
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Beatriz Martínez-López, Chair

Soheil Ghiasi

James Sharpnack

Committee in Charge

2022

Abstract

The livestock industry plays an important role in the global food chain and provides the main source of protein for human consumption. Pork production provides more than one-third of total meat protein worldwide. There is a gap between the amount of data available in the swine industry and its effective use in analytical models and the decision making process of farm management. This work tries to fill this gap by building a data-driven decision framework. This framework allows for risk-based and early intervention in the swine industry which mitigates the overall cost.

First, we focus on the most challenging and costly viral infectious diseases impacting the swine industry called the Porcine Reproductive and Respiratory Syndrome (PRRS). We build a framework to forecast the risk of having a PRRS outbreak on a farm. This forecasting allows for early detection of disease outbreaks and could direct risk-based, and thus more cost-effective, interventions. Machine learning algorithms were trained using multi-scale data (pig group-, farm-, and area-level data). For the first time, on-farm, between-farm, and environmental variables, including farm location, pig movements, production parameters, diagnostic data, and climatic information, were combined for the prediction of PRRS outbreaks. Multi-scale datasets were merged via feature creation, followed by the wrapper and filter feature selection, to find those feature subsets with the best forecasting performance. The predictive value of each features selection mechanism was evaluated in terms of its stability. Numerical results demonstrate good forecasting performance in terms of area under the ROC curve.

Furthermore, we leverage a semi-supervised variational auto-encoder (VAE) deploying Long Short Term Memory (LSTM) to predict the mortality rates (mummified and stillborn) and farrowing rate in the production system. The PRRS can be one of the underlying mortality factors. The use of VAE allows for handling the missing data by building a probabilistic model. We learn the target variable by learning a latent representation using the generative model for samples with unobserved target value, and then learning a generative semi-supervised model, using this representation instead of the raw data.

Finally, a factorized generative model is applied based on fine grained semi-synthatic data for the study of PRRS virus. Using this model, we can predict the PRRS outbreak in all farms of a swine production system by capturing the spatio-temporal dynamics of infection transmission based on the intra-farm pig-level virus transmission dynamics, and inter-farm pig shipment network. We simulate a PRRS infection epidemic based on the shipment network and the SEIR epidemic model using the statistics extracted from real data provided by the swine industry. We develop a hierarchical factorized deep generative model that approximates high dimensional data by a product between time-dependent weights and spatially dependent low dimensional factors to perform per farm time series prediction. The prediction results demonstrate the ability of the model in forecasting the virus spread progression with average error of NRMSE = 2.5%.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to give my warmest gratitude to my PhD advisor Professor Beatriz Martinez-Lopez who unconditionally supported me through all the hardships of this route. Her guidance, encouragement, kindness, and endless generosity carried me through this journey. I would like to thank her for always being there to discuss anything despite her various roles at school and supporting me by any means and in any capacity.

I would like to extend my thanks to my committee members, Professor Soheil Ghiasi and Professor James Sharpnack for their time, feedback and suggestions.

I am grateful to everyone I've collaborated with and a special thanks to my dear colleagues Dr. Kathleen O'Hara and Dr. Bahar Azari for being resourceful and being there when I needed their feedback and for all that I learned from them. A special thanks to Dr. Amirreza Farnoosh for sharing his knowledge.

And my most important thanks to my love ones, I would like to give my deepest and sincerest gratitude to my parents and my sisters for their unparalleled love and support.

I would like to acknowledge and thank the National Science Foundation (NSF) for funding this work under NSF BIGDATA:IA Award #1838207 and NSF Track-D Award #2134901. I also would like to acknowledge swine industry collaborators for the provision of data.

Chapter 1

Motivation, Background and Literature Review

In this chapter the motivation behind this work is discussed, a brief description of some models that are used and examples from the literature are provided.

1.1 Motivation and Significance

Infectious disease outbreaks have caused dramatic financial cost and affected the lives of many human and animals during history. From the the loss of 50 million people in 1918 H1N1 pandemic Tumpey et al. 2005 to the recent COVID-19 pandemic. The number of outbreaks and the rates of spread of some pathogens have been increasing in recent decades VanderWaal and Deen 2018. This can be an alarming threat to both animals and human lives. Especially the transmission of pathogens from animals to humans, such as the H1N1 swine flu pandemic in 2009, highlights the importance of controlling the animal infectious disease outbreaks for human safety Smith et al. 2014.

The live stock industry plays an important role in the global food chain and provides the main source of protein for human consumption. Pork production, provides more than one-third of total meat protein worldwide (Food and Agriculture Organization) and such demand has created areas with closely located and densified pig population which improves the efficiency for the required frequent movement of pig and food among farms in the production system. All these factors increase the chance of having an outbreak Martinez 2002. These outbreaks can bring food insecurity by causing animal loss and restricting the required trades among different farms to keep the production system sustainable. The animal welfare, human risk of death, food insecurity and economic impacts requires academia, industry and governmental authorities to investigate and find better solution to mitigate and control these outbreaks. As a result this work focus on addressing the most challenging and costly viral disease of swine industry, the Porcine Reproductive and Respiratory Syndrome Mateu and Dıéaz 2008, by collaborating with the largest swine production system in United States and with the funding from National Science Foundation.

To mitigate and control the Porcine Reproductive and Respiratory Syndrome outbreaks, different approaches including the experimental vaccine development Nodelijk et al. 2001, theoretical vaccine Bitsouni et al. 2019 modeling, immunological treatment and prevention methods Murtaugh and Genzow 2011, statistical Evans et al. 2010; Islam et al. 2013 and analytical modeling have been conducted. In recent years the researchers have been trying to exploit the machine learning approaches to build better models to understand and predict the occurrence of outbreaks in live stock industry Garcia et al. 2020. However, there are limited machine learning works on Porcine Reproductive and Respiratory Syndrome.

Due to the high level of specialization in production systems, a vast amount of data has been collected in the livestock industry, in particular, swine industry. Specifically, data is gathered in all processes of pig production: pig demographics and production performance, pig movements, farm testing etc. Note that the data is multilevel (i.e., pig production performance indices, pig movement networks between farms, and pathogen test result), constantly changing over time and increasing in size.

Unfortunately, while this vast amount of data is available, its usage in animal health remains circumstantial, and is usually restricted to simple descriptive statistics or sequencing and molecular analyses for specific aspects of animal breeding and pathogen diagnostics. To the best of our knowledge, there is no data-driven decision framework that effectively integrates the multi-level data to better study the complex nature of a Porcine Reproductive and Respiratory Syndrome. This large gap between the data availability and its effective usage motivate the proposed work in this thesis. Our specific aim here is to develop a principled data-driven decision framework that facilitates the early detection and fast control of PRRS, which will save swine producers millions of dollars annually. It will also provide a revolutionary approach that can be adapted to other diseases and other livestock species.

The challenge: PRRS is currently the most challenging and costly viral infectious disease in the swine industry. This emerging disease was discovered first in 1980s in the US and rapidly spread to many swine production countries in Europe and Asia. Now, most of the pig producing countries in the world are infected. PRRS has a huge economic impact on the swine industry in the US and across the world. In the US only, it causes an estimated economic loss of $664 million annually, 55% associated to growing pigs and 45% to breeding farms Holtkamp, Kliebenstein, et al. 2013. The complexity in PRRS control is mainly due to the easy transmission within and between farms and the wide variability of the PRRS virus (PRRSV) due to mutations. PRRSV, an RNA virus, mutates easily, and thus continuously challenges the pig immunity and makes vaccine development difficult. As a result, the current available vaccines are only partially protective. Key factors in the PRRS prevention and control are the early detection of infection, cost-effective monitoring/testing of farm health status, and efficient implementation of immunization and biosecurity (internal and external) mechanisms. To address this challenge, this work tries to build a data-driven decision framework for systematic PRRS prevention and control, based on the multi-level data sources collected during swine production, using advanced data mining and machine learning techniques.

1.2 Predictive Models

The predictive classification problem can be viewed as a task of function approximation in which a binary or categorical variable y can be predicted from a set of predictors X. The predicted variable $y = f(X)$ is called response (target or dependent variable) and the predictors are called features (covariates, attributes, or independent variables). The mapping f is the true underlying model that describes input-output relation and can be approximated by a function $\hat{f}$ by fitting the labelled training data X into the model $\hat{f}$. The task would translate into an optimization problem of minimizing the prediction error between the training labels $y = f(X)$ and predicted values $\hat{y} = \hat{f}(X)$. If f be as close as possible to $\hat{f}$ then the prediction error on unseen data points are expected be be minimized. The property of predicting with low error the labels of a new test set x is called generalization.

Considering the case of predicting a binary variable y where the $y = 1$ means a farm is unhealthy and $y = 0$ means it is healthy, this prediction can be performed with uncertainty. We denote the probability distribution over the possible label, given the input training set D, input test vector x and the model M by $P(y|x, D, M)$. It is notable to mention that in case of binary classification $P(y = 1|x, D, M) + P(y = 0|x, D, M) = 1$ and we only need to determine one class. In other words we looking for the most probable class healthy or unhealthy for a vector of input x corresponding to a farm and expressed mathematically as: $\hat{y} = argmaxP(y = c|x, D, M)$ for $c = 0, 1$ (Murphy 2012). In the rest of this section, different models for classification and the related literature in the field of veterinary medicine are briefly discussed.

Logistic Regression

If a model of $P(y|x)$ is build by mapping input x to binary output y then we are building a discriminative classifier which can discriminate between different class labels. Logistics regression can be mathematically expressed as Murphy 2012: $p(y|x, w) = Ber(y|\mu(x))$, where $\mu(x) = E[y|x] = p(y = 1|x)$ and by defining $\mu(x) = sigm(w^T x)$, we can ensure that $0 <= \mu(x) <= 1$. The $sigm(\eta) = \dfrac{1}{1 + exp(-\eta)}$ referrers to sigmoid function, an S-shaped or squashing function which maps the input into a probability space by bounding it between zero and one. We can obtain the logisitc regression formula as $P(y|x, w) = Ber(y|sigm(w^T x))$. When working with logistic regression the output probability needs to be converted to the class label. This requires to have a threshold for deciding the range of probability for each class. In other words $P(y = 1|x) > threshold$, would result in the data point x be classified as class $y = 1$. In this case everything on one side of the hyper-plane will belong to one class and the points on the other side of the hyper-plane belong to the other class. If the hyper-plane can well different the classes then the problem in hand is of linear nature, otherwise a none-linear decision boundaries are required.

To estimate the parameters of logistic regression, the maximum likelihood estimation is used. The negative log likelihood for logistic regression also known as cross entropy can be written as: $NLL(w) = -\sum_{n=1}^{N}[y_i log(\mu_i) + (1 - y_i) log(1 - \mu_i)]$, where $\mu_i = sigm(w^T x_i)$. If we assume that $y_i \in -1, 1$ instead of $y_i \in 0, 1$ the formula can be rewritten as: $NLL(w) = \sum_{n=1}^{N} log(1 + exp(-\tilde{y}_i w^T x_i))$. To find the NLL, an optimization problem is solved by calculating the gradient and the hessian.

When models with flexible number of paramters such as logistic regression are used, the model might overfit the training data if the model has higher degree of freedom than required for the underlying problem. This happens when the model learns the training data too well, it learns the noise and slight changes in the training data, but does not generalize and perform poorly on the testing data set. The problem of selecting the number of parameter is a model selection problem. A suitable model complexity can be decided by comparing different models on the number of miss classifications on a testing data set. The testing set used for this purpose is called validation data set and typically is composed of 20 percent of the original data. If the order of the data point does not matter, methods such as leave one out and K-fold cross validation are used Hastie, Tibshirani, et al. 2009. However, in this thesis we will be required to choose the data points that chronologically appear after the training data points.

One important assumption in logistic regression is that variables are not linear combinations of each other Midi, Sarkar, and Rana 2010. If there are two or more predictor variables that are highly correlated then we are facing the multicollinearity problem where the coefficient estimates may change erratically in response to small changes in the model or the data. The multicollinearity make coefficients unstable. The general rule is that if correlation coefficient between two features is greater than 0.9, the multicollinearity is a serious problem. Multicollinearity does not decrease the predictive power or reliability of the entire model Midi, Sarkar, and Rana 2010 however the interpretation about the importance of each predictor is not reliable and variable selection in these situation become very difficult. The are a range of solutions for Multicollinearity problem in logistic regression including dropping variables, combining variables into an index, and testing hypothesis about sets of variables, increasing the sample size if possible. Another commonly used approach is regularization. All the features are used for prediction but the coefficients of some features are shrunk to zero, as a result those variables are automatically not selected for prediction. Ridge Hoerl and Kennard 1970 and Lasso Tibshirani 1996 are two mostly used regularization approaches in which a penalizing term is added to the cost function of the model. Ridge and lasso perform a trading off a small increase in bias for a large decrease in variance of the predictions, hence they may improve the overall prediction accuracy Hoerl and Kennard 1970. In Lin et al. 2013, the authors utilize the group lasso algorithm for logistic regression to construct a risk scoring system for predicting PPRS outbreak. The authors of Koene et al. 2012 use ridge to classify animals based on serum protein profiles.

For the above mentioned formulation of logistic regression, a small change in the training data may cause a large change in the coefficient estimates and the model may have high variance. Ridge regression and lasso perform by trading off a small increase in bias for a large decrease in variance of the predictions, hence they may improve the overall prediction accuracy. Ridge logistic regression all the coefficient, except the intercept are shrunk by imposing a penalty on their size Hastie, Tibshirani, et al. 2009. The parameters estimates are obtained by minimizing the log-likelihood function:

$$w^{\text{ridge}} = \underset{w}{\operatorname{argmin}} \sum_{n=1}^{N} [y_i log(\mu_i) + (1-y_i) log(1-\mu_i)] + \lambda \sum_{j=1}^{P} w_j^2 \tag{1.1}$$

where $\mu(x) = sigm(w^T x) = \dfrac{1}{1+exp(-w^T x)}$, λ is the controls the amount of shrinkage. The larger the λ the more shrinkage is applied. There are P predictors, and w_j corresponds to the coefficient of the jth predictor

If there are two correlated variables the coefficient of one can be positive and the other one be negative. In this case, a change in one coefficient can result in change of the coefficient of the correlated variable while having the same prediction value. This results in poor coefficient estimation. This problem can be addressed by specifying the number of coefficient t in the ridge formula, as:

$$w^{\text{ridge}} = \underset{w}{\text{argmin}} \sum_{n=1}^{N} [y_i log(\mu_i) + (1-y_i) log(1-\mu_i)]$$

$$\text{subject to} \qquad \sum_{j=1}^{P} w_j^2 \leq t \tag{1.2}$$

In 1.2, increasing the value of λ will shrink more the coefficient toward zero but does not set them exactly to zero which might become problematic for feature selection. This problem can be addressed by penalizing the log likelihood regression with L1 instead of L2.

The coefficients in the L2 regularization, named as lasso, can be obtained using:

$$w^{\text{lasso}} = \underset{w}{\text{argmin}} \sum_{n=1}^{N} [y_i log(\mu_i) + (1-y_i) log(1-\mu_i)] + \lambda \sum_{j=1}^{P} |w_j| \tag{1.3}$$

Similar to ridge, the intercept is not penalized. In addition to shrinkage, lasso makes feature selection by forcing some of the coefficient exactly to zero and as a result can improve model interpretability. If there are some features with larger coefficient the lasso is expected to perform better and in case of closely valued coefficients ridge is expected to perform better.

The coefficient estimates in both ridge and lasso whose values depend on how large the value of λ should be. The cross validation methods can be used for tuning the value of λ.

Support Vector Machines

Support vector machines (SVM)s are kernel-based algorithms with sparse solutions, i.e., the prediction for a new test point requires kernel function evaluation only at a subset of the training points. SVM uses this subset of the training points to find a separating hyperplane between data of different classes. SVM parameter estimation distill into a convex optimization problem where any local optimum is a global one. In this section, we describe the SVM formulation for the two-class classification problem, which is the interest of our study.

Our training dataset comprises N input vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, with corresponding labels $y_1, y_2, \ldots, y_N$ where $y_n \in \{-1, 1\}$. Our SVM classifier denoted as $f(x)$ is a linear model of form

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b, \tag{1.4}$$

where $\phi(\mathbf{w})$ is the feature-space transformation and b is the bias parameter. For simplicity we first introduce the case that our training data set is linearly separable in feature space, i.e., there exists at least one set of parameters $\phi(\mathbf{w})$ and b such that $f(\mathbf{x}_n) > 0$ for the data points with $y_n = +1$ and $f(\mathbf{x}_n) < 0$ for the data points with $y_n = -1$, shortly $y_n f(\mathbf{x}_n) > 0$ for all training data points. To minimize generalization error, SVM introduces the concept of the margin, which is the smallest distance between the decision boundary and a training data point and tries to select the decision boundary in such a way that the margin is maximized. Intuitively, the solution to the SVM problem involves finding the nearest point to the separating hyperplane, minimum distance training data point $\mathbf{x}$, and maximizing the margin by adapting the parameters $\mathbf{w}$ and b. The minimum-distance margin corresponding to the training data point $\mathbf{x}$ is the perpendicular distance of that point from the hyperplane $f(x) = 0$, where $f(x)$ takes the form (1.4), that is given by $|f(x)|/\|\mathbf{w}\|$. Hence, finding the nearest point $\mathbf{x}_n$ to the separating hyperplane, given that all data points are correctly classified $y_n f(\mathbf{x}_n) > 0$, corresponds to minimizing the following expression

$$\frac{y_n f(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right)}{\|\mathbf{w}\|}. \tag{1.5}$$

Maximizing the margin shown in (1.5), which is the perpendicular distance to the nearest training data point $\mathbf{x}_n$ to the separating hyperplane, with respect to the parameters $\mathbf{w}$ and b yields the solution to the SVM problem as

$$\underset{\mathbf{w},b}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{n} \left[y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right)\right] \right\}. \tag{1.6}$$

Solution of the SVM Problem

Direct solution the optimization problem in (1.6) is quite complex, and in practice, it is converted into an equivalent problem that is much easier to solve. This is possible by rescaling the parameters $\mathbf{w}$ and b in such way that set $y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right)$ for the nearest point to the hyperplane. Note that enforcing the mentioned equality does not change the distance from any point $\mathbf{x}_n$ to the decision hyperplane, $y_n f(\mathbf{x}_n)/\|\mathbf{w}\|$. Any other data points satisfy the constraints

$$y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) \geq 1, \qquad n = 1, \dots, N, \tag{1.7}$$

which is known as the canonical representation of the decision hyperplane. The new optimization problem then becomes that of maximizing $\|\mathbf{w}\|^{-1}$, which is equivalent to minimizing

$$\begin{aligned} &\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to:}\quad & y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) \geq 1, \qquad n = 1, \dots, N. \end{aligned} \tag{1.8}$$

The new SVM problem is a quadratic programming problem in which we minimize a quadratic function subject to a set of linear inequality constraints. This problem can be solved using *Lagrange multipliers*. Lagrange multipliers variables $a_n \geq 0$ are defined for each of the constraints in (1.8), producing the Lagrangian function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ y_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) - 1 \right\}, \tag{1.9}$$

where $\mathbf{a} = (a_1, \dots, a_N)^{\mathrm{T}}$. The solution to this problem is obtained by setting the derivatives of $L(\mathbf{w}, b, \mathbf{a})$ with respect to $\mathbf{w}$ and b equal to zero, and then substituting $\mathbf{w}$ and b in $L(\mathbf{w}, b, \mathbf{a})$, which gives the dual representation of the maximum margin problem in which we maximize

$$\begin{aligned} \tilde{L}(\mathbf{a}) &= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m y_n y_m k(\mathbf{x}_n, \mathbf{x}_m) \\ \text{subject to:}\quad & a_n \geq 0, \quad n = 1, \dots, N, \\ & \sum_{n=1}^{N} a_n y_n = 0. \end{aligned} \tag{1.10}$$

where $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$. The dual problem is a quadratic programming problem in which we optimize a quadratic function of a subject to a set of inequality constraints.

When class distributions Overlaps

If the training data points are not linearly separable in our feature space $\phi(\mathbf{x})$, we assume that the class-conditional distributions overlap. In this case, instead of aiming for the exact separation of the training data, which leads to poor generalization, the SVM is modified in such a way that it allows for some of the training points to be misclassified. Therefore, instead of implicitly using an error function that gives infinite error for a misclassified data and zero error for correct classification, data points are allowed to be misclassified having a penalty that increases with the distance from that boundary.

To allow for few misclassified data points to have more generalisation, one slack variable for each training data point is defined as $\xi_n \geq 0$ where $n = 1, \ldots, N$, (Bennett and Mangasarian 1992; Cortes and Vapnik 1995). The slack variable are set to zero for data points on or inside the correct margin boundary, $\xi_n = 0$, and for the other points we have $\xi_n = |y_n - f(\mathbf{x}_n)|$. As a result, when a data point is on the decision boundary we have $\xi_n = 1$, and when a data point is on the wrong side we have $\xi_n > 1$. Then, the exact classification constraints shown in (1.7) should be changed into

$$y_n \left(\mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_n) + b\right) \geq 1 - \xi_n, \qquad n = 1, \ldots, N, \tag{1.11}$$

where the slack variables are should satisfy $\xi_n \geq 0$. Data points with $\xi_n = 0$ are classified correctly and are either on the margin or on the correct side of the margin. The data points with $0 < \xi_n \leq 1$ lie inside the margin, but on the correct side of the decision boundary. Finally, those data points with $\xi_n \geq 1$ are on the wrong side of the decision boundary and are misclassified.

The procedure of defining a set of slack variables is known as relaxing the hard margin constraint to have a soft margin that allows for some of the training data points to be misclassified. Note that the introduction of slack variables allows for overlapping class distributions. However, this approach is still sensitive to outliers due to the fact that the penalty for misclassification increases linearly with ξ. Now the goal is to maximize the margin while softly penalizing points on the wrong side of the margin boundary. We therefore minimize

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\mathbf{w}^2\| \tag{1.12}$$

where the parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.

Random Forest

Bagging or bootstrapping is a method for reducing the variance of a prediction function. The bagging method work best for procedures with high variation and low bias, such as trees. In regression, the same regression tree is fitted many times to bootstrap sampled versions of the training data. Then the average of all the results constitute the final result. For classification, a group of trees are responsible for class prediction.

Boosting, also, is a committee method, meaning that, it exploits a group of learners to accomplish a prediction task. Although, unlike bagging, it is a committee of weak learners that evolves over time. Boosting usually dominates bagging on most problems.

Random forests (Breiman 2001) is obtained through a substantial modification of bagging algorithm. Random forests build a large collection of de-correlated trees, and then compute the average result. The performance of random forests is comparable with that of boosting on many problems. In addition, they are simpler to train and tune.

Definition of Random Forests

In bagging, the main idea is to average many noisy models, which are approximately unbiased, and hence reduce the variance. Since trees can capture complex structures in the data and are high variance, they are ideal candidates for bagging. In addition, trees can have sufficiently large depth with relatively low bias. However, trees are quite noisy, and hence, they benefit greatly from the averaging. The assumption is that the trees that are generated in bagging are identically distributed. We say the expectation of an average of N trees is the same as the expectation of any one of them. This means the bias of N bagged trees is the same as that of an individual tree. During the bagging procedure, we aim to improve the variance (see figure. 1.1). This is different from the goal in boosting technique, in which the trees are grown in an adaptive way to remove bias, and hence are not identically distributed.

