2018

Application of Machine Learning on Fracture Interference

Dennis Wayne Chamberlain Jr.
West Virginia University, dchambe1@mix.wvu.edu

Application of Machine Learning on Fracture Interference

Dennis Chamberlain

Thesis submitted
to the Benjamin M. Statler College of Engineering and Mineral Resources
At West Virginia University

In partial fulfillment of the requirements for the degree of

Masters of Science in
Petroleum and Natural Gas Engineering

Ali Takbiri-Borujeni, Ph. D., Chair

Samuel Ameri, M.S.

Ebrahim Fathi, Ph. D.

Ming Gu, Ph. D.

Department of Petroleum and Natural Gas Engineering

Morgantown, West Virginia

2018

Keywords: Machine Learning, Artificial Intelligence, Fracture Interference

Abstract

Application of Machine Learning on Fracture Interference

Dennis Chamberlain

A method has been developed that locates and determines well-to-well hydraulic fracture interference (frac-hit) in shale plays using hard data. This method uses Artificial Neural Networks (ANN) with designated parameters and target outputs in conjunction with graphs of gas flowrate, tubing pressure, and cumulative gas prediction. The method was created to address the significant increase in frac-hit occurrences due to the infill wells being completed in shale plays. The production data of the well is first cleaned to eliminate outliers in the initial timeframe of the well and periods of no production so that the ANN model can be accurately trained. The model then predicts daily gas flowrate and is graphed against the wells cumulative gas and tubing pressures. The location of the section of variance from real data versus the predicted results will indicate a phenomenon at a given instant. This can indicate frac-hits through graphing a plot of a parent wells tubing pressure, gas flowrate, and cumulative gas production against a new child well at the location of variance that was observed in the model prediction.

The results of ANN training and test results accurately predicted cases where frac-hits are observed in the given field. This model also was able to predict the onset of the frac-hit which correlated to the same time that a new well was being completed in the area. This method allowed further research into the results since it was able to provide predicted flowrates at the time periods of frac-hits rather than only the time of the hit. Therefore, the ANN model was determined to be an adequate choice in analyzing frac-hits due to the sheer volume of information that can be taken away from the results.

Table of Contents

Introduction

Since the Range Resources completed the Renz 1 Well in the Marcellus Shale Play over a decade ago, many improvements in drilling, production, and completion methods have been made in order to increase production and profits in the field. Countless techniques have been created to optimized well spacing, lateral length, stage spacing, and countless other parameters through created equations; but as a focus shifts to use these methods, other issues may arise such as the case of hydraulic fracture interference (frac-hits).

Frac-hits are communication between wells that have been caused by the hydraulic fracturing treatment performed on a nearby well (Jacobs 2017). This phenomenon is due to the operating companies trying to maximize recovery from the shale assets by decreasing well spacing which increases the number of occurrences of frac-hits. For example, in 2010 many shale wells were drilled with well spacing 1,000 feet apart. Today, many of these designs range from 500 feet to 250 feet while also increasing the completion jobs in an attempt for the infill well to drain the as much of the available reservoir as possible. The completion of these infill wells can cause effects on the parent well. In 2017, Rassenfoss stated that previous research showed these effects have been recorded in upwards of a mile away through a parent well seeing a pressure spike of 10psi. However, it is more common to see the effect in a neighboring well. This is due to the hydraulic pressure used to fracture new wells since fractures typically grow towards zones where pressure has been depleted through production which is caused by the effect of a stress vortex (