From statistics, we know that the average N i,i.d random variables, each with variance σ^2, has variance $\dfrac{1}{N}\sigma^2$. In case of identically distributed variables which are not necessarily independent each with variance σ^2 and with positive pairwise correlation ρ the variance of the average is $\rho\sigma^2 + \dfrac{1-\rho}{N}\sigma^2$. If we increase N, the last term converges to zero and the

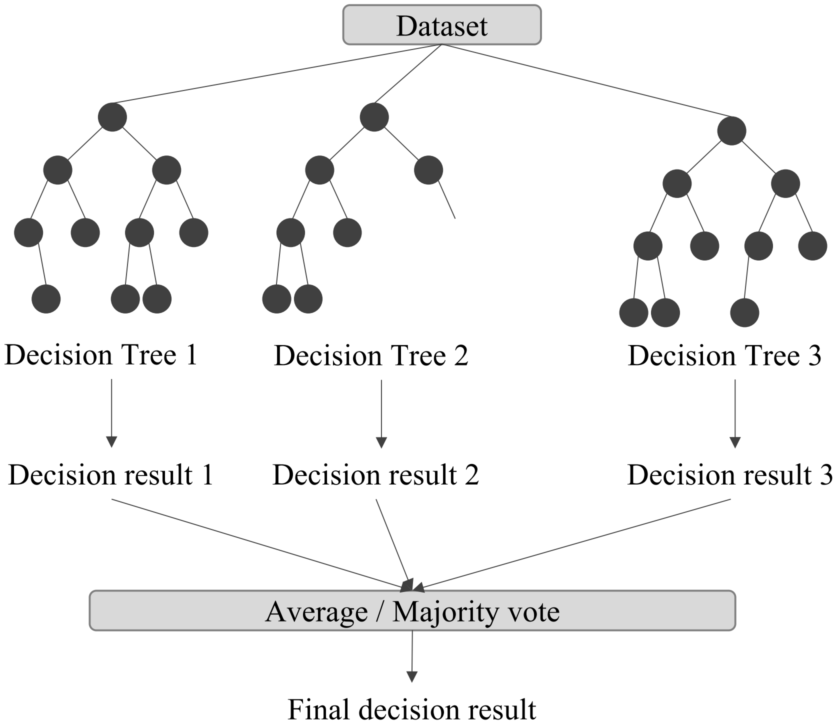


Figure 1.1: A random forest scheme. Random forest goal is to reduce the variance through bagging of trees and to improve this variance reduction by reducing the correlation between the trees.

first term remains. Therefore, the size of the correlation between pairs of random variables governs the variance of the average, and in the case of bagging trees, it limits the benefits of averaging.

In random forests, the goal is to reduce the correlation between the trees to improve the variance reduction induced by bagging. This is achieved by random selection of the input variables in the course of forming each tree. Specifically, each tree is grown on a bootstrapped dataset as follows. Before each split, a set of m input variables is selected at random from D input variables as candidates for splitting (note that $m \leq D$). The value for can be chosen to be $\sqrt{D}$, or as low as one. The set of trees is denoted as $\{T(\mathbf{x}; \Theta_n)\}_1^N$. After building N such trees, the random forest (regression) predictor is of the form

$$\hat{f}_{RF}^N(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} T(\mathbf{x}; \Theta_n). \tag{1.13}$$

The parameters Θ_n characterizes the nth random forest tree in terms of split variables, cut-points at each node, and terminal-node values. It is intuitive to see that reducing m will result in reducing the correlation between any pair of trees in the ensemble, and as a result the variance of the average is reduced. It is worthwhile to note that among all estimators the highly nonlinear ones, such as trees, can be improved by this technique.

Gradient Boosting

Gradient boosting machines (GBMs) J. H. Friedman 2001 are a popular machine learning algorithm that have proven successful across many domains. Boosting is when we build strong learners using a combination/ensemble of weak learners. GBMs build shallow trees in sequence in contrast with random forests, which generate an ensemble of deep independent trees. Each tree learns from the previous one and improves on it. Shallow trees are fairly weak predictive models by themselves, but they can be boosted to produce a powerful committee that can beat other algorithms when properly tuned. In Machado et al. 2019, the authors apply Random Forest, Gradient Boosting and Support vector Machine to integrated environmental and movement factor to identify the occurrence of porcine epidemic diarrhea virus outbreaks.

Boosting Approaches

Supervised machine learning algorithms are mostly composed of a single predictive model such as linear regression models, single decision trees, and support vector machines. There exist other models such as bagging and random forests that owe their power to the combined performance of a set of single models, which together we call them an ensemble. Combining predictions from various base models within an ensemble yields new predictions. Bagging algorithm, and its other similar extension such as random forest, works with averaging that reduces variance. As a result, they are suitable for single models with low bias and high variance. Boosting, however, is a general algorithm for constructing an ensemble from a set of single simple models. Boosting is more efficient when applied to a set of single models with high bias and low variance, and it is usually applied to decision trees.

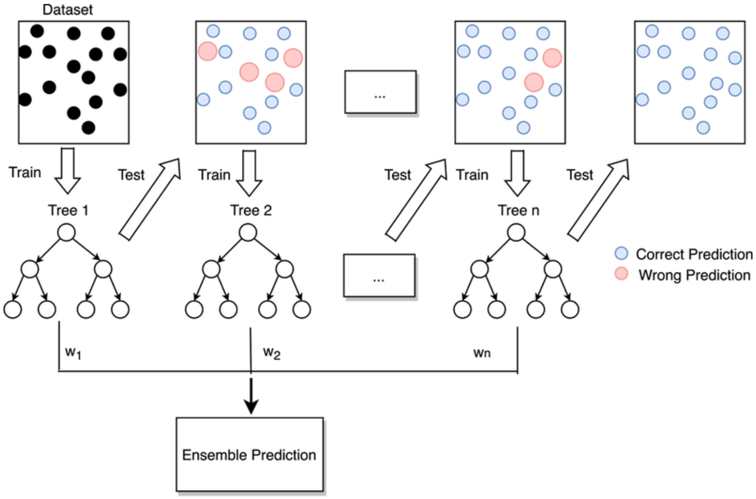


Figure 1.2: Sequential ensemble approach

Sequential Ensemble and Gradient Descent

In boosting a new model is added sequentially to the ensemble. The goal of boosting is to find a compromise between bias and variance by starting with a weak model. A weak model can be a simple one such as a decision tree with only a few splits. Then the performance of the weak model is sequentially boosted by building new trees, where each new tree in the sequence tries to compensate for the biggest error of the previous tree; see figure 1.2.

The essential components of boosting algorithm are the base learners, training of weak models, and the sequential training with respect to errors. In boosting framework, we iteratively improve a weak learning model. Gradient boosting allows for using various classes of weak learners. However in practice, boosted algorithms almost always use decision trees as the base-learner. The error rate associated to a weak model is slightly better than random guessing. In boosting, each model in the sequence improves slightly upon the performance of the previous one. Shallow trees, i.e., trees with relatively few splits, are considered as a weak learner. Boosted trees are grown sequentially, i.e., each tree is grown by using information from previously grown trees to improve performance. For example for boosting regression trees, a decision tree is fitted to the data: $F_1(\mathbf{x}) = y$. Then, the next decision tree is fitted to the residuals of the previous: $h_1(\mathbf{x}) = y - F_1(\mathbf{x})$. A new tree is added: $F_2(\mathbf{x}) = F_1(\mathbf{x}) + h_1(\mathbf{x})$. The next decision tree is fitted to the residuals of $F_2 : h_2(\mathbf{x}) = y - F_2(\mathbf{x})$ and it is added the model. This procedure is continued until it can be stopped based on cross validation. The final model is a stage-wise additive model of b individual trees:

$$f(\mathbf{x}) = \sum_{b=1}^{B} f^{b}(\mathbf{x}). \tag{1.14}$$

GBM Design, Hyperparameters, and Tuning

Different variants of boosting algorithms with focus on classification problems exist, Freund, Schapire, and Abe 1999; Kuhn, Johnson, et al. 2013. A GBM model contains two set of hyperparameters: (i) boosting and (ii) tree-specific. The essential boosting hyperparameters are the *number of trees* and the *learning rate*. The number of trees is the total number of trees in the sequence or ensemble. As a result of the averaging of independently grown trees in bagging and random forests, we can combat the overfiting. Learning rate, also known as shrinkage, affects how fast the algorithm learns and determines how much each tree contributes to the final outcome. The smaller is the learning rate the more accurate the model can be but with the drawback that it will require more trees in the sequence.

The two main tree hyperparameters in a simple GBM model include: Tree depth: Controls the depth of the individual trees. Typical values range from a depth of 3–8 but it is not uncommon to see a tree depth of 1 (Hastie, J. Friedman, and Tibshirani 2001). Smaller depth trees such as decision stumps are computationally efficient (but require more trees); however, higher depth trees allow the algorithm to capture unique interactions but also increase the risk of over-fitting. Note that larger n or p training data sets are more tolerable to deeper trees. Minimum number of observations in terminal nodes: Also, controls the complexity of each tree. Since we tend to use shorter trees this rarely has a large impact on performance. Typical values range from 5–15 where higher values help prevent a model from learning relationships which might be highly specific to the particular sample selected for a tree (overfitting) but smaller values can help with imbalanced target classes in classification problems.

GBMs can have high variability in accuracy which is dependent on their hyperparameter settings (Probst, Bischl, and Boulesteix 2018). Therefor, an appropriate strategy for tuning is necessary. A good approach is to choose a relatively high learning rate, e.g., something around 0.05 to 0.2 such as 0.1. Then the optimum number of trees for this learning rate should be determined. With the fixed tree hyperparameters, the learning rate can be tuned and the speed vs. performance can be assessed. The tree-specific parameters for decided learning rate can then be tuned accordingly. At a later step, after the tree-specific parameters are set, the learning rate can be reduced to assess for any improvements in accuracy.

Chapter 2

Porcine Reproductive and Respiratory Syndrome (PRRS) Outbreak Forecasting using Machine Learning

Abstract. Porcine Reproductive and Respiratory Syndrome (PRRS) is one of the most challenging and costly viral infectious diseases impacting the swine industry. The disease transmission pathways for PRRS are very complex, requiring a combined approach of intensive surveillance (i.e., testing), biosecurity, and vaccination for control and eradication. This study builds a proactive framework to forecast the risk of having a PRRS outbreak on a farm. This forecasting allows for early detection of disease outbreaks and could direct risk-based, and thus more cost-effective, interventions. Machine learning algorithms were trained using multi-scale data (pig group-, farm-, and area-level data). For the first time, on-farm, between-farm, and environmental variables, including farm location, pig movements, production parameters, diagnostic data, and climatic information, were combined for the prediction of PRRS outbreaks. Multi-scale datasets were merged via feature extraction, followed by the wrapper and filter feature selection, to find those feature subsets with the best forecasting performance. The predictive value of each features selection mechanism was evaluated in terms of its stability. Numerical results demonstrate good forecasting performance in terms of area under the ROC curve.

2.1 Introduction

The livestock industry capitalizes on the production of the highest quality animals through the most economically efficient means. The success of a given producer relies on their ability to maintain the health of their herds through good management practices, and the capacity to prevent, detect and control both endemic and epidemic diseases.

The US is the world's second largest pork producer and the second largest meat exporter (North American Meat Institute, 2016). Within the US, most pigs are raised within multi-site swine production systems (i.e. separate facilities by pig type and age), allowing for specialized housing and feed. However, this multisite system intrinsically requires the frequent movement of live animals between sites, providing a source of disease movement and introduction. Further, these intensive production systems create environments of high pig density, which increases the risk of disease spread. Porcine Reproductive and Respiratory Syndrome Virus (PRRSV) is currently the most challenging and costly viral infectious disease in the US swine industry, accounting for over $660M in losses annually (Holtkamp, Kliebenstein, et al. 2013). Among these outbreaks, 55% are associated with growing pigs and 45% with breeding farms. The high viral mutation rate seen in PRRSV results in high levels of sequence variability, making vaccine development and implementation a challenge (Mateu and Dıéaz 2008). The high cost of diagnostic screening tests, biosecurity (e.g., air filtration) and vaccination, as well as the direct losses associated with outbreaks, highlight the need to develop forecasting models to help identify farms at highest risk of having an outbreak. Such models allow more cost-effective and efficient disease mitigation efforts, with risk-based surveillance, vaccination and outbreak response strategies.

The current approach to PRRSV control includes the maintenance of high biosecurity, routine disease surveillance via diagnostic testing, and the use of standard vaccine protocols (Corzo et al. 2010). Serologic and molecular diagnostic tests are available for use on blood, oral fluids, and tissue samples from live and dead pigs (Nodelijk 2002). The shedding (using PCR) and exposure (using ELISA) status of a herd can be determined based on the results of these tests. For breeding herds (sow and nursery farms) there are four disease status categories: (I) positive unstable, (II) positive stable, (III) provisional negative and (IV) negative (Holtkamp, Polson, et al. 2011). Growing herds (finishing herds) are classified as either positive or negative status. The challenge is that untested farms have uncertain status and cannot be easily categorized as positive or negative. Some farm managers accept the risk of an outbreak rather than continuously running tests. Therefore, the level of diagnostic information, as well as the biosecurity and vaccination protocols, may vary by farm.

The aim of this study is to examine different machine learning models and to explore those variables or features that would most effectively forecast and enable the early detection of PRRSV outbreaks (Alkhamis et al. 2017; Shamsabardeh et al. 2019). This is a study based on multi-scale data (pig group-, farm-, area- level data). On-farm, between-farm, and environmental variables, including farm location, pig movements (**Valdes-Donoso et al.**; **2017**), production parameters, diagnostic data, and climatic information are evaluated. The ability to forecast high risk farms can inform strategies for more efficient testing and targeted mitigation plans to reduce the impact of PRRSV on the swine industry. This study focuses on finishing farms, which currently have the lowest frequency of disease screening and the lowest standards of immunization and biosecurity. Finishing farms could greatly benefit from a system that helps to forecast outbreaks. Importantly, improving health outcomes at finishing farms would contribute to reducing the burden of disease transmission to breeding herds, thus improving the health status of the entire system.

PRRSV transmission can occur by both direct and indirect contacts. The two main modes of PRRSV between-farm transmission are 1) the transportation of infected live pigs (Lee et al. 2017) and 2) airborne transmission from nearby infected farms (Otake et al. 2010). Other indirect routes of transmission include the use of infected semen, contaminated personnel, tools or materials, or insects which can act as mechanical vectors. In this study, we just considered the two main pathways for disease transmission: direct transmission through the reception of pigs from other farms, and indirect airborne transmission from nearby farms.

Different features are created to represent these disease pathways and other risk factors that may contribute to PRRSV epidemics. In general, adding additional features potentially increases the accuracy of a forecasting model. However, using a large number of features with comparably few data samples can result in overfitting to training data, and consequently, decreases the generalization of the model to new data samples (Guyon and Elisseeff 2003).

To combat this issue, feature selection methods are used. Feature selection is the process of selecting a subset of relevant features that are useful for predicting response variables. In this work, filter method feature selection based on correlation (Hall, 1999), and wrapper method based on recursive feature elimination (RFE) (Granitto et al. 2006; Guyon, Weston, et al. 2002; Haury, Gestraud, and Vert 2011), are used to find the most relevant features influencing PRRSV outbreaks. Furthermore, to compare the robustness of each feature selection algorithm with respect to different training data samples, stability analysis using Tanimoto distance is performed Kalousis, Prados, and Hilario 2007).

Overall, this work examines multiple machine learning models for outbreak forecasting and early detection in finishing farms using a combination of diagnostic, production, and pig trade data. This work demonstrates the strength of these techniques and provides the basis for future real-time dashboards that can allow producers to actively monitor and respond to shifting disease dynamics on their farms. In addition, an architecture composed of two forecasting models stacked to each other in order to exploit the data of the farms with unknown PRRS status is proposed.

2.2 Data and Feature Generation

In this section, the data source and structure, data pre-processing steps to build the features, and methods to forecast the probability of having a PRRSV outbreak are explained.

Data Sources

This study is conducted based on one large-scale swine production system with multiple sow, nursery, and finishing farms in the midwest of the United States.

For the time period 2006-2019, a rich database from this system provided information on the movement of pigs between farms, production of the farms, and PRRSV testing results. During this period, there were over 230,000 movement records to or from farms within the production system. For each movement entry, the source and destination, the farm type, the number of shipped pigs, the total weight of the shipped pigs, and the date of the movement are available.

At each finishing farm, the period of time from the first pig entering the farm to the last one leaving the farm is defined as Finishing Period (FP). Lab results demonstrate that 620 out of the 3770 FP during our study period experienced at least one outbreak. In practice, most of the farms are tested only when there is evidence of health problems on the farm. Thus the lab results are positive for almost all submitted samples and negative samples are not statistically representative of the negative class. To build a machine learning model that can classify negative and positive samples, samples for both classes are needed. In this study, domain knowledge expertise is used to define criteria for an assumed negative classification. A farm is assumed to be negative if it meets two conditions: the mortality rate is in the lowest 10 percent (i.e., in the 10th percentile), and the percentage of exiting pigs with weight in the standard range is in top 10 percent (i.e. at 90th percentile). This results in 5 percent of FPs being negative.

For each farm, climate information was obtained from the closest weather station. The data was obtained using the R package 'riem', which queries the data from an online interface to obtain weather information. The location of the weather station is not reported for data confidentiality. Temperature, wind speed, relative humidity and altitude are considered for this study.

Data Pre-Processing

Data Cleansing

Identification and correction of inaccurate data is an important step of data analysis. Using incomplete or inaccurate data samples in the training procedure may lead to poor model performance. Hence, the data were extensively analyzed to correct incomplete or inaccurate data samples. Some fields, such as weather information, were missing for several records. Missing fields were assigned the average value of the same period of time in the previous years. Weather related missing fields associated with each location (farm) are replaced with the average taken over previous years of the same period of time (e.g., month). . Due to discrepancies in the naming system in production data and lab results we could not associate some production data to the health status of the farm. Records with such inconsistencies were removed from the dataset. Moreover, inaccurate and invalid records were removed by applying a set of rational range constraints. Specifically, in some records the number of dead and survived pigs did not add up to the number of pigs entering the farm. Also, some weights of the pigs were out of the reasonable range for that type of a farm.

Feature Engineering

Machine learning can provide good predictions if it can extract the relevant information from the data. This means that its success depends on both the goodness of the model and the data representation, the transformation of the raw data into feature vectors. The better the data representation, the simpler the deployed model can be for the same performance metrics, meaning less chance of over-fitting and better generalization. Feature engineering is the manual construction of features from raw data. The importance of data representation and feature engineering becomes clearer when the number of data samples are small compared to the model complexity required to capture the relationship between dependent and independent parameters. The feature engineering step is the most time-intensive step of this work.

Domain knowledge is key to the construction of relevant features. One key contribution of this work is to construct features that represent those factors that affect the risk of having an outbreak on a farm. This paper combines data across different scales for better forecasting. After the construction of different features, feature selection methods can be used to evaluate whether a feature is improving the forecasting performance or not. For example, using temperature, different features were created based on the expectation that the spread of PRRSV would follow different patterns in warm (Dee, Deen, Rossow, Weise, et al. 2003) versus cold (Dee, Deen, Rossow, Wiese, et al. 2002) temperatures. Next, the best subset of temperature features can be chosen in the features selection step.

First Pathway (Direct Contact) Features: To model the PRRSV transmission through direct contacts, different risk factors were considered. Most pigs are able to clear PRRSV infection after getting infected, but some become persistently infected and can then act as carriers, spreading the virus if that pig is transferred to another farm. To capture this effect, a feature was created representing the number of entering pigs that are coming from a farm that has had an outbreak during the lifetime of that pig on the farm, i.e. if the nursery that the pig is coming from had an outbreak during that pig's lifetime. In addition, the total number of times that pigs enter a farm during a given FP, and the total number of different sources that pigs are coming from, are additional risk factors. The total number of pigs on a farm was also considered.

Second Pathway (Airborne) Features: The second pathway is through airborne transmission from nearby farms. To model this pathway, the vicinity of the farm was defined as the circular area around the farm within a defined distance. For each farm, the total number of movements and number of pigs, entering or exiting the vicinity were calculated. In addition, different neighborhood sizes (vicinity diameters) of 5km, 10km and 20km were examined in this study. Figure 1 depicts the neighborhood for farm F1 at the center of the circle. The dashed red arrows show the movements that the model counts for airborne effect. The solid red arrows represent direct contact movements.

Each movement feature has versions based on time period: 1) from the start of the FP up to its forecasting date, and 2) the historical equivalent of this feature for the one year prior to FP start date. These two sets of features are highly correlated, but together can indicate how the current FP movements are different from what is expected on average for the farm's neighborhood. Each of the features for current FP were normalized by dividing the feature value by the period of time for which they were calculated.

A farm with a higher density of neighboring farms is at higher risk for having an outbreak. More importantly, the number of outbreaks happening in the neighborhood of the farm during the FP, and the historical number of the outbreaks in the neighborhood in the one year prior to the start of FP, represent how risky the area is.

Production Features: Production data include total feed consumed and exiting weight at the end of the FP. This information cannot be used for the purpose of forecasting an outbreak for that same FP because it will violate causality. However, suchthese data can be used for the evaluation of future FPs, because it is a good indicator of the overall performance/management practices/risk of a farm. Thus, features for historical production data for each farm were built. The total number of pigs, and the average weight of pigs entering a farm, are two features that can be used for the current FP. The following features based on

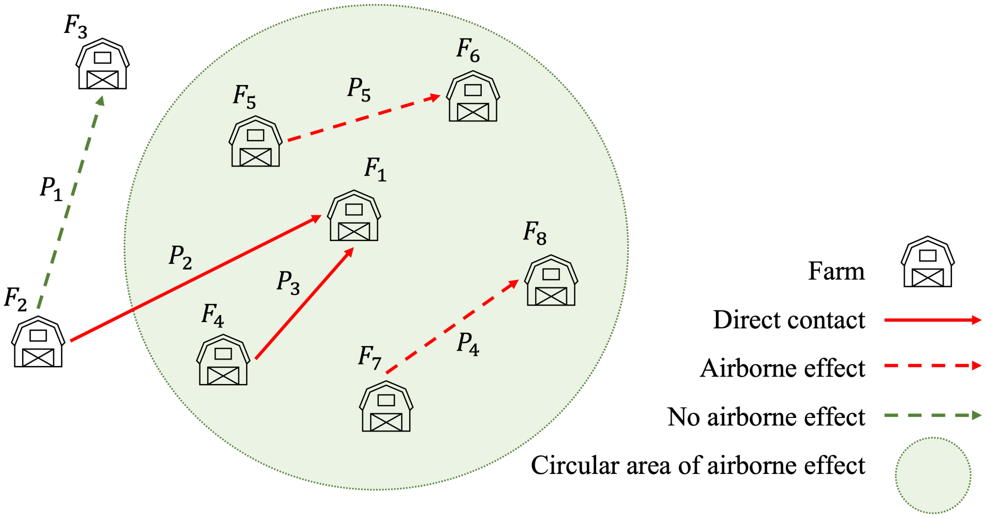


Figure 2.1: The two main PRRSV pathways for farm F3:1) Airborne effect from neighboring farms (movements (orange color) with source or destination to farms that are located in the circle), 2) direct reception of pigs (blue colors). Other movements (grey) and farms are assumed to have no effect.

historical data were used for each FP. Based on the weight of existing pigs, the percentage of sub-standard pigs was calculated by dividing the number of survivingsurvived pigs that were not within standard weight range by the total number of survived pigs. In addition, the average weight of sub-standard pigs on that farm was determined. Similarly, the percent and average weight of exiting pigs falling within the standard range were calculated. The total net weight survived is the weight difference of survived pigs from entrance to exit and is divided by the number of survived pigs to obtain average net weight survived. Total pig days is the number of days pigs spendspent on a farm. The total net weight survived can be divided by the total pig days to obtain Average Daily Gain (ADG). Similarly, the Average Daily Feed (ADF) was calculated as the ratio of total consumed feed and total pig days. Next, the ratio between ADG and ADF provides the net weight survived per one pound of food consumed. From the information on the number of days on-farm the following features were calculated for dead and survived pigs: the total live pig days is the number of days that survived pigs were alive; total dead pig days is the total number of days that pigs were on-farm before their death. The total live pig days and total dead pig days were divided by the number of survived pigs and the number of dead pigs, respectively. The mortality rate is the ratio of the number of dead pigs to entering pigs. All these are used as features. Other Overall Management Practices/Performance Features Additional features that can demonstrate the general management practices of a farm were created. Good management practices include the disinfection of the farm before the start of each FP. Additionally, receiving new animals all together within a few days of the start of the FP (i.e. all-in all-out), is considered a better practice than to allow continual additions throughout the FP. The continuous reception of pigs, versus a single time point population of the farm, results in a staggered cycle of animals leaving the farm at different time points, meaning there is no time at which the entire farm can be disinfected, allowing for the possible retention of infection from previous FPs. To represent this risk factor, a feature was created for the number of days between the first and last reception of new pigs. The percentage of time that a farm has had an outbreak in the past is a mixed indicator of all of the above mentioned historical factors. Climate Features In addition to wind speed, relative humidity, and altitude provided in the climate data, the average and lowest temperature of the past 15, 30, 45, 60 and 90 days prior to the forecasting date were built. It was not deemed necessary to have exact temperature measurements for every time point given the expected variability between true on-farm temperature values versus those recorded by weather stations due to on-site thermoregulation, distance from the weather station, and data collection in windows of a minimum of 15 days. Missing weather data were removed, and the average temperature for the past 60 days was selected as the best temperature feature for the model. The season of the FP was defined as the season of the forecasting day. It is a categorical feature and was represented using one hot encoding. In other words, four binary features were created, and a single value was assigned to a given FP based on the corresponding season for a given forecasting day.

2.3 Machine Learning Methods

In this section, the machine learning methods that were used to forecast the PRRSV outbreak probability are explained. As in the other classification problems, the goal was to find the discriminator function that most efficiently mapped features to target labels.

Standard Machine Learning Models

Various machine learning algorithms including: Logistic Regression (LR), Support Vector Machine (SVM), Gradient Boosting (GB), and Random Forest (RF) were trained to forecast the PRRSV status of a farm. The probability of each farm being classified as positive for PRRSV is obtained from the output of each of these models. A farm is identified as positive by the model if its probability of infection is higher than a given threshold. Thus, metrics such as accuracy, sensitivity and specificity are dependent on this threshold. The Receiver Operating Characteristic (ROC) does not have this issue as it can be computed for every possible threshold. ROC curve shows how true positive rate (sensitivity) changes with false positive rate (1- specificity) for different thresholds. Therefore, the Area Under Curve (AUC) of the ROC is a good metric for comparing different models and is used here.

Cross Validation and Hyper-parameter Tuning

Given the past and the present observations on the health status of a farm and its features, the goal of this work is to predict the future health status of the farm. Therefore, all data about events that occur chronologically after the time of forecasting should be withheld and not used for prediction. The data were therefore split temporally into two non-overlapping parts for training and testing. The first portion was used for model training, while the second half was reserved for performance evaluation of the model.

Each model has different hyper-parameters that govern its complexity. We tuned hyper-parameters, namely, learning rate, tree depth, etc., to find the best fit model, i.e., to prevent the model from both overfitting and underfitting. To achieve this, the training data were further divided temporally into two chunks. The first chunk included 80 percent of the training data and was used to train the models with different sets of hyper-parameters; the remaining 20 percent of training data was used to test these models and find the best hyper-parameters values. First, hyperparameter-tuning was performed using the training data by performing a grid search over a range of values for the hyperparameters in different types of models. Specifically, each of these models was trained using eighty percent of the training data, and then tested over the remaining twenty percent for each collection of values of the hyperparameters to find the best hyper-parameters values. Using these best hyper-parameters values, a performance evaluation was then carried out by repeatedly training the model using a randomly selected subset (60%) of the training set and reporting the area under the receiver operator characteristic curve (AUC-ROC) on a randomly selected subset (80%) of the test set. Note that this approach for model validation was chosen so that all the points used for testing come chronologically after the ones used in training as the standard K-fold cross validation is not appropriate for time series data.

Feature Selection

The process of selecting features with the highest contribution towards forecasting the output is called feature selection. Having a high-dimensional feature space can cause the training algorithm to have impaired learning performance, be prone to overfitting, and become computationally cumbersome. The main goal of feature selection is either: 1) to find the subset of features that minimizes generalization error, or 2) to select the smallest possible subset of features that satisfies the performance criterion and allows for better model interpretability. The main approach in this work is the latter.

Prior to the feature selection, we used hierarchical clustering to observe the degree to which the features are correlated. Hierarchical clustering groups the features such that the features constituting one group have more similarity among themselves than features in the other groups. Feature selection methods are categorized into wrapper, filter and embedded methods. In this study, we used filter and wrapper methods as explained below:

Filter Methods: The filter method performs a feature selection procedure regardless of the type of learning model. A scoring measure based on data characteristics such as distance, information, or correlation, is used as the metric to filter those features that seem more relevant to the response variable. The filtering in this work was done based on Pearson correlation and mutual information.

Pearson correlation measures the similarity between two variables. In a univariate method, the correlation between each feature and the response variable is obtained, and feature selection is done based on the correlation with the response variable (target). Another popular filter method is a mutual information-based feature selection Hoque, Bhattacharyya, and

Kalita 2014 which uses mutual information as the entropy measure to choose the subset of important features. Mutual information is a measure between two random variables, e.g., each input feature and the response variable, that quantifies the amount of information obtained about one, through the other. The drawback of these two approaches is that they do not take the correlation between features into account (only that between the feature and response variable), and may thereby choose two highly correlated features such that one is redundant in presence of the other. To solve this problem, we proposed a metric based on Pearson correlation and mutual information, described in Algorithm 1, to find the desired subset of features. Specifically, the metric denoted as M is directly proportional to the mutual information MI and to the correlation with target P_{Target} . It is, however, reversely proportional to the correlation with the previously selected features, as we want to avoid selecting highly correlated features and choose features that can contribute different information to the classifier accuracy. We convert the proportionality to equality as:

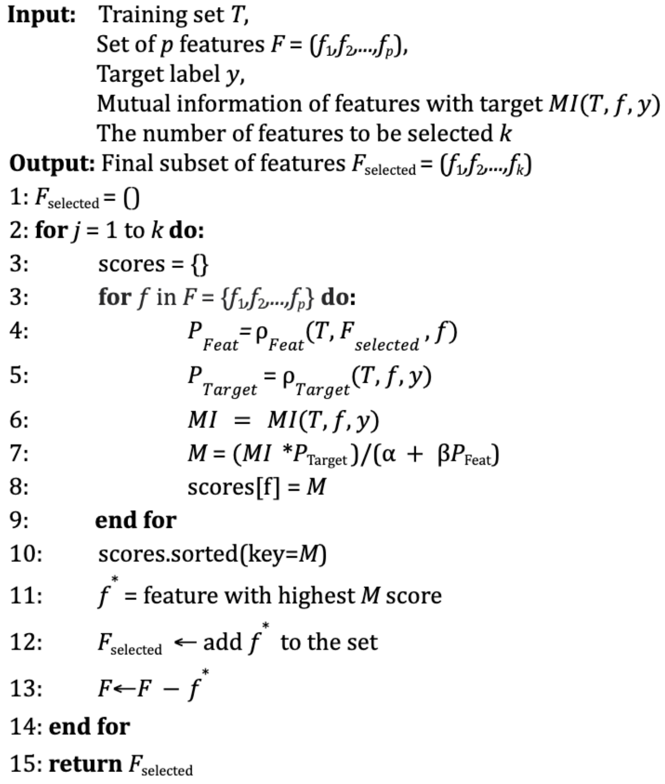
$$M = \frac{MI \times P_{\text{Target}}}{\alpha + \beta P_{\text{Feat}}}, \tag{2.1}$$

where α and β are hyperparameters that control the dependency of the metric M to feature correlation P_{Feat} and correlation with response variable P_{Target}. High ratios of α/β will eliminate the dependency of M on P_{Feat} and high values of α removes the dependency of the M on P_Target. We performed a grid-search over the range of values 1, 10 and 100 for both and to determine their optimal value.

The Pearson's correlation function $\rho_{\text{Feat}}(T, F_{\text{selected}}, f)$ takes the training set T computes the Pearson's correlation of feature f with all the features in set $F_{selected}$ and returns the maximum within feature correlation over features in set $F_{selected}$. The function $F_{\text{Target}}(T, f, y)$ computes the Pearson's correlation of feature f with target y. Finally, the function $MI(T, f, y)$ take the training set T and computes the Mutual information of feature f with the target y.

Wrapper Methods: Wrapper methods merge feature selection and learning steps allowing the learning algorithm to interact with the bias of the feature selection step, decreasing the total bias. Thus, using wrapper methods, a subset of features that result in better prediction performance will be selected. The Recursive Feature Elimination (RFE) method (Guyon,

Algorithm 1 Algorithm for proposed filter method.



```
Input:  Training set $T$,
        Set of $p$ features $F = (f_1, f_2, ..., f_p)$,
        Target label $y$,
        Mutual information of features with target $MI(T, f, y)$
        The number of features to be selected $k$
Output: Final subset of features $F_{selected} = (f_1, f_2, ..., f_k)$
$F_{selected} = ()$
for $j = 1$ to $k$ do:
    scores = {}
    for $f$ in $F = \{f_1, f_2, ..., f_p\}$ do:
        $P_{Feat} = \rho_{Feat}(T, F_{selected}, f)$
        $P_{Target} = \rho_{Target}(T, f, y)$
        $MI = MI(T, f, y)$
        $M = (MI * P_{Target})/(\alpha + \beta P_{Feat})$
        scores[f] = $M$
    end for
    scores.sorted(key=$M$)
    $f^*$ = feature with highest $M$ score
    $F_{selected} \leftarrow$ add $f^*$ to the set
    $F \leftarrow F - f^*$
end for
return $F_{selected}$
```

Weston, et al. 2002) is a commonly used wrapper model. It is a recursive algorithm that ranks features according to some measure of their importance. For example, SVM-RFE ranks the features based on SVM, Sanz et al. 2018. In this paper, we use LR-RFE, SVM-RFE, GB-RFE and RF-RFE to eliminate and rank features.

Stability of Feature Selection

Many feature selection algorithms have been successful at improving the forecasting accuracy of learning models while reducing feature-space dimensionality and model complexity (Khalid, Khalil, and Nasreen 2014). Beyond high accuracy, the stability of feature selection is another important attribute of these algorithms. The stability of a feature selection algorithm is defined as the robustness of the feature set it produces to differences in training sets drawn from the same generating distribution $P(X, C)$, where C is the class label for X. Here we use the stability measure proposed in Kalousis, Prados, and Hilario 2007 to compare several feature selection methods, informing selection of the one that best fits our dataset and performance needs. Similarity between two subsets of features using a straightforward Ss takes values in [0,1] with 0 meaning there is no overlap between the two sets and 1 that the two sets are identical. To empirically estimate the stability of a feature selection algorithm for a given dataset, the distribution P(X,C) from which the training sets are drawn is simulated by using a re-sampling technique

$$S_s(s, s') = 1 - \frac{|s| + |s'| - 2|s \cap s'|}{|s| + |s'| - |s \cap s'|}, \tag{2.2}$$

where S_s takes values in $[0, 1]$ with 0 meaning there is no overlap between the two sets, and 1 that the two sets are identical. To calculate the stability, K data subsets were created by randomly shuffling the data and dividing it into folds. A small K does not produce a robust estimation of the variance for stability estimation as there are few instances of its measurement. A large K, on the other hand, decreases the number of data points in each fold, and as a result does not yield a reliable AUC-ROC score. Therefore, as a compromise between accuracy for AUC-ROC and stability, we chose $K = 5$ folds. For each fold, the selected features are computed according to the feature selection method. Then, the similarity of each pair of selected features, i.e. $K(K - 1)/2$ pairs, is computed using the similarity measure.

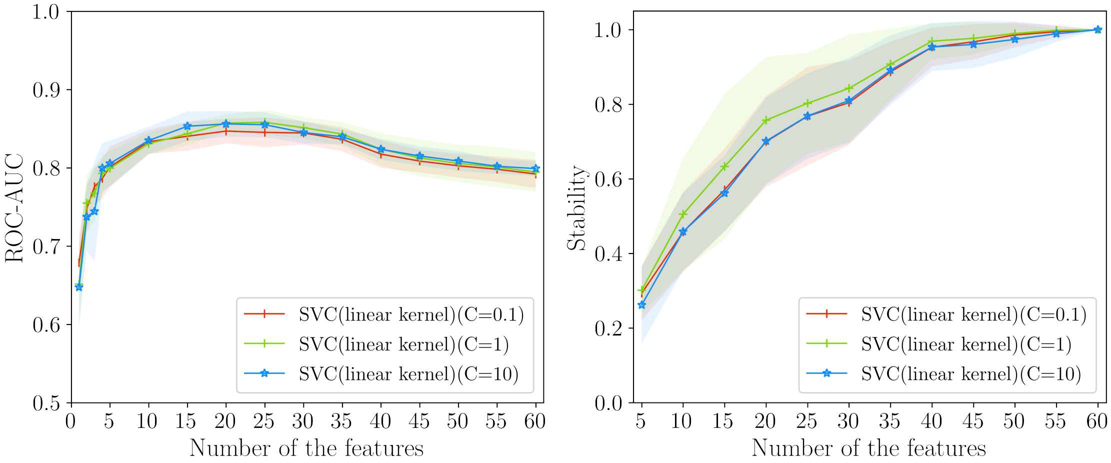
2.4 Result

In this section, the performance analysis, in terms of test ROC-AUC and stability of the feature selection, regarding the four predictive models on the extracted features are presented.

Performance and Stability Results

We used 196 data points for the method training and evaluation. Specifically, the training set consists of 157 data points (80%), and the remaining 39 data points (20%) constitute the testing set. A performance evaluation was conducted by repeatedly training the model using 94 randomly selected data points (60% of the training set) and reporting the AUC-ROC on 31 randomly selected data points (80% of the test set).

The various hyper-parameters for each model, alongside the related AUC-ROC score for N=10, 20, 40 features, are presented in Table 1. The best hyper-parameter that achieves higher AUC-ROC score, while introducing less complexity in terms of the number of parameters is indicated with bold font. An example of hyper-parameter tuning for support vector classifier for features length ranging from 1 to 60 can be seen in Figure 2. A range of three values for the regularization parameter (C) were considered, where the strength of the regularization is inversely proportional to C.



(a) AUC-ROC score versus number of features

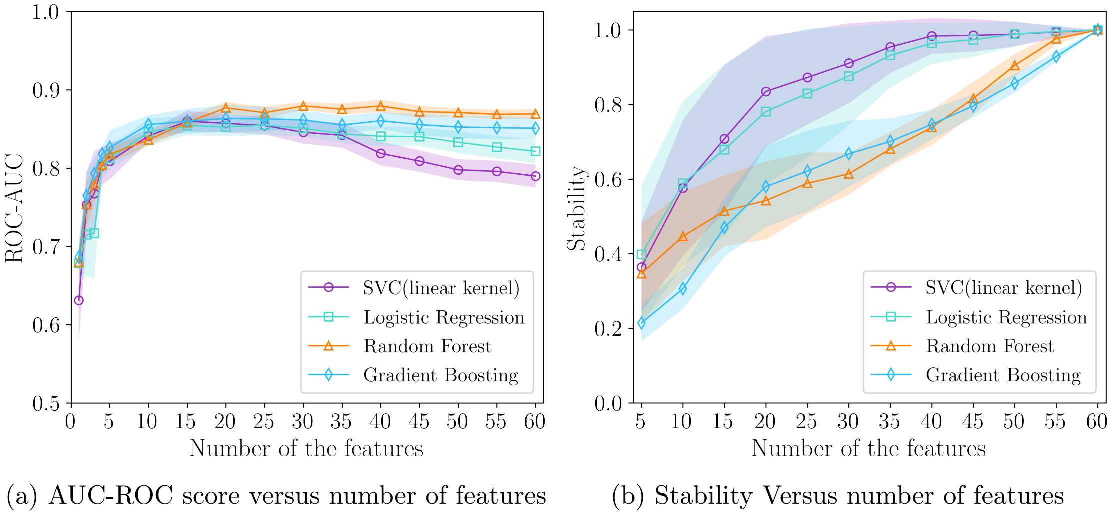
(b) Stability Versus number of features

Figure 2.2: (a) Stability score for different regularization parameters (C) in Support Vector Classifier (SVC) for different sizes of feature sets. (b) the area under the ROC curve (AUC-ROC) score for different regularization parameters for different sizes of feature set. The strength of the regularization is inversely proportional to the regularization parameter C. The regularization C=10 yields the best AUC-ROC but the regularization C=1 has better stability (equivalent to part of the table 1).

For each model, Figure 3, demonstrates the AUC-ROC and stability measure mean and standard deviation across the folds. This figure shows the performance of each of these models together with the stability of the corresponding RFE-based feature selection method in terms of AUC-ROC score. According to the results, the two non-linear models, GB and RF, have better AUC-ROC scores than the linear models, SVC and LR (Figure 3a). In Figure 3b, it can be seen that the non-linear models, GB and RF, are not as stable as the linear models in selecting a robust subset of features.

Models	Parameters		AUC-ROC		
	Kernel	C	N = 10	N=20	N=40
SVC	linear	0.1	0.82±0.014	0.85±0.014	0.81±0.016
	linear	1	0.83±0.014	0.86±0.014	0.81±0.018
	linear	**10**	**0.85±0.014**	**0.86±0.012**	**0.83±0.013**
	Penalty	C	N=10	N=20	N=40
LR	L1	0.1	0.84±0.010	0.84±0.011	0.82±0.010
	L1	1	0.86±0.014	0.85±0.011	0.83±0.013
	L1	10	0.84±0.013	0.85±0.014	0.82±0.012
	L2	**0.1**	**0.86±0.007**	**0.87±0.009**	**0.86±0.010**
	L2	1	0.83±0.021	0.86±0.014	0.85±0.013
	L2	10	0.83±0.019	0.84±0.019	0.84±0.014
	Estimators	Maximum depth	N=10	N=20	N=40
RF	50	4	0.82±0.017	0.86±0.010	0.87±0.009
	50	8	0.82±0.010	0.86±0.008	0.86±0.010
	50	16	0.82±0.020	0.86±0.019	0.86±0.019
	50	32	0.83±0.017	0.87±0.019	0.87±0.008
	100	4	0.82±0.013	0.86±0.012	0.86±0.010
	100	**8**	**0.83±0.018**	**0.87±0.013**	**0.88±0.003**
	100	16	0.83±0.012	0.87±0.006	0.88±0.015
	100	32	0.83±0.014	0.87±0.011	0.88±0.013
	Learning rate	Maximum depth	N=10	N=20	N=40
GB	0.01	3	0.84±0.016	0.84±0.014	0.84±0.012
	0.01	5	0.86±0.014	0.85±0.011	0.83±0.013
	0.1	**3**	**0.85±0.021**	**0.86±0.019**	**0.86±0.020**
	0.1	5	0.84±0.014	0.86±0.011	0.86±0.013

Table 2.1: Hyper-parameter tuning for the four classifiers used for PRRSV outbreak prediction. The ROC-AUC for each set of hyper-parameters is reported for different feature sizes N = 10, 20, 40, and the best set is shown in bold font.



(a) AUC-ROC score versus number of features

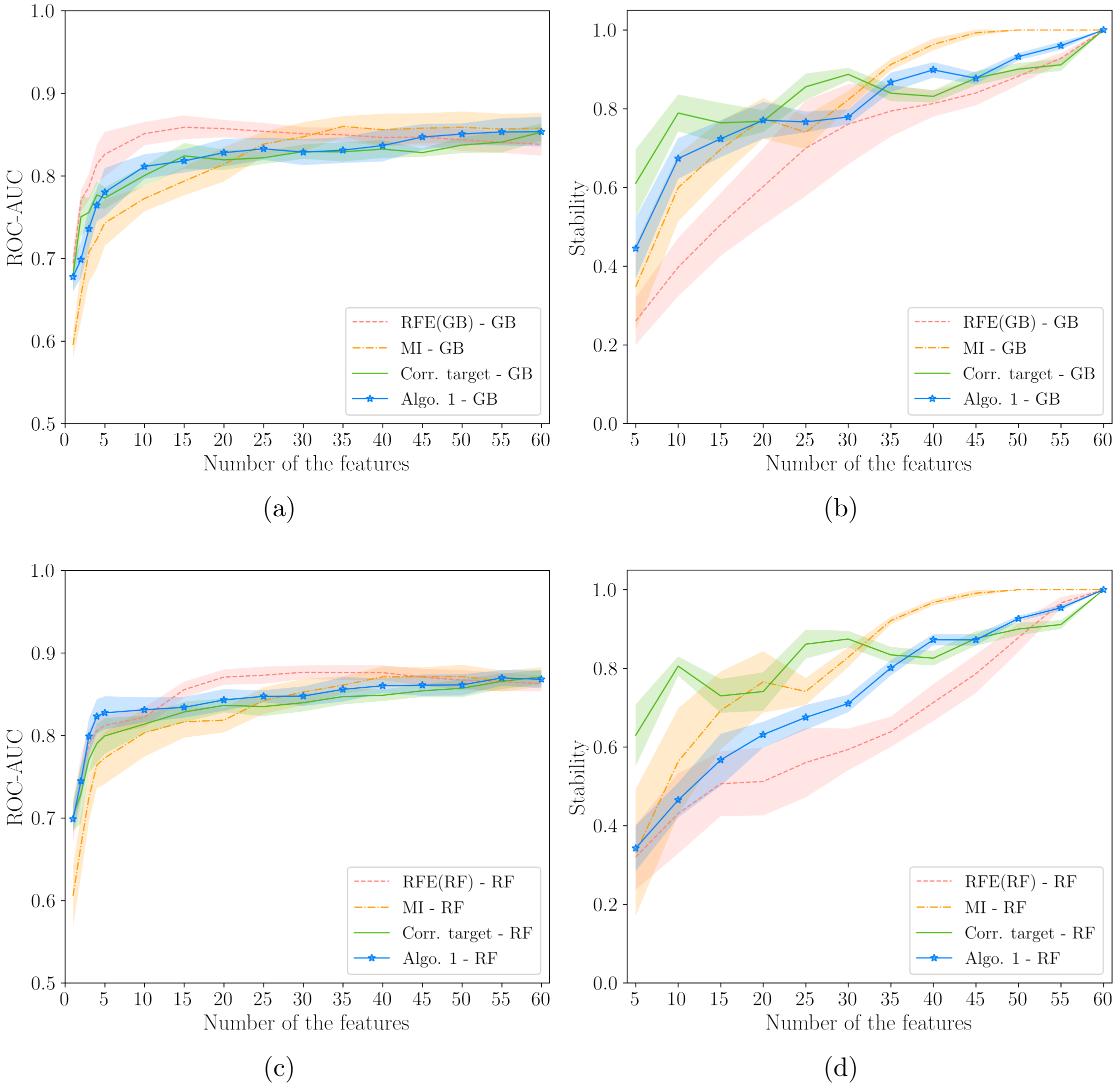
(b) Stability Versus number of features

Figure 2.3: Performance assessment of Recursive Feature Elimination (RFE) merged with different classifiers for PRRSV outbreak prediction in terms of: (a) The area under the ROC curve (AUC-ROC), and (b) stability score for different size features sets. Each graph is labeled according to the classifier used in the RFE feature selection algorithm.

Since the tree-based models (GB and RF) have a higher AUC-ROC, we used them as the base classifiers to assess other filter-based feature selection methods (see Figure 4). In Figure 4a-b, we show the performance assessment of different feature selection methods using GB as the classifier. Specifically, we compared a RFE-GB feature selection method and three other filter-based feature selection methods: correlation with target, Mutual Information (MI), and our proposed algorithm (Algorithm 1). Similarly, in Figure 4c-d, we showed the performance assessment of these feature selection methods using RF as the classifier. As demonstrated in these two figures, the filter-based feature selection methods had higher stability, but lower ROC-AUC, in comparison with the RFE-based methods. Algorithm 1 surpassed the stability of RFE-based feature selection methods, while showing a comparable ROC-AUC performance.

Feature Selection

To identify highly correlated features, hierarchical clustering was performed as shown in Figure 5. The features are clustered according to their correlation-based distance and the cluster value, shown as a distinct color in the horizontal bar plot. Each feature belonging to



(a)

(b)

(c)

(d)

Figure 2.4: Performance demonstration of different feature selection methods (wrapper and filter methods) merged with the Gradient Boosting (GB) Random Forest (RF) as classifier for PRRSV outbreak prediction in terms of: (a and c) The area under the ROC curve (AUC-ROC), and (b and d) Stability score for different size features sets. Each graph is labeled according to its feature selection method (on the left of dash) and its classifier method (on the right of dash).

a specific cluster is representative of that cluster and has relatively the same contribution in terms of classification performance.

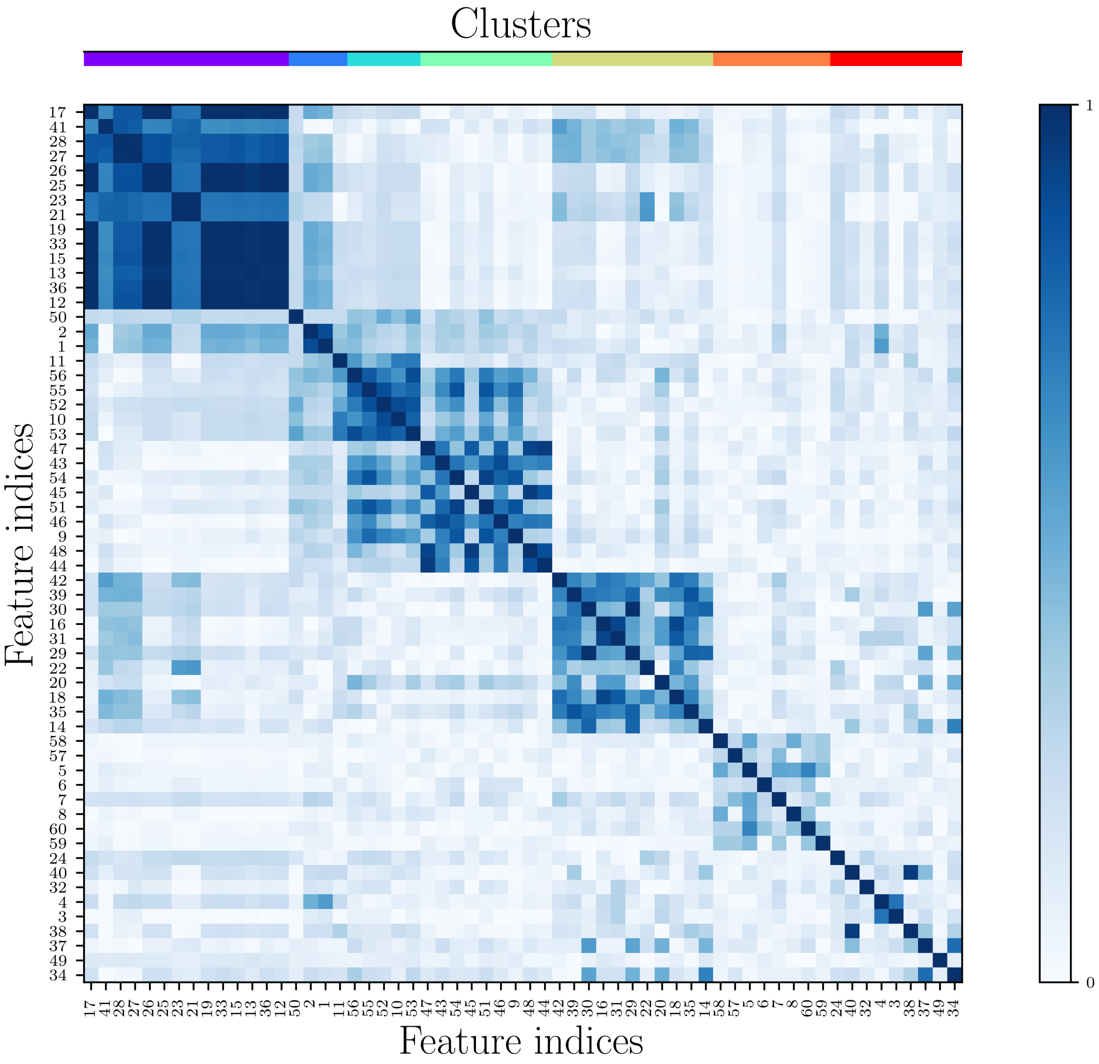
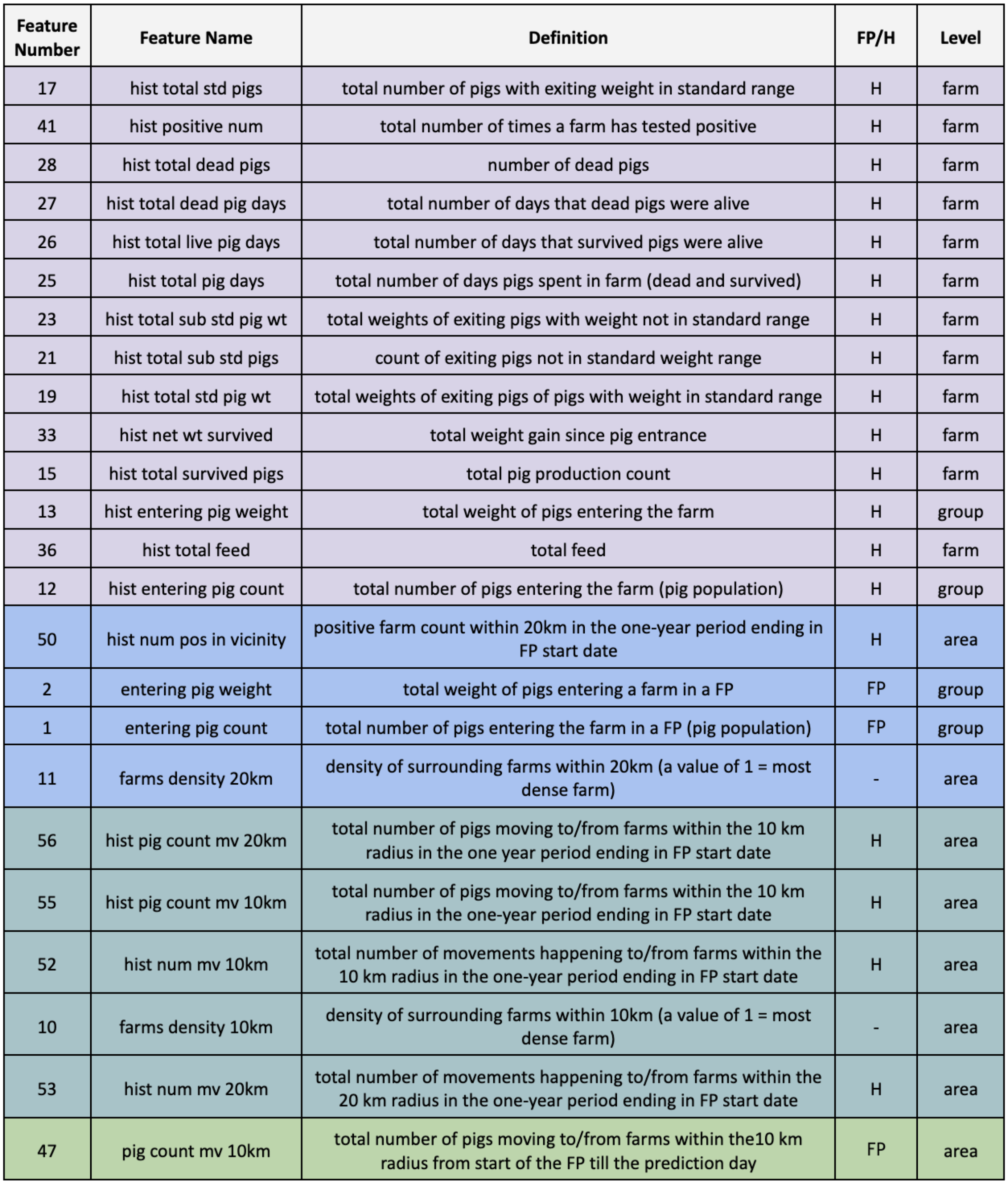


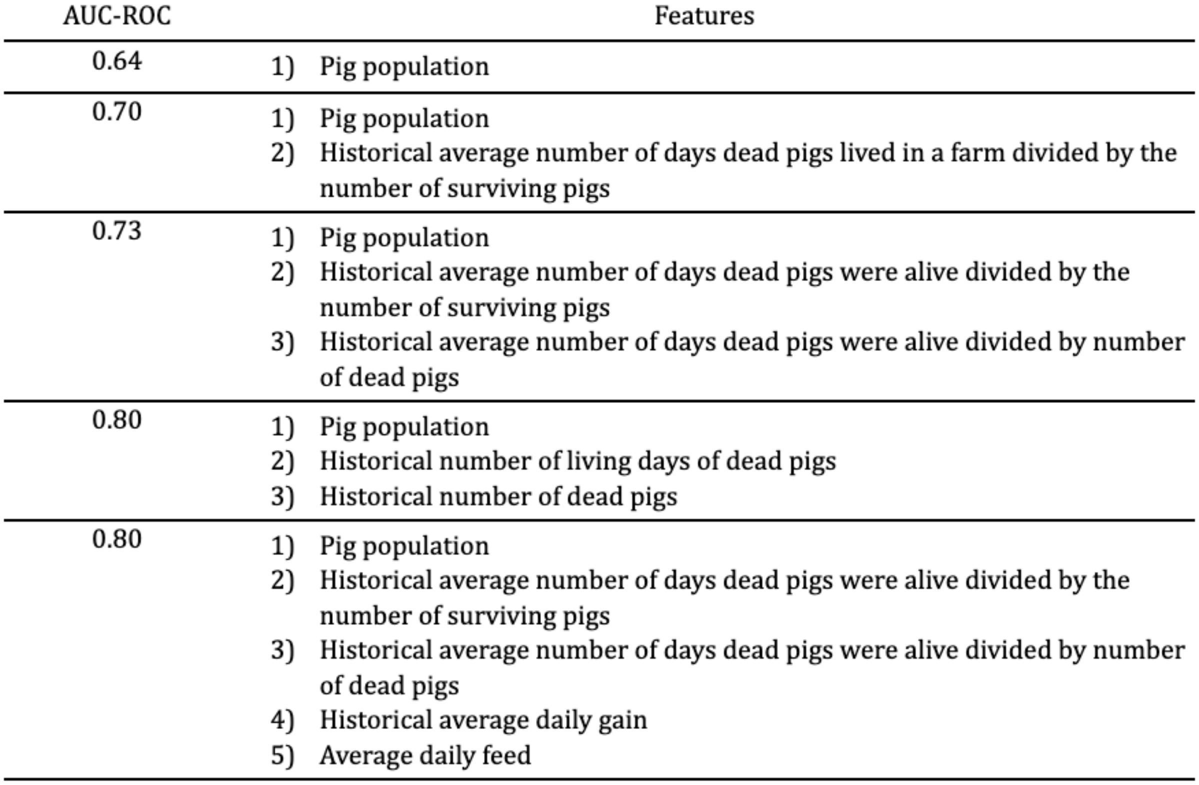
Figure 2.5: Feature similarity grouped by hierarchical clustering. Hierarchical clustering is used to analyze the similarity between features in terms of their correlation. The darker colors are representative of higher correlations among features. The formed blocks are indicative of the clusters of similar features. The exact margins are shown by the horizontal bar at the top where each color represents a cluster. The description of each feature number can be found in Table 1 of Supplement 1.

According to these sampled selected feature subsets, it was found that the average 60 prior days was the best temperature feature when combined with other features for forecasting, thus these values were used in the model. The prior 60 day average temperature and wind



Feature Number	Feature Name	Definition	FP/H	Level
17	hist total std pigs	total number of pigs with exiting weight in standard range	H	farm
41	hist positive num	total number of times a farm has tested positive	H	farm
28	hist total dead pigs	number of dead pigs	H	farm
27	hist total dead pig days	total number of days that dead pigs were alive	H	farm
26	hist total live pig days	total number of days that survived pigs were alive	H	farm
25	hist total pig days	total number of days pigs spent in farm (dead and survived)	H	farm
23	hist total sub std pig wt	total weights of exiting pigs with weight not in standard range	H	farm
21	hist total sub std pigs	count of exiting pigs not in standard weight range	H	farm
19	hist total std pig wt	total weights of exiting pigs of pigs with weight in standard range	H	farm
33	hist net wt survived	total weight gain since pig entrance	H	farm
15	hist total survived pigs	total pig production count	H	farm
13	hist entering pig weight	total weight of pigs entering the farm	H	group
36	hist total feed	total feed	H	farm
12	hist entering pig count	total number of pigs entering the farm (pig population)	H	group
50	hist num pos in vicinity	positive farm count within 20km in the one-year period ending in FP start date	H	area
2	entering pig weight	total weight of pigs entering a farm in a FP	FP	group
1	entering pig count	total number of pigs entering the farm in a FP (pig population)	FP	group
11	farms density 20km	density of surrounding farms within 20km (a value of 1 = most dense farm)	-	area
56	hist pig count mv 20km	total number of pigs moving to/from farms within the 10 km radius in the one year period ending in FP start date	H	area
55	hist pig count mv 10km	total number of pigs moving to/from farms within the 10 km radius in the one-year period ending in FP start date	H	area
52	hist num mv 10km	total number of movements happening to/from farms within the 10 km radius in the one-year period ending in FP start date	H	area
10	farms density 10km	density of surrounding farms within 10km (a value of 1 = most dense farm)	-	area
53	hist num mv 20km	total number of movements happening to/from farms within the 20 km radius in the one-year period ending in FP start date	H	area
47	pig count mv 10km	total number of pigs moving to/from farms within the10 km radius from start of the FP till the prediction day	FP	area

were selected more frequently than relative humidity and altitude. Seasonal features were not selected when a temperature feature was chosen. In addition, it was found that the number of movements in the neighborhood for different radii (the number of movements to/from any farm located in 5km, 10km and 20 km) can be used together to improve prediction.



AUC-ROC	Features
0.64	1) Pig population
0.70	1) Pig population 2) Historical average number of days dead pigs lived in a farm divided by the number of surviving pigs
0.73	1) Pig population 2) Historical average number of days dead pigs were alive divided by the number of surviving pigs 3) Historical average number of days dead pigs were alive divided by number of dead pigs
0.80	1) Pig population 2) Historical number of living days of dead pigs 3) Historical number of dead pigs
0.80	1) Pig population 2) Historical average number of days dead pigs were alive divided by the number of surviving pigs 3) Historical average number of days dead pigs were alive divided by number of dead pigs 4) Historical average daily gain 5) Average daily feed

Table 2.2: Subsets of selected features using Recursive Feature Elimination + Gradient boosting classifier to forecast PRRSV outbreaks as evaluated by AUC-ROC. Historical is defined as a period of one year prior to the start of the current finishing period.

The 20km radius features were more frequently selected than those of the 5km and 10km vicinity. In general, the number of movements were more important than the number of animals being shipped in a given movement. The total number and weight of incoming pigs, and the percent of existing pigs with substandard weight were important. The number of dead pigs and the average number of days that dead pigs have lived on the farm are also important predictive features. Moreover, average daily feed, past outbreak frequency in the farm, and the number of outbreaks in the neighborhood during the current finishing period, were amongst the most important features.

2.5 Discussion

This study incorporates swine farm- and area-level data in the forecasting of the farm-level PRRSV outbreaks. Using a uniquely rich real-world dataset, obtained from our industry collaborators, we included a level of detail that, to the best of our knowledge, has not been previously considered in the prediction of PRRSV outbreaks. We integrated production data, movement data, and climate information for our predictions. Further, we demonstrate the generation of new features from standard industry variables that better represent farm-level management practices and risk for use in forecasting models. In this manner, we have addressed the two main PRRSV transmission pathways, direct contact and airborne, as well as onsite disease history and management practices, and the role of near-farm status, on outbreak risk.

Based on the AUC-ROC and stability results, the two tree-based models have superior AUC-ROC scores in comparison with the linear models when used as classifiers. This is expected as the tree-based models are non-linear in nature and, therefore, can capture the nonlinearities in the data. However, due to their inherent randomness, they do not show a reasonable stability when used as RFE-based feature selection methods. As observed in the results section, filter-based feature selection methods demonstrated considerable stability and, hence, were used as the basis for developing a new feature selection algorithm. By combining Algorithm 1 for feature selection, which inherits the superior stability of the filter-based feature selection, and a tree-based model as classifier, we achieve high predictive performance and stability.

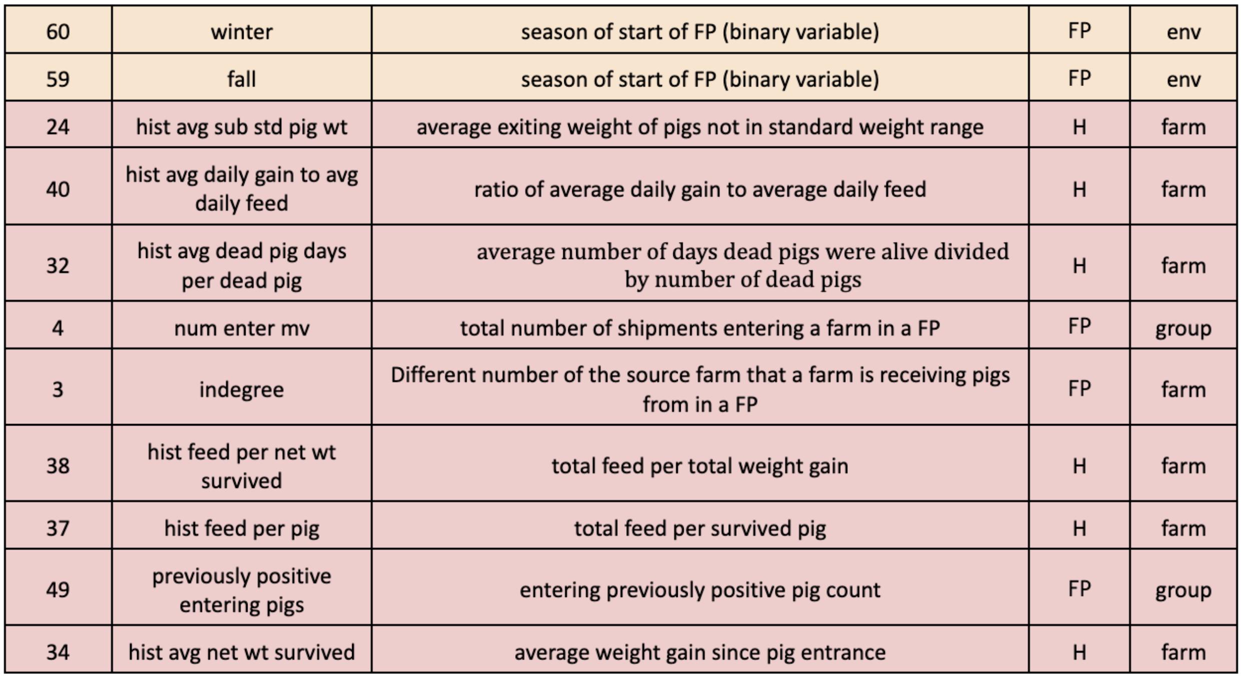
Considering Table 2 and 3, we observe that all different types of data, i.e., shipment, diagnostic, and production, play an important role in improving the prediction performance of the model. Based on the feature selection results, the most predictive feature in the dataset is the pig population. As the most frequently selected features across different methods, features representing movements in a neighborhood during the current FP are strong predictive features. Features related to pig movements to/from farms located in a 20 km neighborhood is a strong predictive feature to capture the effect of airborne disease transmission pathway. Features representing the number of dead pigs and the number of days they lived can represent the magnitude and impact of the PRRSV infection (and associated co-morbidities) on the farm. The historical features, which are the averages over the past measurements, are important because they provide the model with the information about the biosecurity and management practices of the farm over the time, while the current FP features are informative in terms of the recent events. The superior performance of the 60-day climatic period may be due to the fact that it capture seasonality; or, it may outperform other time periods because more shipments happen during the 60 days prior to the forecasting day and thus this average may best represent the temperature that pigs were exposed to during shipment.

Different subsets or combinations of features can yield the same performance. This is due to the fact that a feature in a given subset may be substituted with another feature that is highly correlated with it. This gives flexibility to those wanting to generate their own models. The clusters presented in Figure. 5, and on the list and description of features included in each cluster (Supplement 1), provide alternative features for use in forecasting when all the data fields used in this work are not available. In general, selecting 4 features, one from each of the first 4 clusters in Figure. 5 should provide high predictive power.

2.6 Conclusion

To the best of our knowledge, this is one of the first attempts to apply multiple machine learning models for PRRSV forecasting using multi-level data. We have demonstrated the strength of these methods for disease prediction in the swine industry, and believe they could be readily adapted for use on other diseases and for additional livestock species. This approach could save the swine industry millions of dollars through the improved efficiency and reduced economic burden provided by early, targeted, risk mitigation strategies. This work uses a rich, multi-scale (pig group-, farm-, area-level data) feature set - assessing on-farm characteristics at a level previously unreported for farm health analysis in the swine industry. Additionally, the integration of historical data with current cycle data to improve forecasting accuracy is a novel approach applied in this work. Generating an expansive set of features, ranging across farm level, time and space, allowing the evaluation of multiple disease transmission pathways, environmental factors, and management practices as risk factors for disease occurrence, resulted in improved outbreak forecasting ability. Stable feature selection allowed us to identify and represent the most important risk factors for PRRSV outbreaks. These variables can now be further explored by the industry and research community as points for future intervention. These approaches offer a strong basis for ongoing work, and we hope the adaptation of these methods into dynamic dashboards within the Disease BioPortal (https://bioportal.ucdavis.edu) will provide industry users with near real-time information for improved health management decisions.

43	num mv 5km	total number of movements happening to/from farms within the 5 km radius from start of the FP till the prediction day	FP	area
54	hist pig count mv 5km	total number of pigs moving to/from farms within the 5 km radius in the one-year period ending in FP start date	H	area
45	num mv 20km	total number of movements happening to/from farms within the 20 km radius from start of the FP till the prediction day	FP	area
51	hist num mv 5km	total number of movements happening to/from farms within the 5 km radius in the one-year period ending in FP start date	H	area
46	pig count mv 5km	total number of pigs moving to/from farms within the 5 km radius from start of the FP till the prediction day	FP	area
9	farms density 5km	density of surrounding farms within 5km (a value of 1 = most dense farm)	-	area
48	pig count mv 20km	total number of pigs moving to/from farms within the 20 km radius from start of the FP till the prediction day	FP	area
44	num mv 10km	total number of movements happening to/from farms within the 10 km radius from start of the FP till the prediction day	FP	area
42	hist positive percent	percentage of times a farm has tested positive	H	farm
39	hist avg daily feed	average daily feed	H	farm
30	hist avg live pig days per survived pig	ratio of total live pig days per survived pig	H	farm
16	hist percent survived pigs	percentage of survived pigs	H	farm
31	hist avg dead pig days per survived pig	ratio of total dead pig days per survived pig	H	farm
29	hist avg pig days	ratio of the total pig days per survived pig	H	farm
22	hist percent sub std pigs	percentage of exiting pigs not in standard weight range	H	farm
20	hist avg std pig wt	average exiting weight of pigs in standard weight range	H	farm
18	hist percent std pigs	percentage of pigs with exiting weight in standard range	H	farm
35	hist average daily gain	average daily gain	H	farm
14	hist avg entering pigs wt	average weight per pig of total entering pigs	H	group
58	summer	season of start of FP (binary variable)	FP	env
57	spring	season of start of FP (binary variable)	FP	env
5	avg temperature	average temperature in Fahrenheit	H	env
6	relative humidity	average of the relative humidity	H	env
7	wind speed	average wind speed in knot	H	env
8	altitude	farm altitude	-	env



60	winter	season of start of FP (binary variable)	FP	env
59	fall	season of start of FP (binary variable)	FP	env
24	hist avg sub std pig wt	average exiting weight of pigs not in standard weight range	H	farm
40	hist avg daily gain to avg daily feed	ratio of average daily gain to average daily feed	H	farm
32	hist avg dead pig days per dead pig	average number of days dead pigs were alive divided by number of dead pigs	H	farm
4	num enter mv	total number of shipments entering a farm in a FP	FP	group
3	indegree	Different number of the source farm that a farm is receiving pigs from in a FP	FP	farm
38	hist feed per net wt survived	total feed per total weight gain	H	farm
37	hist feed per pig	total feed per survived pig	H	farm
49	previously positive entering pigs	entering previously positive pig count	FP	group
34	hist avg net wt survived	average weight gain since pig entrance	H	farm

Figure 2.6: The following table provides the description of the features used in the models. The feature numbers correspond to the indices shown in Figure 2.5, and the features color-coded based on the clustering result shown in Figure 2.5. FP refers to the current finishing period, while H refers to the historical features. Furthermore the features are categorized into farm-level, group-level, area-level, and environment-level.

Chapter 3

Semi-supervised LSTM-VAE for Mortality Rate Prediction

In this chapter, we focus on the sow farm of a production system and different performance metrics for the sows and piglets.

The moralities in piglet can have different causes. These causes can be infection-based such as PRRS or non-infection such as genetics, sow's body condition, age at delivery, farrowing birth assistance, order of birth, environmental factors (heating control system) and etc. Vanderhaeghe et al. 2013. These moralities can occur at different times, before, during or after weaning (weaning is the shift in food of piglets from the sow's milk to a other foods). It can happen during early gestation and the fetus be resorbed, after day 40 of pregnancy but before farrowing (mummified piglet), or close to birth at farrowing (still-born). The general management practices of the farm can affect these mortalities.

Pigs can die from PRRS at any age. The reproductive failure caused by PRRS can occur in both early and late gestation, when the virus has the ability to cross the placental wall and infect the embryos (early developmental stage within the uterus). Embryos dead prior to 35 days are generally resorbed. However, in late pregnancy the result of infection is more sever and can cause abortion of up to 40% Pena et al. 2019. It can also cause the early farrowing and the birth of weak piglets with different abnormality at a later time.

We aim to predict the still-born and mummified mortality rates. We also are interested in farrowing rates (the proportion of females served that farrow) in sow farms of a production system. Building a predictive model can help the farm management better understand these mortality rates. We are looking to build a prediction model based on the data of the years 2016 to 2021 of a large swine production system. It is a time series data that provides weekly update for the breeding performance, farrowing, weaning management (laction), inventory, the number of death and culling. This data indicate the general management practice of a farm together with the mortality rates.

To solve this problem we deploy a predictive model known as variational recurrent auto-encoder. Variational recurrent auto-encoder, can combine the advantage of VAE and LSTM. This model leverages the powerful recurrent Long Short Term Memory (LSTM) neural network and variational Auto Encoder (VAE). LSTM exploits the time dependency for better prediction while VAE learns the distribution of the data by learning the latent space. We also take a semi-supervised approach allowing for prediction of response variables in a semi-supervised setting.

3.1 Introduction

We are looking at a well-known class of models, known as semi-supervised deep generative models, endowed with LSTM networks, for predicting multivariate time-series based on features that are in form of a time series themselves. We are focusing on semi-supervised approaches since in supervised learning problems, including classification and regression, we are sometimes dealing with a dataset in which a portion of the data is unlabelled and the response variable has no value. Classical methods only made use of the labeled portion of the data to train a supervised model. However, there are still a lot of information in the unlabeled data that can inform the modeling of the dependent variable. The term semi-supervised learning mitigate this issue by using both unlabeled and labeled data to improve supervised learning's generalization error. The main goal of this techniques is to get the best performance with a lowest amount of labelled data, Zhu and Goldberg 2009.

Furthermore, the problem at hand is a nonlinear regression problem that arises because we have different variables in the dataset, and the goal is to predict the value of one variable, known as dependent variable, given the values of the other independent variables. In general, regression analysis builds a mathematical model based on the data that allows the prediction of the output variable to be most accurate Mendenhall, Sincich, and Boudreau 2003. The regression problem can be extended in the semi-supervised setting, i.e., given a set of labeled training data and a set of unlabeled data points, the goal is to predict the dependent variable, also known as regressand, for any new observation. In contrast to the supervised learning, where we only use the labeled data, in this case, the insight from the unlabeled data points is also deployed by the learning algorithm Kostopoulos et al. 2018.

Semi-supervised deep generative models are a sub-group of latent variable models, which are statistical models that contain latent variables, i.e. unobserved variables. In addition to the types of observed variables, latent variables can be discrete or continuous. This allows for broad classifications of latent variable models. Formally, a latent variable model is a probability distribution over two sets of variables $\mathbf{x}$ and $\mathbf{z}$: $p(\mathbf{x}, \mathbf{z}; \theta)$ where the $\mathbf{x}$ is the observed variable, and $\mathbf{z}$ is the latent variable. We have both discriminative and generative latent variable models. We use latent variable models when some data in our model is unobserved. These models also enable us to leverage the prior knowledge when defining a model. They can also be viewed as a tool to increase the expressive power of our model. For example in case Gaussian mixture models, we can model a much more expressive distribution using a mixture of Gaussian than using a single Gaussian.

The learning process in semi-supervised deep generative models is according to the framework of Bayesian inference, which is a statistical approach for updating the a priori known probability distribution over unobserved variable using Bayes' theorem when more empirical observation becomes available. More formally, in the Bayesian inference framework, we have some unobserved variable, or a set of parameters, denoted as θ, and some observed data or evidence, which is, denoted as $\mathbf{x}$. The problem of probabilistic inference is considered the problem of calculating the conditional probability distribution over θ, given the evidence $\mathbf{x}$.

3.2 Semi-supervised Deep Generative Model

The goal of this chapter is to formalize the problem of mortality rate prediction and modeling as a latent variable modeling problem. Note only we care about the mortality rate itself, we also care about the feature values that cause a specific mortality rate. Formally, We have a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ containing pairs of $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_n \in \mathbb{R}^D$ is the nth observation and $y_n \in \mathbb{R}$ is the corresponding regressand in our semi-supervised regression model, which represent the mortality rate. We assume that our observed features have corresponding latent variables denoted by $\mathbf{z}_n$. We also assume that the mortality rate is only know for a subset of our observations, and we want to predict the rest in an inference procedure. Therefore, we have two different empirical distributions, one over the part of the data for which y is known $p_{\text{sup}}(\mathbf{x}, y)$, and one over the part of the data for which y is not known $p_{\text{un}}(\mathbf{x}, y)$. We deploy a semi-supervised generative modeling approach that improves predictive performance of the model by exploiting the generative descriptions of the data.

Generative Model for Features

In many of the complex real-world applications, data includes hundreds of features. Usually, we can reduce the dimensionality of the feature space to comparably lower dimension. We call the new space with the reduced dimension the embedding or representation space and we define it by a latent variable. In latent variable modeling, we construct a model that defines a feature representation of the data also known as embedding. If the data is generated according to several different lower dimensional factors, introducing the embedding space will allow for a clustering of observations in the latent feature space, which can be later used for accurate prediction of some dependent variable, e.g., regressand in the regression problem. We leverage neural networks as an expressive non-linear function estimator to construct a deep generative model of the data that provides a rich set of of latent features. The generative model can be formally described as:

$$
\begin{aligned}
p(\mathbf{z}) &= \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) \\
p_\theta(\mathbf{x} \mid \mathbf{z}) &= f(\mathbf{x}; \mathbf{z}, \boldsymbol{\theta})
\end{aligned} \tag{3.1}
$$

where $f(\mathbf{x}; \mathbf{z}, \boldsymbol{\theta})$ is likelihood probability distribution which can take the form of a Gaussian, Bernoulli, or any other appropriate distribution. Note that the parameters of these probability distributions, e.g., mean and variance in case of a Gaussian distribution, are non-linear

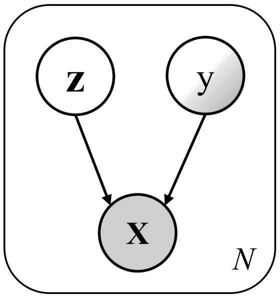


Figure 3.1: Probabilistic graphical model for semi-supervised generative model based on Durk P Kingma et al. 2014

transformations of the latent variables $\mathbf{z}$. These transformations are parameterized with deep neural networks, in which the weights are collectively denoted by $\boldsymbol{\theta}$).

In a classification problem, we use approximate samples from the posterior distribution of the latent variables $p(\mathbf{z}|\mathbf{x})$ to train a classifier for predicting the class labels. We can use the same methodology in a regression problem, i.e., using the posterior distribution samples as a guide to predict the dependent variable y, the mortality rate. Hence, we can do regression in a lower dimensional space, which is the latent space.

Generative Model with Regressand

For the part of data with known dependent variable y, we can assume that the observed data is generated by a latent regressand variable y in addition to a latent variable $\mathbf{z}$. In this case, the genrative model is described as:

$$\begin{aligned} p(y) &= \mathcal{N}(y \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \\ p(\mathbf{z}) &= \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}), \\ p_\theta(\mathbf{x} \mid y, \mathbf{z}) &= f(\mathbf{x}; y, \mathbf{z}, \boldsymbol{\theta}) \end{aligned} \tag{3.2}$$

where $\mathcal{N}(y \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covaraiance $\boldsymbol{\Sigma}$, the dependant variable y is treated as a latent variable when no y is recorded and $\mathbf{z}$s are the other latent variables. Here, as shown the graphical model of figure. 3.1 we assume that these two set of latent variables are marginally independent, which is a valid assumption in that it separates the regression-specific dependencies from other variation in

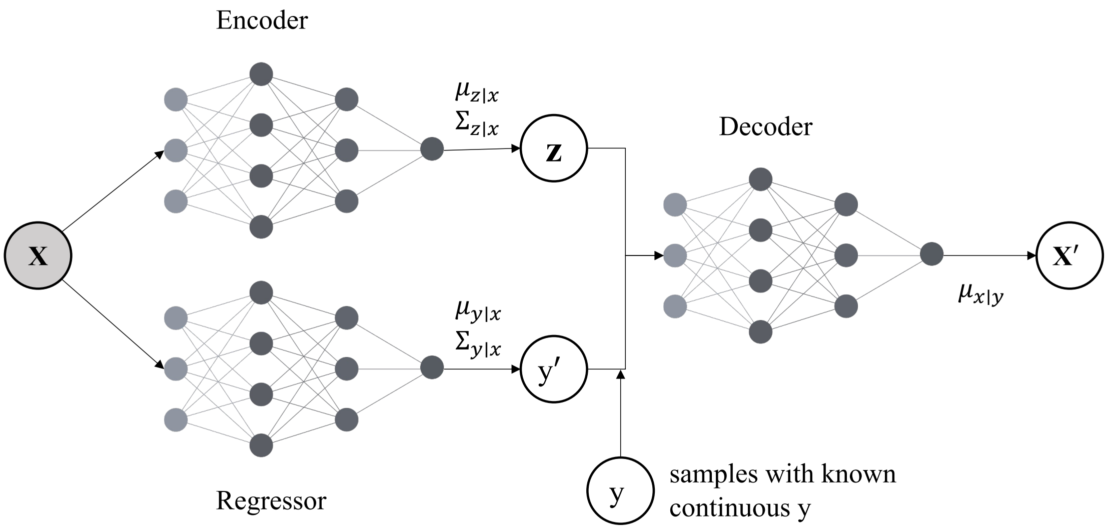


Figure 3.2: Variational autoencoder for semi-supervised learning.

the fully observed data **x**. As before, $f(x; y, z, \theta)$ is a non-linear likelihood function such as Gaussian distribution, which is parameterized by a non-linear transformation of all the latent variables. This function is parameterized with deep neural networks representing the non-linear transformation. For the sample with unobserved y, this value is estimated during the inference process. The inference process involves predictions for the y value based on the inferred posterior distribution $p_\theta(y \mid x)$.

Generative Model for Observed and Unobserved y

Semi-supervised learning approach involves learning a latent representation using the generative model for samples with unobserved y, and then learning a generative semi-supervised model, using this representation instead of the raw data **x**. Therefore, we have a deep generative model with two layers of stochastic variables (see figure. 3.2).

3.3 Variational Inference

In almost the variational autoencoder models, the exact computation of posterior distribution is intractable because of the nonlinear, non-conjugate dependencies between the random variables. The advances in variational inference (Diederik P Kingma and Welling 2014a; Rezende, Mohamed, and Wierstra 2014) made it possible to perform inference and parameter learning. To this end, an approximate distribution denoted as $q_\phi(\mathbf{z} \mid \mathbf{x})$ with parameters ϕ is introduced to approximate the true posterior distribution $p(\mathbf{z} \mid \mathbf{x})$. A lower bound on the marginal likelihood of the model $p(\mathbf{x})$, or $p(\mathbf{x}, y)$, is derived according to the variational principle, which is thought of as the objective function. The ultimate goal is to force the approximate posterior to be as close as possible to the true posterior.

A well-known approach for for efficient variational inference is to build the approximate posterior distribution $q_\phi(\cdot)$ as an inference model (Dayan 2000; Diederik P Kingma and Welling 2014a; Rezende, Mohamed, and Wierstra 2014). In standard variational inference methods, variational parameters should be computed for each data point, and hence, the number of parameters grows with the number of samples. In this study, we use an inference network, which is at its core a network parameterized with a set of global variational parameters ϕ. This technique, known as amortized variational inference, diminishes the need to compute per data point variational parameters. In this way, the cost of inference is reduced as the posterior estimates for all latent variables can be computed through learning the parameters of the inference network. In addition, using a single inference network allows for fast inference at both training and testing time.

The inference network introduced for all latent variables are parameterized as deep neural networks in which the outputs are the parameters of the approximate distribution $q_\phi(\cdot)$. We introduce two approximate distributions: (i) one for the samples with unknown y denoted as $q_\phi(\mathbf{z} \mid \mathbf{x})$ for the latent variable $\mathbf{z}$ and (ii) one for the samples with observed y denoted as $q_\phi(\mathbf{z}, y \mid \mathbf{x})$ for each of the latent variables $\mathbf{z}$ and y. For the latter, we assume a factorized form as

$$q_\phi(\mathbf{z}, y \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x}) q_\phi(y \mid \mathbf{x}), \tag{3.3}$$

where $q_\phi(\mathbf{z} \mid \mathbf{x})$ and $q_\phi(y \mid \mathbf{x})$ have appropriate Gaussian distributions. We have

$$\text{Unobserved } y\text{: } q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{z} \mid \boldsymbol{\mu_\phi(\mathbf{x})}, \mathrm{diag}(\boldsymbol{\sigma^2_\phi(\mathbf{x})})\big)$$

$$\text{Observed } y\text{: } q_\phi(\mathbf{z} \mid y, \mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu_{\phi_1}(y, \mathbf{x})}, \mathrm{diag}(\boldsymbol{\sigma^2_{\phi_1}(\mathbf{x})})),$$

$$q_\phi(y \mid \mathbf{x}) = \mathcal{N}(y \mid \boldsymbol{\mu_{\phi_2}(\mathbf{x})}, \mathrm{diag}(\boldsymbol{\sigma^2_{\phi_2}(\mathbf{x})})).$$

Lower Bound Objective

The variational bound $\mathcal{J}(\mathbf{x})$ for the samples with unobserved y on the marginal likelihood for a single data point can be derived as:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &\geq E_{q_\phi(y,\mathbf{z}|\mathbf{x})}\Bigg[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p_\theta(\mathbf{z}) - \log q_\phi(y,\mathbf{z}|\mathbf{x})\Bigg] \\
&= E_{q_\phi(y|\mathbf{x})}\Bigg[E_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \text{const} + \log p_\theta(\mathbf{z}) - \log q_\phi(y|\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})\big]\Bigg] \\
&= E_{q_\phi(y|\mathbf{x})}\Bigg[E_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|y,\mathbf{z})\big] + \text{const} - KL[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})] - \log q_\phi(y|\mathbf{x})\Bigg] \\
&= E_{q_\phi(y|\mathbf{x})}\big[-\mathcal{L}(\mathbf{x},y) - \log q_\phi(y|\mathbf{x})\big] \\
&= \sum_y \big[q_\phi(y|\mathbf{x})(-\mathcal{L}(\mathbf{x},y)) - q_\phi(y|\mathbf{x})\log q_\phi(y|\mathbf{x})\big] \\
&= \sum_y q_\phi(y|\mathbf{x})(-\mathcal{L}(\mathbf{x},y)) + \mathcal{H}(q_\phi(y|\mathbf{x})) = -\mathcal{J}(\mathbf{x})
\end{aligned} \tag{3.4}$$

In this model, the inference network $q_\phi(\mathbf{z} \mid \mathbf{x})$, which is parameterized by neural network, uses both samples with observed y and samples with unobserved y. This approximate posterior is then used as a feature extractor for the samples with known y, and the features used for training the regressor.

For the samples with observed y, the variational objective for a single data point $(\mathbf{x}, y)$ is:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}, y) \geq E_{q_\phi(\boldsymbol{\mu}, \Sigma, \mathbf{z}|\mathbf{x}, y)} &\Bigg[\log p_\theta(\mathbf{x}|y, \mathbf{z}, \boldsymbol{\mu}, \Sigma) + \log p_\theta(\boldsymbol{\mu}, \Sigma|y) \\
&+ \log p_\theta(y) + \log p_\theta(\mathbf{z}) - \log q_\phi(\boldsymbol{\mu}, \Sigma, \mathbf{z}|\mathbf{x}, y) \Bigg] \\
&= E_{q_\phi(\mathbf{z}|\mathbf{x})} \Bigg[\log p_\theta(\mathbf{x}|y, \mathbf{z}) + \log p_\theta(y) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}) \Bigg] \\
&\quad + E_{q_\phi(\boldsymbol{\mu}, \Sigma|\mathbf{x})} \Bigg[\log p_\theta(\boldsymbol{\mu}, \Sigma|y) - \log q_\phi(\boldsymbol{\mu}, \Sigma|\mathbf{x}) \Bigg] \\
&= -\mathcal{L}(\mathbf{x}, y) - KL[q_\phi(\boldsymbol{\mu}, \Sigma|\mathbf{x}) || p_\theta(\boldsymbol{\mu}, \Sigma|y)] \\
&\geq -\mathcal{L}(\mathbf{x}, y) + \alpha \log q_\phi(y|\mathbf{x}) + \text{const } 2
\end{aligned} \tag{3.5}
$$

where α is a hyper-parameter.

Training the Semi-Supervised Model

For training the semi-supervised model, the final loss function is of the following form:

$$
\mathcal{J} = \sum_{\mathbf{x} \in \mathcal{D}_{unlabelled}} \big[\sum_y q_\phi(y|\mathbf{x})(\mathcal{L}(\mathbf{x}, y)) - \mathcal{H}(q_\phi(y|\mathbf{x})) \big] + \sum_{(\mathbf{x}, y) \in \mathcal{D}_{labelled}} \big[\mathcal{L}(\mathbf{x}, y) - \alpha \log q_\phi(y|\mathbf{x}) \big] \tag{3.6}
$$

With the loss function of equation 3.6, we train the network by taking a mini-batch, sampling the priors, computing the values in the networks, and computing the loss function. The ELBO bound includes the parameters θ and ϕ that are related to the generative distributions, which defines the distribution over data $p_\theta(X)$, and the variational distribution, respectively. We maximize this bound with respect to the parameters θ to learn the generative model and maximize it over the parameters ϕ to perform inference.

3.4 LSTM for Encoder and Decoder Networks

In order to combine the advantages of VAE with recurrent networks, such as LSTM, we use LSTM networks for both encoder and decoder part of our semi-supervised VAE model.

Recurrent Neural Network (RNN) are a type of neural network designed for time series data. The RNN has two major problems and the Long Short Term Memory (LSTM) was developed to address these problems. First the so called vanishing and exploding gradient descent: During the training of the neural networks with gradient decent algorithm, the goal is to minimize the cost function error by back-propagating that error and updating the weights of the neurons that participated in creation of the error. These neurons in feed forward neural networks are below each other from one layer to the previous one and the error needs to be propagated back to the input of the same time.

In a recurrent neural network the error should be additionally back-propagated through time to the first neuron that was used for the prediction of the output of current time stamp. The so called problem of vanishing gradient descent occurs when multiple weights have value less than one and the result of this multiplication would become so small. This problem is more severe for the RNN as the recurrent weights (those that hold the information from the previous time points) would be multiplied by themselves multiple times (depending on how long of history the model considers in time) and the gradient decline rapidly for small values of weights. The similar problem of exploding gradient decent appears if the values of weights are large. The problem with gradient values is addressed in LSTM architecture by setting the value of the recurrent weights to 1. This technique of preserving error in temporal weights is called constant error carousel and is the key feature of LSTM. Another problem of RNN that LSTM can address is the shortcoming of RNN in dealing with long term dependencies and dealing with understanding of the context. In LSTM there exists two state variables, namely cell and hidden state variables. The hidden state is known as Short term memory and the cell state is known as Long term memory. The cell state goes straight through the cell with only some minor linear interactions to the next cell. The memory problem of RNN is addressed by deploying a series of probabilistic gates that decides when to keep, forget or ignore data that are being transferred to the next LSTM cell. Gates are composed of a sigmoid neural net layer and multiplication operation to optionally let information through.

Forget Gate: A forget gate is responsible for throwing away the irrelevant information stored in the cell state. This gate decides how much of the information coming from the previous timestamp is to be remembered or forgotten. This decision is based on the hidden state from the previous cell and the input at current time step. After going through a sigmoid function the output will be between 0 and 1. The value of zero means that the forget gate decides the cell state to completely forget that piece of information. In other words, forget gate learns to reset the internal state of the memory cell.

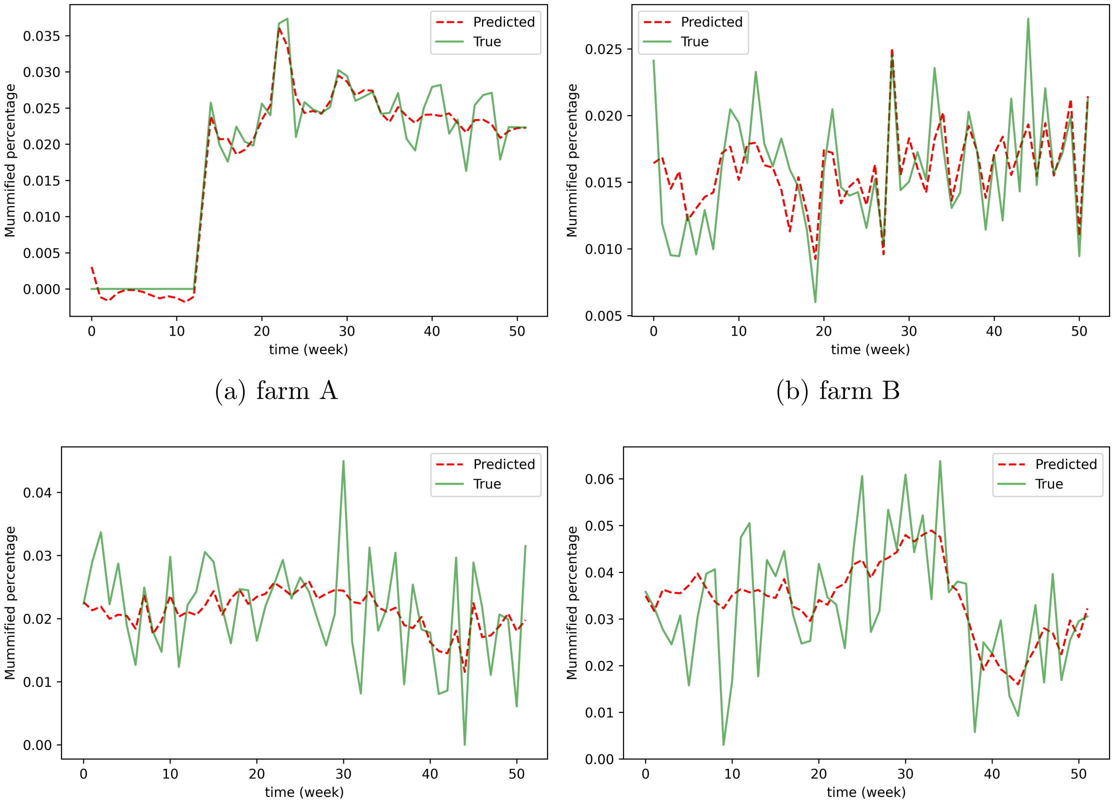
Input Gate: The input gate is responsible to learn new information from the input by adding the information to the cell state. A sigmoid layer decides which values to update and a tanh layer creates a vector of new candidate values and the cell state is updated accordingly.

Output Gate: The output gate creates a filtered version of the updated information from the current timestamp to the next timestamp. A sigmoid layer is used to decide which parts of the cell state should be passed to the next cell. Then, the values of cell state are mapped to -1 and 1 by going through tanh and then multiply it by the output of the sigmoid gate, to only transfer the information we want.

3.5 Experimental Results

We have access to a time series data that provides weekly update for the breeding performance, farrowing, weaning, laction, inventory, the number of death and culling. This data indicate the general management practice of a farm together with the mortality rates. These moralities can have different causes including PRRS.

The main features in this study includes, average total female inventory obtained by dividing the total number of days that all female lived by the the number of days in period. Average mated inventory is obtained by dividing the total number of mated female by the number of days in period. Average gilt pool inventory is obtained by dividing the total number of days gilts have been in inventory by the number of days in period. For the weaning, we use the total number of weaning and normalized version by the number of sow, average age of weaning, the pigs weaned divided by sow farrowed, pig weaned by number of mated female pigs. For laction performance, total piglet death and normalized version by the inventory are used as features. Also the Pre-wean mortality is defined as the sow weaned cohort piglet death divided by the number of born alive piglet. For farrowing, average



(a) farm A

(b) farm B

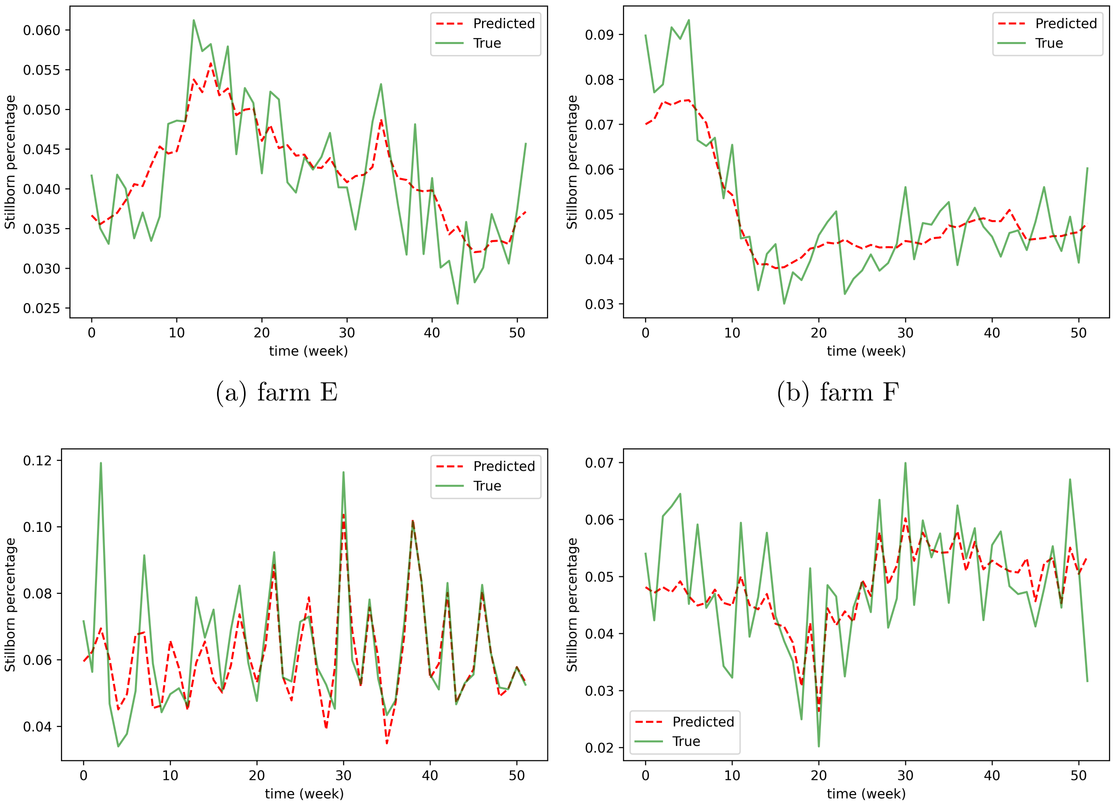
(c) farm C

(d) farm D

Figure 3.3: Semi-supervised learning of mummified percentage. The green curve shows the true value, the red curve is the inference for samples with known y, $(\mathbf{x}, y)$, and the orange curve shows the inference of the value y for samples with unknown y. Mummified piglets are born with the remainder of the litter. Infectious disease such as PRRS can be the cause of mummifcation and piglet deaths. The average normalized error bound across time for farm A, B, C, and D are 9% ,13%, 12% and 16%, respectively.

total born is calculated by dividing the total born pig by the total farrowed in the period. The total pig stillborn and total pig mummified can be good indicators of different diseases especially PRRS.

We show the performance validation of this model on predicting future trend in the time series using the experiment we describe in the following. We use various important features such as Mummified percentage, Litters/mated female/yr(LMFY), Stillborn percentage, Gilt pool inventory percentage, etc as input features $(\mathbf{x})$ and keep one as target feature(y). Figure. 3.3, figure. 3.4, figure. 3.5, and figure. 3.6 illustrates the regressand y associated to mummified mortality rate, Stillborn percentage, Pre-wean mortality rate, Sows farrowed percentage for four different farms, respectively. The first 80% of the dataset is used as samples with observed y, while we assume the rest of the dataset (20%) has no observed y, for which we need to infer the value of y. The green curve shows the true value, the red curve is the inference for samples with known y, $(\mathbf{x}, y)$, and the orange curve shows the inference of the value y for samples with unknown y, $(\mathbf{x})$. It can be observed that the smoother is the trend, the better is the ability of the model to capture the variation in the time series.



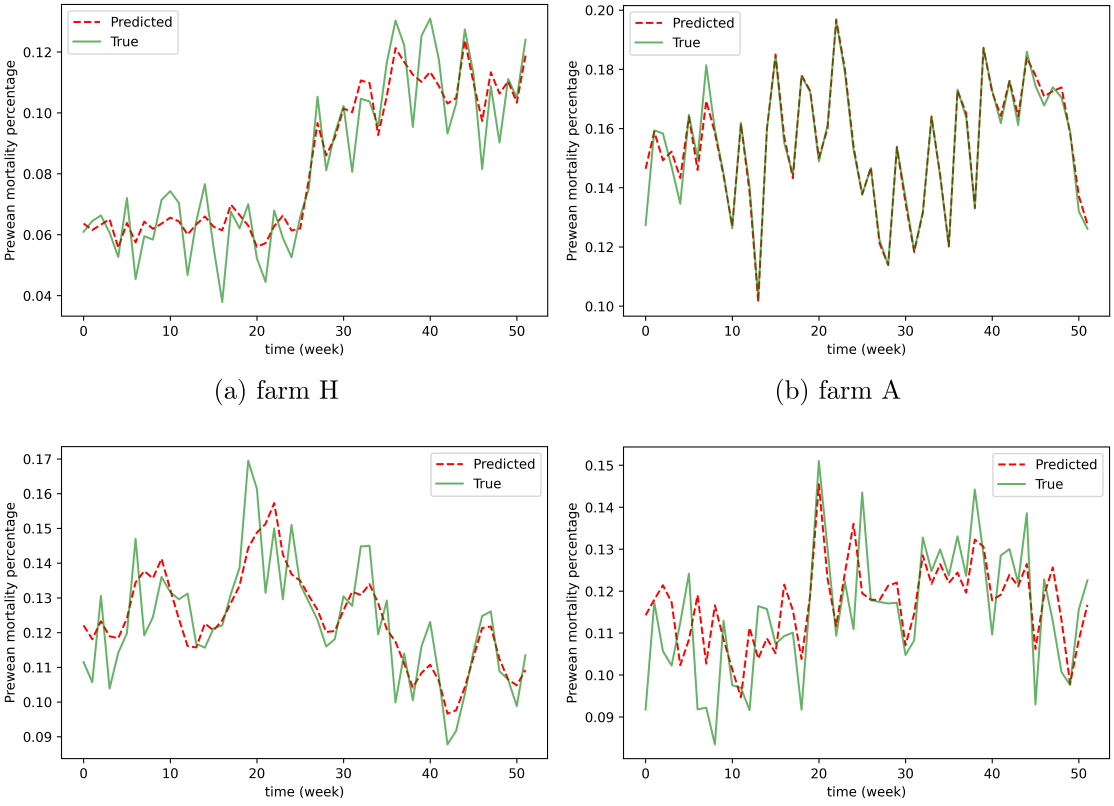
(a) farm E

(b) farm F

(c) farm G

(d) farm H

Figure 3.4: Semi-supervised learning of stillborn percentage. The green curve shows the true value, the red curve is the inference for samples with known y, $(\mathbf{x}, y)$, and the orange curve shows the inference of the value y for samples with unknown y. The average normalized error bound across time for farm E, F, G, and H are 9% ,9%, 11% and 15%, respectively.



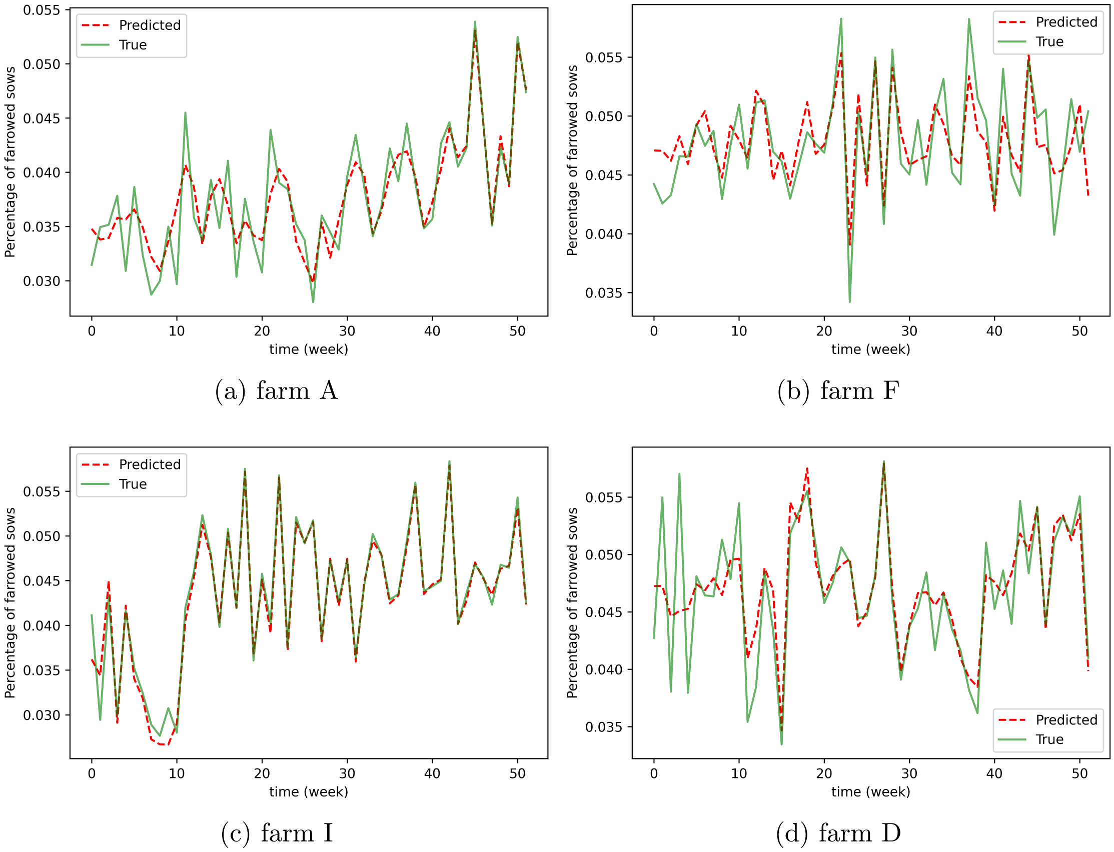
(a) farm H

(b) farm A

(c) farm B

(d) farm D

Figure 3.5: Semi-supervised learning of pre-wean mortality percentage. The green curve shows the true value, the red curve is the inference for samples with known y, $(\mathbf{x}, y)$, and the orange curve shows the inference of the value y for samples with unknown y. The average normalized error bound across time for farm A, B, C, and D are 8% ,15%, 12% and 18%, respectively.



(a) farm A

(b) farm F

(c) farm I

(d) farm D

Figure 3.6: Semi-supervised learning of percentage of farrowed sows. The green curve shows the true value, the red curve is the inference for samples with known y, $(\mathbf{x}, y)$, and the orange curve shows the inference of the value y for samples with unknown y. The average normalized error bound across time for farm A, B, C, and D are 12% ,11%, 13% and 14%, respectively.

3.6 Discussion & Conclusion

The moralities in piglets can occur in different stages of production and for infection and none-infection reasons. While the historical management practices in a farm determine the expected mortality rates in that farm, a forecasting model and analysis of variables can help the management to better understand the underlying factors. An increase in predicted mortality rates can be caused due to change in current risk factors of the farm. Based on a time series data of 2016-2021 with weekly update for the breeding performance, farrowing, weaning management, laction, inventory, the number of death and culling of a large production system, we built a machine learning model to predict the still-born, mummified and farrowing rates which are the mortality rates at different time instances.

The use of LSTM is a strong choice when dealing with time series data due to recurrent nature and its ability to forget and remember when necessary. The superiority of LSTM over other time series prediction models has been indicated in multiple studies (Siami-Namini, Tavakoli, and Namin 2018).

The generative framework allows for a semi-supervised approach in which we can use of all the data, with or without target variable. This enhance the learning of the latent variable and the mode

The data is none-smooth and none-stationary. The model can always learn the training data, however, it would over-fit and will not generalize to testing data. A tuned model can always capture the trend in testing data as can be seen in the figures 1-4 and majority of the time can detect the seasonal variation. We also tried random forest and gradient boosting in a supervised setting, and both had lower predictive performance on testing data.

Chapter 4

PRRS Outbreak Prediction: Deep State Space Modeling

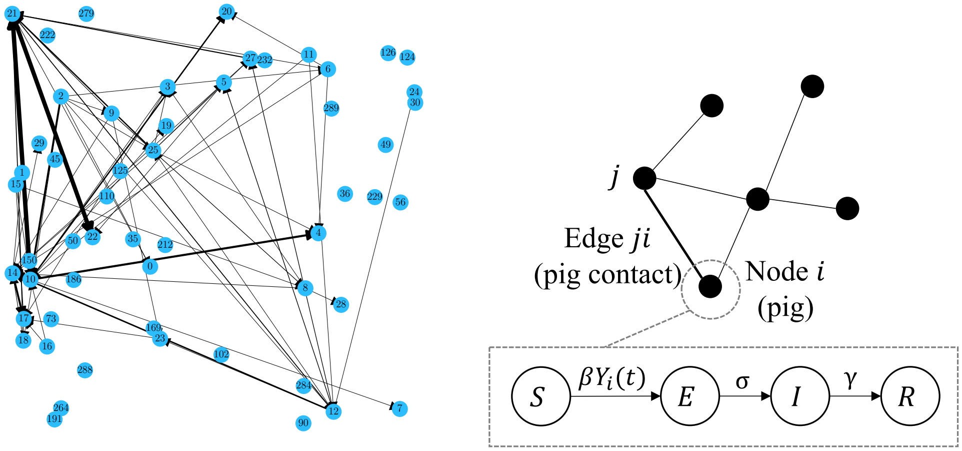
Abstract. We propose an epidemic analysis framework for the outbreak prediction in the livestock industry, focusing on the study of the most costly and viral infectious disease in the swine industry – the PRRS virus. Using this framework, we can predict the PRRS outbreak in all farms of a swine production system by capturing the spatio-temporal dynamics of infection transmission based on the intra-farm pig-level virus transmission dynamics, and inter-farm pig shipment network. We simulate a PRRS infection epidemic based on the shipment network and the SEIR epidemic model using the statistics extracted from real data provided by the swine industry. We develop a hierarchical factorized deep generative model that approximates high dimensional data by a product between time-dependent weights and spatially dependent low dimensional factors to perform per farm time series prediction. The prediction results demonstrate the ability of the model in forecasting the virus spread progression with average error of NRMSE $= 2.5\%$.

4.1 Introduction

Similar to chapter 2, the goal here is to develop a predictive models that can help to identify farms at high risk of infection to support risk-based, more cost-effective, target interventions. Such a framework will allow for more efficient testing, vaccination and outbreak prevention. Due to the high level of specialization in the swine industry, vast amount of data has been collected by the swine production systems. However, they have not yet been exploited enough due to difficulty of data access, integration and analyses (i.e., data are not consistently gathered, are non-standardized which makes their integration difficult, and are usually scattered across stakeholders). Examples of these data include diagnostic information, including the number of infected or dead animals, animal movements between farms or production data, which can give insights regarding farm health status and its contact network. Usually, these data do not satisfy the granularity required for learning an advanced predictive model (e.g. diagnostic samples are only taken once or twice per month per farm). However, using the real-world data, we can simulate epidemics to produce fine-grained time series data to analyze it further with an advanced novel prediction method based on a generative and variational inference model.

The direct contact with an infected pig is the main pathway for PRRS virus transmission. The network of between-farm movements, shown in Fig. 4.1a together with the intra-farm (local) pig-level contact pattern based on Susceptible-Exposed-Infectious-Removed (SEIR) epidemic model (see Fig. 4.1b) allows for constructing a system-level (global) pig-level disease transmission contact network, Ferdousi et al. 2019. Network-based SIR or SIR-extended epidemic models have been extensively studied in the literature Lee et al. 2017. In Newman 2002, Newman studied a network-based SIR epidemic model where infection is transmitted through a random network of contacts between individuals. The disease transmission contact network is a probabilistic graph that can be sampled to generate virtual contact. Using this, we generate fine-grained spatio-temporal time-series data based on statistics of real-world data.

The spatio-temporal data is often considered to have a high level of correlation between spatial dimensions, and, therefore, they can be assumed to be governed by a smaller number



(a) farm-level

(b) pig-level

Figure 4.1: Contact Network. (a) The swine shipment network (directed graph). The premises are displayed by a number-labeled node and edge weights corresponds to the shipment rate. The between-premises shipment rate network is showcased for 10% of nodes randomly selected among over 300 existing nodes. (b) **Top:** Pig level network graph. **Bottom:** State-transition diagram for a single node.

of underlying components. For modeling the temporal dynamic of the time series including the number of infected, dead, or recovered pigs, we employ a non-linear vector auto-regressive latent model inspired by the work of Farnoosh, Azari, and Ostadabbas 2021. Our spatio-temporal time-series data is first *factorized* into temporal weights and spatial factors. The temporal weights are modeled using a non-linear auto-regressive model parameterized by neural networks governed with a Markovian chain of discrete switches to capture higher-order multimodal latent dependencies, Becker-Ehmck, Peters, and Smagt 2019; Chang and Athans 1978; Ghahramani and Hinton 1996; Linderman et al. 2017; Nassar et al. 2019.

4.2 Background

Before diving into the state-of-the-art problem formulation that has been adopted in this work, we provide some history, preliminaries, and notation pertaining to linear Gaussian dynamical systems and the SEIR disease propagation model.

Switching State Space Model

Linear Gaussian dynamical systems operating in Markov dependent switching environment have long been investigated in the literature, Ackerson and Fu 1970; Chang and Athans 1978; Fox et al. 2009; Ghahramani and Hinton 1996; Hamilton 1990; Murphy 1998. These models, also known as switching linear dynamical system (SLDS), decompose nonlinear time series data into series of simpler, repeated dynamical modes. The SLDS model learns the underlying nonlinear generative process of the data as it breaks down the data sequences into coherent, potentially interpretable, discrete units, similar to the *piecewise affine* (PWA) framework in control systems Juloski, Weiland, and Heemels 2005; Paoletti et al. 2007; Sontag 1981 . The generative process starts with sampling a discrete latent state $s_t \in \{1, \ldots, S\}$ at each time $t = 1, \ldots, T$ according to Markovian dynamics $s_t \mid s_{t-1}, \mathbf{\Phi} \sim \pi_{s_{t-1}}$, where $\mathbf{\Phi}$ is the Markov transition matrix and π_s is the categorical distribution parameter. Then, a continuous latent state $w_t \in \mathbb{R}^K$ is sampled from a normal distribution whose mean follows a conditionally linear dynamics as $w_t = \mathbf{A}_{s_t} w_{t-1} + \mathbf{b}_{s_t} + \nu_{t-1}, \;\; \nu_{t-1} \overset{iid}{\sim} \mathcal{N}(0, \mathbf{Q}_{s_t})$, for matrices $\mathbf{A}_s, \mathbf{Q}_s \in \mathbb{R}^{K \times K}$ and vectors $\mathbf{b}_s \in \mathbb{R}^K$ for $s = 1, 2, \ldots, S$. Finally, a linear Gaussian observation $x_t \in \mathbb{R}^D$ is generated from the continuous latent state w_t according to $x_t = \mathbf{C}_{s_t} w_t + \mathbf{d}_{s_t} + \mu_t, \;\; \mu_t \overset{iid}{\sim} \mathcal{N}(0, \mathbf{G}_{s_t})$, for matrices $\mathbf{C}_s \in \mathbb{R}^{D \times K}, \mathbf{G}_s \in \mathbb{R}^{D \times D}$ and vectors $\mathbf{d}_s \in \mathbb{R}^D$. SLDS parameters are learned in a Bayesian inference approach. In this framework, the probabilistic dependencies are in such a way that $s_{t+1} \mid s_t$ is independent of the continuous state w_t, and hence the model cannot learn the transition of the discrete latent state when continuous latent state enters a particular region of state space. This problem is addressed in recurrent switching linear dynamical system (rSLDS), Linderman et al. 2017; Nassar et al. 2019 by allowing the discrete state transition probabilities to depend on the preceding continuous latent state, i.e, $s_t | s_{t-1}, w_{t-1}$. rSLDS studies proposed to use auxiliary variable methods for approximate inference in a multi-stage training process.

Nassar et al. 2019 extended rSLDS of Linderman et al. 2017 by enforcing a tree-structured prior on the switching variables in which subtrees share similar dynamics. Becker-Ehmck, Peters, and Smagt 2019 proposed to learn an rSLDS model through a recurrent variational autoencoder (rVAE) framework, and approximated switching variables by a continuous relaxation. This amortized inference compromised the applicability of their model on missing data, as they only included physics-simulated experiments.

SIR Model

Kermack–McKendrick established the fundamentals of mathematical modeling for how an epidemic spread through population. Based on the epidemiological status of each individual the population is divided into different compartments or classes and then different classes are related to each other with differential equations. In the basic form of this model the compartments are Susceptible (S), Infected (I) and Recovered (R). Susceptible individuals are those that are uninfected and susceptible to the disease (see Figure. 4.2). Infected individuals are infected and can infect susceptibles. Recovered individuals have been recovered from the infection and will not get infected again. This classification may not be related to an individual health status but indicates the ability of an individual in host and spread a pathogen Keeling and Danon 2009.

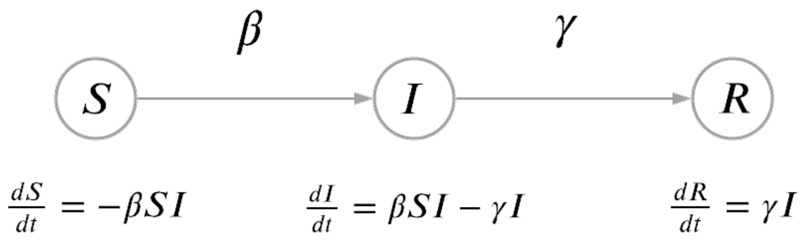

β
γ
S
I
R
$\frac{dS}{dt} = -\beta SI$
$\frac{dI}{dt} = \beta SI - \gamma I$
$\frac{dR}{dt} = \gamma I$

Figure 4.2: SIR transition model

Using this model the future of an epidemic process can be predicted. Given the initial population of each class and the transmission rate among classes, the future population of each class can be determined, thus we can know the number of infected individuals at each time. The rate at which a susceptible get infected and is moved to infected class is assumed to be proportional to the number of infected population. The rate at which an individual get infected can be determined as the multiplication of three factors. First the probability at which two individuals have direct contact. Second, the transmission probability of pathogen and third the number of infected individuals in the population.

In an SIR model, first the number of infected cases initially increases exponentially until the proportion of susceptible in the population has been sufficiently depleted that the growth rate slows. This process continues until the epidemic can no longer be sustained and the number of cases drops eventually leading to extinction of the infection.

The real-world predictability of the SIR models depend on accuracy of estimation the parameters of the model.

The SIR model can become one step closer to reality by considering the disease incubation time. For that, a new class called Exposed (E) is added to the model to represent those individuals that has been in contact with an infected individual and the pathogen has been transmitted to them but the symptoms has not yet appeared. When the symptom appears these individuals are moved from Exposed class to Infected class with rate σ (see Figure. 4.3).

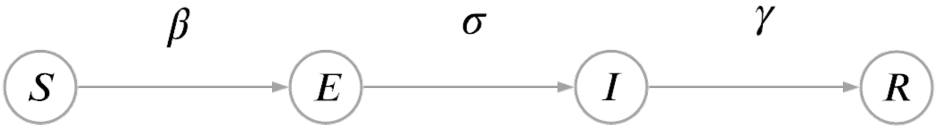


Figure 4.3: SEIR transition model

4.3 Time Series Data Simulation

Based on the rich database of an extensive anonymous swine production system located in the Midwest of the United States, we have access to farm-level pig shipment data, and PRRSV testing results Shamsabardeh et al. 2019; Mohammadsadegh Shamsabardeh, Azari, and Martıénez-López 2022. From 2006 to 2021, there have been over 260,000 movement records to or from farms within farm farm this production system. For each movement entry, the data includes the source and destination information, the number of transported pigs, and the date of the movement. Based on the farm-level shipment data we generate a *farm-level movement network* for the entire production system. Furthermore, the frequent PRRSV testing in each farm gives insight into how the virus is transmitted, e.g., what is the virus's transmission rate, incubation time, etc. Using the SEIR model we can produce a *pig-level contact network*, Ferdousi et al. 2019. The combination of this two network built on the statistics extracted from our real data results in an intricate contact network by which we can simulate complex time series data showing the number of infected, dead, or recovered pigs in each farm in the production system.

To create the farm-level movement network we build a probabilistic graph, i.e., a graph in which the existence of edges is uncertain with some probability. A node in the graph represents a farm, while the weight of an edge is proportional to the shipment rate (probability of shipment) between farms. Fig. 4.1a shows a part of the shipment network of over 300 farms for our real data. The edge thickness is representative of the shipment rate. For local (intra-farm) pig-level contact network in each farm, we consider a basic random graph model based on Erdös—Rényi model Erdos and Rényi 1959; Ferdousi et al. 2019, that produces pig contact graphs with an edge probability of 0.5 between any pair of pigs. The global (inter-farm) pig-level contact network is constructed when we sample a random generalization of a between-farm shipment over time (each day) and as a result, we create pig contact between farms (see Fig. 4.1b, top).

The simulation is formed on the network-based SEIR epidemic model for PRRS. In a network-based model, we consider a graph in which nodes represent individual pigs, and edges indicate direct or indirect contacts between pigs, which are considered infection pathways of PRRS. Each animal can be in one of the four states, Susceptible (S), Exposed (E), Infected (I), or Recovered (R) as the result of the epidemic progression. The state-transition diagram between these states is shown in figure. 4.1b, Bottom. In the generated swine pig-level network, a PRRS outbreak is introduced by randomly selecting a pig farm, and infecting an arbitrary random number of pigs. We collect several time snapshots representing the progression dynamics of the disease spread, such as the number of infected pigs in each farm over time. The healthy pigs which are free from PRRS virus infection are classified as Susceptibles. If such a healthy pig comes into contact with infected pigs containing the virus, it may get infected at the rate $\beta Y_i(t)$, where $Y_i(t)$ is the number of infected neighbors of node i at time t. If the transmission of pathogen occurs, a healthy pig enters into the Exposed group where it stays for the duration of the incubation period. On average, this period is denoted by $^1/_\sigma$. Once it shows symptoms, it moves into the Infected group. It stays there for an average time of $1/\gamma$ before it is recovered. We choose the parameters values $\beta = 0.087, \sigma = 7, \gamma = 6.5$ based on Charpin et al. 2012; Phoo-ngurn, Kiataramkul, and Chamchod 2019. This dynamic produces a spatio-temporal time series from all farms over time that can be used for predication modeling in the next section (Sec. 4.4).

4.4 Farm Disease Propagation Predication

Our spatio-temporal data indicates the number of pigs categorized within a particular stage, e.g., infected, recovered, etc., in every time instance in each farm. We denote this data as the matrix $X \in \mathbb{R}^{T \times D}$, where T is the number of time points and D the number of spatial locations, e.g., the number of farms. Building on previous work by Farnoosh, Azari, and Ostadabbas 2021, our assumption is that X can be decomposed into a weighted summation of $K \ll D$ factors over time as:

$$X \approx [w_1, \cdots, w_T]^\top [f_1; \cdots; f_K] = W^\top F, \tag{4.1}$$

where $f_k \in \mathbb{R}^D$ is the k^{th} spatial factor and $w_t \in \mathbb{R}^K$ is the weight vector at time t. Our intuition for adopting this model for some pig-specific collected measurements in D farm over T time points is that there are $K \ll D$ underlying factors using which we can approximate the overall dynamics of the disease propagation in the data. Our main goal is to predict the future outbreak behavior given the past time samples in each farm.

We assume that the weights, $W = \{w_t\}_{t=1}^{T}$, are generated according to a set of temporal lags, ℓ, through a deep probabilistic switching auto-regressive model. These weights are furthermore governed by a Markovian chain of discrete latent states, $\mathcal{S} = \{s_t\}_{t=1}^{T}$ as follows: $w_t \sim p(w_t | w_{t-\ell}, s_t),\ s_t \sim p(s_t | s_{t-1})$. In addition, we assume that spatial factors, $F = \{f_k\}_{k=1}^{K}$, are controlled by a shared low dimensional latent variable, z, as follows: $f_{1:K} \sim p(F|z),\ z \sim p(z)$. Fig. 4.4 shows the relation among above-mentioned random variables in a probabilistic graphical model diagram form.

We train the model using stochastic variational methods Hoffman et al. 2013; Diederik P Kingma and Welling 2014b; Ranganath et al. 2013; Rezende and Mohamed 2015 by approximating the posterior $p_\theta(\mathcal{S}, W, z, F|X)$ using a variational distribution $q_\phi(\mathcal{S}, W, z, F)$, and by maximizing a lower bound (known as ELBO) $\mathcal{L}(\theta, \phi) \leq \log p_\theta(X)$:

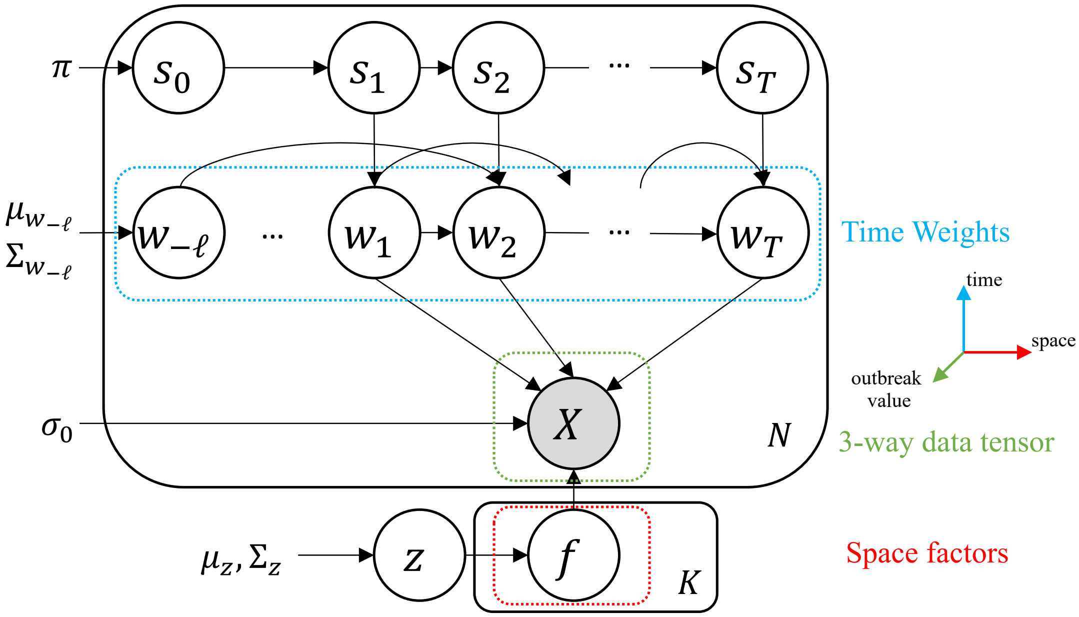


Figure 4.4: Probabilistic graphical model.

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathcal{S}, W, z, F)} \left[\log \frac{p_\theta(X, \mathcal{S}, W, z, F)}{q_\phi(\mathcal{S}, W, z, F)} \right] \tag{4.2}$$
$$= \log p_\theta(X) - \mathrm{KL}(q_\phi(\mathcal{S}, W, z, F) \,||\, p_\theta(\mathcal{S}, W, z, F|X)).$$

By maximizing the bound with respect to the parameters θ, we learn the generative distribution over datasets $p_\theta(X)$, and by maximizing the bound over the parameters ϕ, we do Bayesian inference by approximating the distribution $q_\phi(\mathcal{S}, W, z, F) \simeq p_\theta(\mathcal{S}, W, z, F|X)$ over latent variables for each data point. According to the graphical model in Fig. 4.4, the joint distribution of observations and latents will be:

$$p_\theta(X, \mathcal{S}, \mathcal{Z}) = p(F|z)p(z) \prod_{n=1}^{N} p(X_n|W_n, F) p(w_{n,-\ell}) p(s_{n,0})$$
$$\prod_{t=1}^{T} p(s_{n,t}|s_{n,t-1}) p(w_{n,t}|w_{n,t-\ell}, s_{n,t}), \tag{4.3}$$

where $\mathcal{Z} = \{W, z, F\}$). Furthermore, we assume a fully factorized variational distribution for the latent variables posterior as:

$$q(\mathcal{S}, \mathcal{Z}) = q(F)q(z) \prod_{n=1}^{N} q(w_{n,-\ell}) q(s_{n,0}) \prod_{t=1}^{T} q(s_{n,t}) q(w_{n,t}). \tag{4.4}$$

Generative Parameters

Here we describe the generative distribution parameters and model assumptions. We assume the variable $s_{n,t}$ to represents a categorical variable of dimensionality S, i.e., the number of modes/switches that a system can be in at a specific time t. The sequence of the discrete latents, $s_{n,1:T}$, are in form of a Markov chain and govern the state transitions over time with distributions:

$$\begin{aligned} p_\theta(s_t|s_{t-1}) &= \mathrm{Cat}(\mathbf{\Phi}_\theta\, \boldsymbol{\pi}_{s_{t-1}}) \\ q_\phi(s_{t-1}) &= \mathrm{Cat}(\boldsymbol{\pi}_{s_{t-1}}), \end{aligned} \tag{4.5}$$

where $\boldsymbol{\pi}_{s_{t-1}} = [\pi_1, \cdots, \pi_S]$ represents the probabilities of the categorical distribution for s_{t-1}, and $\mathbf{\Phi}_\theta \in \mathbb{R}^{S\times S}$ is a valid probability transition matrix.

For the temporal weights, w_t, we assume a switching Gaussian dynamic for the temporal latent transitions governed by the discrete latent states, s_t. In other words, we assume that the marginal distribution of temporal weights follows a Gaussian mixture distribution in the latent space, as:

$$p_\theta(w_t|w_{t-\ell}, s_t = s) = \mathcal{N}\Big(\mu^w_{\theta_s}(w_{t-\ell}), \Sigma^w_{\theta_s}(w_{t-\ell})\Big),$$

where $s \in \{1, \cdots, S\}$, and state-specific $\mu^w_{\theta_s}(\cdot)$ and diagonal $\Sigma^w_{\theta_s}(\cdot)$ are parameterized by multilayer perceptrons (MLPs), hence, follow a *nonlinear* vector auto-regressive model given $w_{t-\ell}$. Namely, we feed $w_{t-\ell}$ to a multi-head MLP for estimating the Gaussian parameters, e.g.,

$$\mu^w_{\theta_s} = \mathrm{FC}_s(h_s), \quad h_s = \sum_{l\in\ell} \sigma(\mathrm{FC}_{s,l}(w_{t-l})),$$

where FC denotes a fully connected layer, and σ is a non-linear activation function.

For the spatial factors, F, we assume a diagonal Gaussian distribution for spatial factors parameterized with an MLP as

$$p_\theta(F|z) = \mathcal{N}\big(\mu^F_\theta(z), \Sigma^F_\theta(z)\big) \tag{4.6}$$

where z is sampled from a normal distribution: $z \sim \mathcal{N}(0, I)$. The latent z is introduced as a low dimensional spatial embedding that encourages the estimation of a multimodal distribution among spatial factors. Given the temporal weights and spatial factors, we reconstruct the data by consolidating the two factorized part as:

$$X_n \sim p_\theta(X_n|W_n, F) = \mathcal{N}\Big(\big[w_{n,1}, \cdots, w_{n,T}\big]^\top F, \sigma_0^2\Big), \tag{4.7}$$

where σ_0 is a hyperparameter for observation noise.

Variational Parameters

The trainable variational parameters, ϕ, are assumed to have fully factorized distributions. The variational distribution for the continuous variables, $q(z;\phi^z)$, $q(F;\phi^F)$, and $\{q(w_{n,t};\phi^w_{n,t})\}_{n=1,\,t=-\ell}^{N,T}$, are considered to be Gaussian distributions with diagonal covariances. In addition, the variational parameters for the distribution of discrete latents, $\left\{q(s_{n,t};\phi^s_{n,t})\right\}_{n=1,\,t=1}^{N,T}$, are considered based on the mean-field approximation assumption to compensate information loss (see Farnoosh, Azari, and Ostadabbas 2021 for more detail.)

Training Procedure

The Monte-Carlo estimate of the gradient of ELBO is computed with respect to generative, θ, and variational, ϕ, parameters using a re-parameterized sample, Diederik P Kingma and Welling 2014b, from the posterior of continuous latents, $\{W, z, F\}$. For the discrete latent, $\mathcal{S}$, however, we compute the expectations by summing over the S, without the need for explicit sampling. This regularizes the S nonlinear auto-regressive priors based on their corresponding weighting. We can analytically calculate the Kullback-Leibler (KL) divergence terms of ELBO for both multivariate Gaussian and categorical distributions, which leads to lower variance gradient estimates and faster training as compared to e.g., noisy Monte Carlo estimates often used in literature. We use the Adam optimizer, Diederik P Kingma and Ba 2014, with learning rate of 0.01 for training. We initialized all the parameters randomly, and adopted a linear KL annealing Bowman et al. 2016 schedule to increase from 0.01 to 1 over the course of 100 epochs.

4.5 Experimental Results

Future Trend Prediction

For future sample prediction, we predict the test set sequentially using the generative model and spatial factors learned on the train set. We predict the next time point on the test set using the generative model and spatial factors learned on the train set: $\hat{X}_{t+1} = \hat{w}_{t+1}^{\top} F$, where $\hat{w}_{t+1} \sim p(\hat{w}_{t+1}|w_{t+1-\ell}, \hat{s}_{t+1})$, and $\hat{s}_{t+1} \sim p(\hat{s}_{t+1}|s_t)$. We then run inference on X_{t+1}, the actual observation at $t+1$ (if not missing), to obtain w_{t+1} and s_{t+1}, and add them to the historical data for prediction of the next time point $\hat{X}_{t+2}$ in the same way. We repeat these steps to make predictions in a rolling manner across a test set. We keep the generative model and spatial factors fixed during the entire prediction. We report normalized root-mean-square error (NRMSE%). The test set NRMSE% is related to the expected *negative test-set log-likelihood* for our case of Gaussian distributions (with a multiplicative/additive constant), hence it is used for evaluating the predictive generative models.

Swine Infection Progression Prediction

We used time series of epidemic progression from over 300 farms simulated for 700 time points. We kept last 20% of the time series as the test set. We then performed a short-term prediction tasks by adopting a rolling prediction scheme reported in Chen et al. 2019. For short-term prediction, the next time point is predicted on the test set using the generative model and spatial factors learned on the train set. We reported the test set normalized root-mean-square error (NRMSE%), which is related to the expected negative test-set log-likelihood for the case of Gaussian distributions, and it is used for evaluating the predictive generative models. We obtained NRMSE of 2.5% averaged over all the farms. Fig. 4.5 shows the number of infected pigs over time for nine selected farms. In this figure, we illustrate the actual number of infected pigs in the simulated data with a solid green curve, the mean estimate of the predictive model with a dashed purple curve, and the standard deviation of the data with a shaded red error bar. Each row in the figure represents a group of relatively highly connected farms in terms of the frequency of the pig shipments. Note that each group show a relatively strong correlation regarding the outbreak progression and the predictive model was able to capture this.

One observation regarding the performance of the model in situations when we do not have a curve with smooth behavior is the fact that when the increase and decrease of the number of infected pigs is abrupt, the model tends to converge to a middle point between the current value and the future one. This issue can be addressed using a model selection choice.

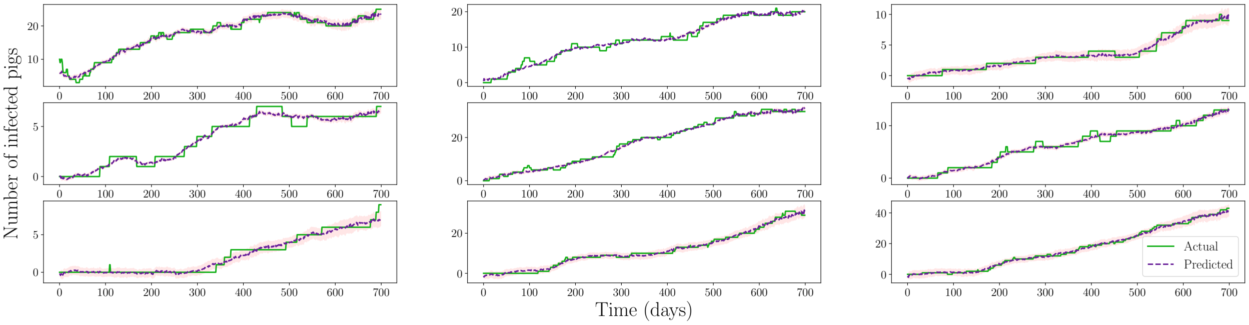


Figure 4.5: Short-term (one-day) predication. Each plot demonstrates the actual number of infected pigs in the simulated data (solid green), the mean estimate of the predictive model (dashed purple), and the standard deviation of the prediction estimate (shaded red error bar). Each row represents neighbouring farms that are connected in terms of pigs movement.

Specifically, we can control the number of factors K in order to increase the flexibility of the model in capturing different levels of curve smoothness.

Flu Spread Future Trend Prediction

To show the applicability of our model in the analysis of other time-series datasets and future trend prediction, we used Google Flu Trends (please see *Google Flu Trends Data* 2016 for dataset information), which represents the number of flu outbreaks in different countries. This dataset provides another enlightening example of future trend prediction in time series. We applied our training and prediction algorithm to the spatio-temporal Flu trend. The comparison with the ground truth can be seen in figure. 4.6. However, due to the page limitation, we would not incorporate this result into the paper. This is just to show the performance evaluation of our model for a publicly available dataset. Kindly note that we are committed to releasing the code and training example upon the publication date.

Model Performance for Non-Smooth Data

As explained in the method, the data matrix $X \in \mathbb{R}^{T \times D}$ (T: the number of time points, D: the number of spatial locations) can be decomposed into a weighted summation of $K \ll D$ factors over time as $X \approx [w_1, \cdots, w_T]^\top [f_1; \cdots ; f_K] = W^\top F$. Note that K is the

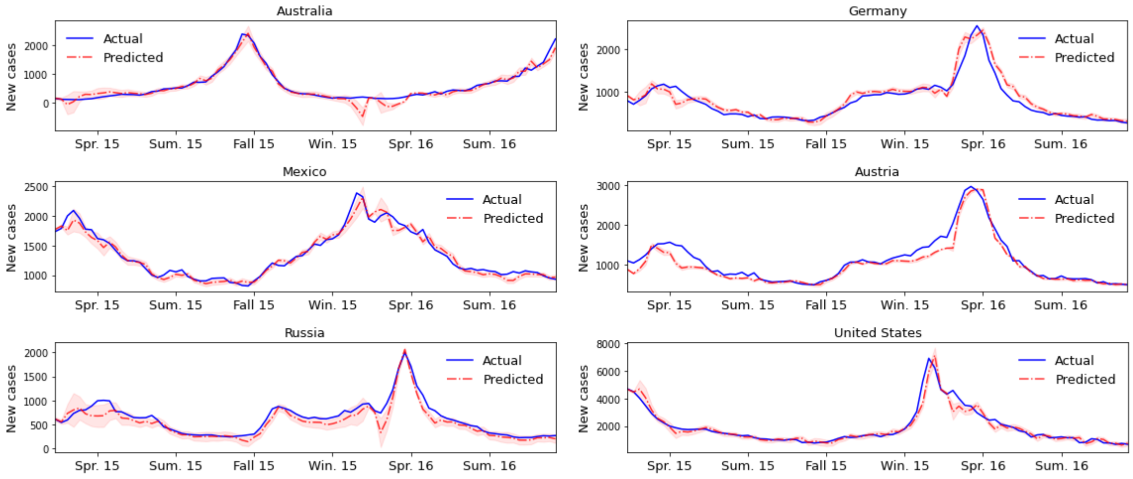
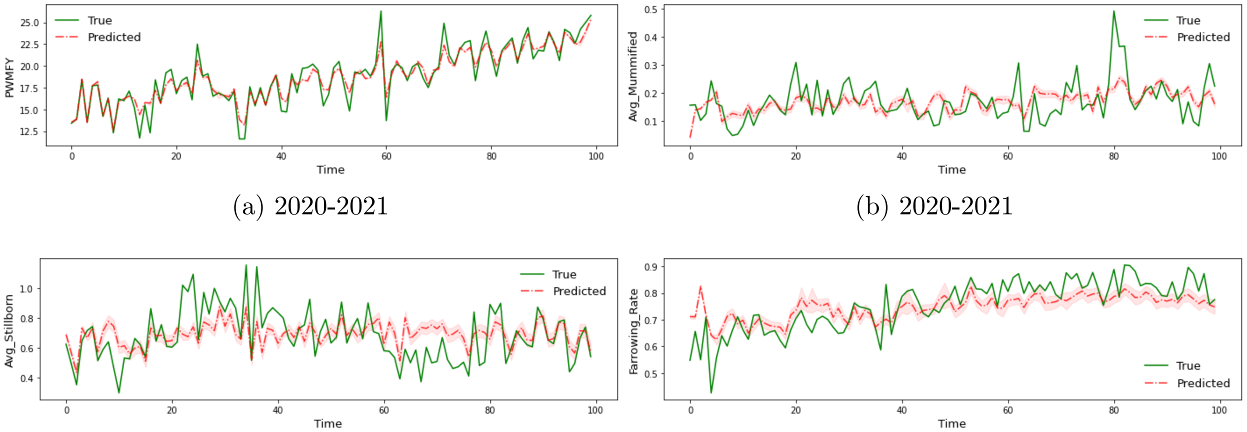


Figure 4.6: Google Flu future trend prediction.

number of *underlying factors* using which we can approximate the overall dynamic of the data. Therefore, K is a model order parameter (also referred to as a hyperparameter) that we will choose based on a compromise between good fit (negative log-likelihood) and complexity, Stoica and Selen 2004. The higher is K, the more complex the model is, i.e., we have a higher number of parameters. Therefore, we can capture non-smooth high-frequency variations in the signal. On the other hand, when K is small, the model cannot follow the rapid changes in the signal, and consequently, we observe some performance drop. However, note that although an efficiently parameterized model is unable to catch some abrupt changes in the data, it will generalize better than an over parameterized model.

Multivariate Model: Sow Farm Case Study

The generative model in this chapter was developed to address the spatio temporal dynamic of the disease propagation. However, we can use this model also as a multivariate predictive model for one specific location. In this scenario the spatioal dimension is replaced with the feature space. In this case the input and outputs are similar to the models in chapter one in the sense that some features are used to predict a response variable at one specific location. The data are the sow farms from previous chapter. To compare the performance of this model with those used in the previous chapter we use the same dataset and the model will learn all the features at the same time. The data composed of different categories. The Sow



(a) 2020-2021

(b) 2020-2021

(c) 2020-2021

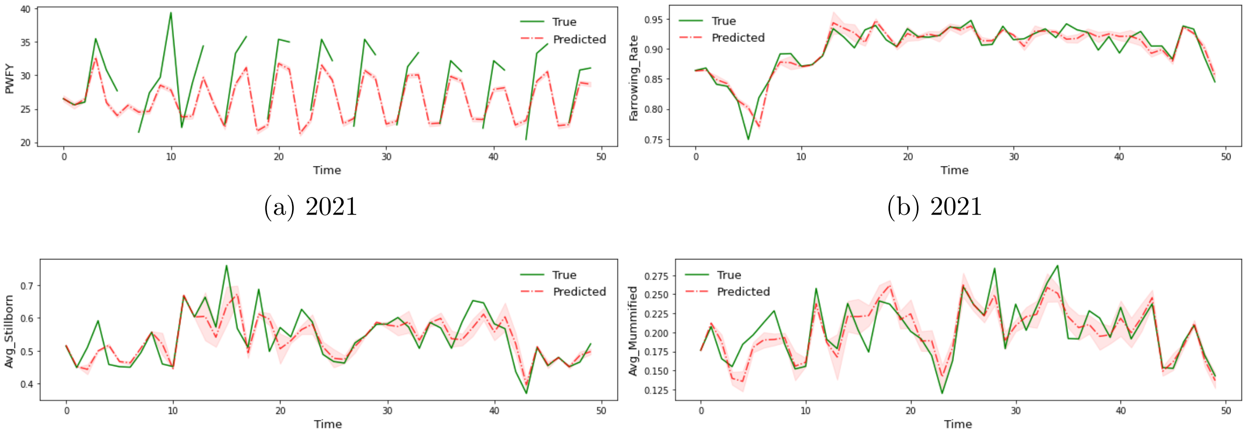
(d) 2020-2021

Figure 4.7: Farm A

Farm efficiencies are based on normalized weaned pigs. We have chosen the PWMFY in which the the total pig weaned are calculated per average mated female and per year. For the farrowing the number of stillborn and mummified are considered. The total pig stillborn in a period is normalized by dividing it to the total sow farrowed in the farm. Similarly, the total number of mummified pigs are divided by the total sow farrowed in the farm. The lactation metric is indicated by the pre-weaned mortality rate by calculating the ratio of the sows weaned cohort piglet deaths and born alive. The percentage of dead piglet is calculated by dividing the total piglet death in a period to average piglet inventory in the period. We also considered the farrowing rate and the percentage of multiple mating. We have used the data from 2016 to 2020 for training and the last two years for training.

4.6 Discussion & Conclusion

The PRRS outbreak cause an economic loss of over $664 million annually Holtkamp, Kliebenstein, et al. 2013, which can be significantly mitigated by early detection and risk-based intervention practices. Direct contact is the main disease transmission pathway. Therefore, the pig contact network provides a substantial basis to develop an outbreak prediction framework. We create a system-wide pig contact network by combining the SEIR epidemic model based on intra-farm infection transmission parameters, and inter-farm pig shipment network.



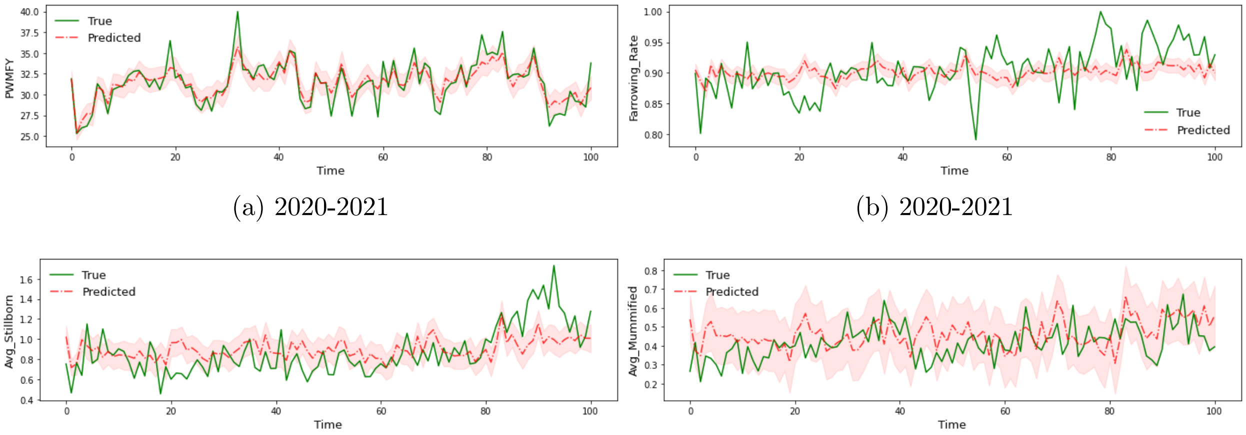
(a) 2021

(b) 2021

(c) 021

(d) 2021

Figure 4.8: Farm B



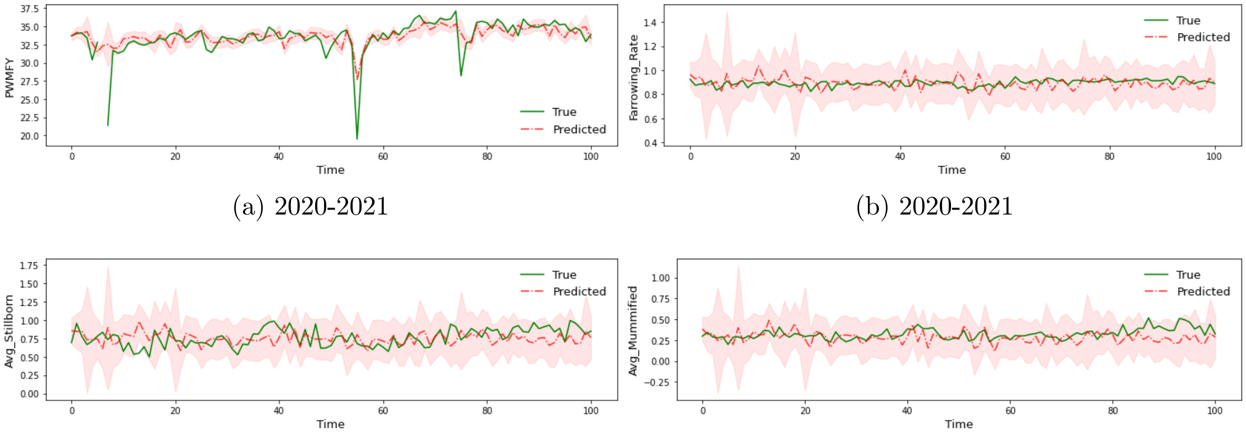
(a) 2020-2021

(b) 2020-2021

(c) 2020-2021

(d) 2020-2021

Figure 4.9: Farm C



(a) 2020-2021

(b) 2020-2021

(c) 2020-2021

(d) 2020-2021

Figure 4.10: Farm D

We presented a hierarchical factorized deep generative model of our spatio-temporal data that can capture the underlying dynamics of the disease spread with the aim to predict the number of infected pigs in all farms. Our result demonstrates the ability of the model in forecasting the virus spread progression with an average one-day prediction error of NRMSE = 2.5%. We also considered the model for multivariate prediction of sow farms. A potential future direction is to incorporate the per farm disease transmission parameters for the SEIR model to represent variations in the disease control implementation. Additionally, indirect disease transmission pathways, such as airborne, can be included in the framework for a more comprehensive analysis.

Chapter 5

Conclusion

In this thesis, we tried to address some of the challenges of swine industry by building data driven frameworks. Due to the high level of specialization in production systems, a vast amount of data has been collected in the swine industry, however, the usage of this data in animal health remains circumstantial, and is usually restricted to simple descriptive statistics or sequencing. One of the most important challenges are outbreaks specifically the PRRS. These outbreaks can bring food insecurity by causing animal loss and restricting the required trades among different farms to keep the production system sustainable. In this work we tried to use the machine learning approaches to build better models to understand and predict the occurrence of outbreaks in swine industry.

The PRRS outbreak cause an economic loss of over $664 million annually, which can be significantly mitigated by early detection and risk-based intervention practices. We built a framework to forecast the risk of having a PRRS outbreak on a farm. This forecasting allowed for early detection of disease outbreaks and could direct risk-based, and thus more cost-effective, interventions. Machine learning algorithms were trained using multi-scale data. For the first time, on-farm, between-farm, and environmental variables, including farm location, pig movements, production parameters, diagnostic data, and climatic information, were combined for the prediction of PRRS outbreaks. Multi-scale datasets were merged via feature extraction, followed by the wrapper and filter feature selection, to find those feature subsets with the best forecasting performance. The predictive value of each features selection mechanism was evaluated in terms of its stability. Numerical results demonstrate good forecasting performance in terms of area under the ROC curve.

Furthermore, we developed a semi-supervised variational auto-encoder (VAE) deploying Long Short Term Memory (LSTM) to predict the mortality rates (mummified and stillborn) and farrowing and weaning factors in the production system. The use of VAE allows for handling the missing data by building probabilistic model. We learned the target variable y with learning a latent representation using the generative model for samples with unobserved y, and then learning a generative semi-supervised model, using this representation instead of the raw data.

Finally, we created a system-wide pig contact network by combining the SEIR epidemic model based on intra-farm infection transmission parameters, and inter-farm pig shipment network. We presented a hierarchical factorized deep generative model of our spatio-temporal data that can capture the underlying dynamics of the disease spread with the aim to predict the number of infected pigs in all farms. Our result demonstrates the ability of the model in forecasting the virus spread progression with an average one-day prediction error of NRMSE $= 2.5\%$.

Bibliography

Ackerson, G and K Fu (1970). “On state estimation in switching environments”. In: *IEEE transactions on automatic control* 15.1, pp. 10–17.

Alkhamis, Moh A et al. (2017). “Novel approaches for spatial and molecular surveillance of Porcine Reproductive and Respiratory Syndrome Virus (PRRSv) in the United States”. In: *Scientific reports* 7.1, pp. 1–14.

Becker-Ehmck, Philip, Jan Peters, and Patrick van der Smagt (2019). “Switching Linear Dynamics for Variational Bayes Filtering”. In.

Bennett, Kristin P and Olvi L Mangasarian (1992). “Robust linear programming discrimination of two linearly inseparable sets”. In: *Optimization methods and software* 1.1, pp. 23–34.

Bitsouni, Vasiliki et al. (2019). “Predicting vaccine effectiveness in livestock populations: A theoretical framework applied to PRRS virus infections in pigs”. In: *PLoS One* 14.8, e0220738.

Bowman, Samuel R et al. (2016). “Generating Sentences from a Continuous Space”. In: *CoNLL 2016*, p. 10.

Breiman, Leo (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Chang, Chaw-Bing and Michael Athans (1978). “State estimation for discrete systems with switching parameters”. In: *IEEE Transactions on Aerospace and Electronic Systems* 3, pp. 418–425.

Charpin, Céline et al. (2012). “Infectiousness of pigs infected by the Porcine Reproductive and Respiratory Syndrome virus (PRRSV) is time-dependent”. In: *Veterinary Research* 43.1, pp. 1–11.

Chen, Xinyu et al. (2019). "Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model". In: *Transportation Research Part C: Emerging Technologies* 104, pp. 66–77.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Corzo, Cesar A et al. (2010). "HIF-1α regulates function and differentiation of myeloid-derived suppressor cells in the tumor microenvironment". In: *Journal of Experimental Medicine* 207.11, pp. 2439–2453.

Dayan, Peter (2000). "Helmholtz machines and wake-sleep learning". In: *Handbook of Brain Theory and Neural Network. MIT Press, Cambridge, MA* 44.0.

Dee, Scott, John Deen, Kurt Rossow, Carrie Weise, et al. (2003). "Mechanical transmission of porcine reproductive and respiratory syndrome virus throughout a coordinated sequence of events during warm weather". In: *Canadian journal of veterinary research* 67.1, p. 12.

Dee, Scott, John Deen, Kurt Rossow, Carrie Wiese, et al. (2002). "Mechanical transmission of porcine reproductive and respiratory syndrome virus throughout a coordinated sequence of events during cold weather". In: *Canadian Journal of Veterinary Research* 66.4, p. 232.

Erdos, P. and A. Rényi (1959). "On Random Graphs I". In: *Publicationes Mathematicae (Debrecen)* 6, pp. 290–297.

Evans, CM et al. (2010). "A stochastic mathematical model of the within-herd transmission dynamics of porcine reproductive and respiratory syndrome virus (PRRSV): fade-out and persistence". In: *Preventive veterinary medicine* 93.4, pp. 248–257.

Farnoosh, Amirreza, Bahar Azari, and Sarah Ostadabbas (2021). "Deep Switching Auto-Regressive Factorization: Application to Time Series Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8, pp. 7394–7403.

Ferdousi, Tanvir et al. (2019). "Generation of swine movement network and analysis of efficient mitigation strategies for African swine fever virus". In: *PloS one* 14.12, e0225785.

Fox, Emily et al. (2009). "Nonparametric Bayesian learning of switching linear dynamical systems". In: *Advances in neural information processing systems*, pp. 457–464.

Freund, Yoav, Robert Schapire, and Naoki Abe (1999). "A short introduction to boosting". In: *Journal-Japanese Society For Artificial Intelligence* 14.771-780, p. 1612.

Friedman, Jerome H (2001). “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics*, pp. 1189–1232.

Garcia, Rodrigo et al. (2020). “A systematic literature review on the use of machine learning in precision livestock farming”. In: *Computers and Electronics in Agriculture* 179, p. 105826.

Ghahramani, Zoubin and Geoffrey E Hinton (1996). *Switching state-space models.* Tech. rep. Citeseer.

Google Flu Trends Data (2016). https://www.google.org/flutrends/about/.

Granitto, Pablo M et al. (2006). “Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products”. In: *Chemometrics and intelligent laboratory systems* 83.2, pp. 83–90.

Guyon, Isabelle and André Elisseeff (2003). “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Guyon, Isabelle, Jason Weston, et al. (2002). “Gene selection for cancer classification using support vector machines”. In: *Machine learning* 46.1, pp. 389–422.

Hamilton, James D (1990). “Analysis of time series subject to changes in regime”. In: *Journal of econometrics* 45.1-2, pp. 39–70.

Hastie, Trevor, Jerome Friedman, and Robert Tibshirani (2001). “Model Inference and Averaging”. In: *The Elements of Statistical Learning*. Springer, pp. 225–256.

Hastie, Trevor, Robert Tibshirani, et al. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer.

Haury, Anne-Claire, Pierre Gestraud, and Jean-Philippe Vert (2011). “The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures”. In: *PloS one* 6.12, e28210.

Hoerl, Arthur E and Robert W Kennard (1970). “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1, pp. 55–67.

Hoffman, Matthew D et al. (2013). “Stochastic variational inference”. In: *The Journal of Machine Learning Research* 14.1, pp. 1303–1347.

Holtkamp, Derald J, James B Kliebenstein, et al. (2013). "Assessment of the economic impact of porcine reproductive and respiratory syndrome virus on United States pork producers". In: *Journal of Swine Health and Production* 21.2, pp. 72–84.

Holtkamp, Derald J, Dale D Polson, et al. (2011). "Terminology for classifying swine herds by porcine reproductive and respiratory syndrome virus status". In: *Journal of Swine Health and Production* 19.1, pp. 44–56.

Hoque, Nazrul, Dhruba K Bhattacharyya, and Jugal K Kalita (2014). "MIFS-ND: A mutual information-based feature selection method". In: *Expert Systems with Applications* 41.14, pp. 6371–6385.

Islam, Zeenath U et al. (2013). "Quantitative analysis of porcine reproductive and respiratory syndrome (PRRS) viremia profiles from experimental infection: a statistical modelling approach". In: *PloS one* 8.12, e83567.

Juloski, Aleksandar Lj, Siep Weiland, and WPMH Heemels (2005). "A Bayesian approach to identification of hybrid systems". In: *IEEE Transactions on Automatic Control* 50.10, pp. 1520–1533.

Kalousis, Alexandros, Julien Prados, and Melanie Hilario (2007). "Stability of feature selection algorithms: a study on high-dimensional spaces". In: *Knowledge and information systems* 12.1, pp. 95–116.

Keeling, MJ and L Danon (2009). "Mathematical modelling of infectious diseases." In: *British medical bulletin* 92.1.

Khalid, Samina, Tehmina Khalil, and Shamila Nasreen (2014). "A survey of feature selection and feature extraction techniques in machine learning". In: *2014 science and information conference*. IEEE, pp. 372–378.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Max Welling (2014a). "Auto-Encoding Variational Bayes". In: *stat* 1050, p. 1.

— (2014b). "Auto-Encoding Variational Bayes". In: *stat* 1050, p. 1.

Kingma, Durk P et al. (2014). "Semi-supervised learning with deep generative models". In: *Advances in neural information processing systems* 27.

Koene, Miriam GJ et al. (2012). “Serum protein profiles as potential biomarkers for infectious disease status in pigs”. In: *BMC Veterinary Research* 8.1, pp. 1–14.

Kostopoulos, Georgios et al. (2018). “Semi-supervised regression: A recent review”. In: *Journal of Intelligent & Fuzzy Systems* 35.2, pp. 1483–1500.

Kuhn, Max, Kjell Johnson, et al. (2013). *Applied predictive modeling*. Vol. 26. Springer.

Lee, Kyuyoung et al. (2017). “Unraveling the contact patterns and network structure of pig shipments in the United States and its association with porcine reproductive and respiratory syndrome virus (PRRSV) outbreaks”. In: *Preventive veterinary medicine* 138, pp. 113–123.

Lin, Hui et al. (2013). “Construction of disease risk scoring systems using logistic group lasso: application to porcine reproductive and respiratory syndrome survey data”. In: *Journal of Applied Statistics* 40.4, pp. 736–746.

Linderman, Scott et al. (2017). “Bayesian learning and inference in recurrent switching linear dynamical systems”. In: *Artificial Intelligence and Statistics*, pp. 914–922.

Machado, Gustavo et al. (2019). “Identifying outbreaks of Porcine Epidemic Diarrhea virus through animal movements and spatial neighborhoods”. In: *Scientific reports* 9.1, pp. 1–12.

Martinez, Stephen W (2002). *Vertical coordination of marketing systems: Lessons from the poultry, egg, and pork industries*. Tech. rep.

Mateu, E and Ivan Dıéaz (2008). “The challenge of PRRS immunology”. In: *The Veterinary Journal* 177.3, pp. 345–351.

Mendenhall, William, Terry Sincich, and Nancy S Boudreau (2003). *A second course in statistics: regression analysis*. Vol. 6. Prentice Hall Upper Saddle River, NJ.

Midi, Habshah, Saroje Kumar Sarkar, and Sohel Rana (2010). “Collinearity diagnostics of binary logistic regression model”. In: *Journal of Interdisciplinary Mathematics* 13.3, pp. 253–267.

Murphy, Kevin P (1998). “Switching kalman filters”. In.

— (2012). *Machine learning: a probabilistic perspective*. MIT press.

Murtaugh, Michael P and Marika Genzow (2011). “Immunological solutions for treatment and prevention of porcine reproductive and respiratory syndrome (PRRS)”. In: *Vaccine* 29.46, pp. 8192–8204.

Nassar, J et al. (2019). “Tree-Structured Recurrent Switching Linear Dynamical Systems for Multi-Scale Modeling”. In: *International Conference on Learning Representations (ICLR).*

Newman, Mark EJ (2002). “Spread of epidemic disease on networks”. In: *Physical review E* 66.1, p. 016128.

Nodelijk, G (2002). “Porcine reproductive and respiratory syndrome (PRRS) with special reference to clinical aspects and diagnosis: a review”. In: *Veterinary quarterly* 24.2, pp. 95–100.

Nodelijk, G et al. (2001). “A quantitative assessment of the effectiveness of PRRSV vaccination in pigs under experimental conditions”. In: *Vaccine* 19.27, pp. 3636–3644.

Otake, Satoshi et al. (2010). “Long-distance airborne transport of infectious PRRSV and Mycoplasma hyopneumoniae from a swine population infected with multiple viral variants”. In: *Veterinary microbiology* 145.3-4, pp. 198–208.

Paoletti, Simone et al. (2007). “Identification of hybrid systems a tutorial”. In: *European journal of control* 13.2-3, pp. 242–260.

Pena, Ramona N et al. (2019). “Genetic markers associated with field PRRSV-induced abortion rates”. In: *Viruses* 11.8, p. 706.

Phoo-ngurn, Phithakdet, Chanakarn Kiataramkul, and Farida Chamchod (2019). “Modeling the spread of porcine reproductive and respiratory syndrome virus (PRRSV) in a swine population: transmission dynamics, immunity information, and optimal control strategies”. In: *Advances in Difference Equations* 2019.1, pp. 1–12.

Probst, Philipp, Bernd Bischl, and Anne-Laure Boulesteix (2018). “Tunability: Importance of hyperparameters of machine learning algorithms”. In: *arXiv preprint arXiv:1802.09596.*

Ranganath, Rajesh et al. (2013). “An adaptive learning rate for stochastic variational inference”. In: *International Conference on Machine Learning*, pp. 298–306.

Rezende, Danilo Jimenez and Shakir Mohamed (2015). “Variational inference with normalizing flows”. In: *International Conference on Machine Learning*, pp. 1530–1538.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). “Stochastic backpropagation and approximate inference in deep generative models”. In: *International conference on machine learning*. PMLR, pp. 1278–1286.

Sanz, Hector et al. (2018). “SVM-RFE: selection and visualization of the most relevant features through non-linear kernels”. In: *BMC bioinformatics* 19.1, pp. 1–18.

Shamsabardeh, M et al. (2019). “A novel way to predict PRRS outbreaks in the swine industry using multiple spatio-temporal features and machine learning approaches”. In: *Frontiers in Veterinary Science* 6.

Shamsabardeh, Mohammadsadegh, Bahar Azari, and Beatriz Martıénez-López (2022). “Spatio-Temporal PRRS Epidemic Forecasting via Factorized Deep Generative Modeling”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3978–3982.

Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin (2018). “A comparison of ARIMA and LSTM in forecasting time series”. In: *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, pp. 1394–1401.

Smith, Katherine F et al. (2014). “Global rise in human infectious disease outbreaks”. In: *Journal of the Royal Society Interface* 11.101, p. 20140950.

Sontag, Eduardo (1981). “Nonlinear regulation: The piecewise linear approach”. In: *IEEE Transactions on automatic control* 26.2, pp. 346–358.

Stoica, P. and Y. Selen (2004). “Model-order selection: a review of information criterion rules”. In: *IEEE Signal Processing Magazine* 21.4, pp. 36–47. DOI: 10.1109/MSP.2004.1311138.

Tibshirani, Robert (1996). “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Tumpey, Terrence M et al. (2005). “Characterization of the reconstructed 1918 Spanish influenza pandemic virus”. In: *science* 310.5745, pp. 77–80.

Vanderhaeghe, C et al. (2013). “Non-infectious factors associated with stillbirth in pigs: a review”. In: *Animal reproduction science* 139.1-4, pp. 76–88.

VanderWaal, Kimberly and John Deen (2018). “Global trends in infectious diseases of swine”. In: *Proceedings of the National Academy of Sciences* 115.45, pp. 11495–11500.

Zhu, Xiaojin and Andrew B Goldberg (2009). "Introduction to semi-supervised learning". In: *Synthesis lectures on artificial intelligence and machine learning* 3.1, pp. 1–130.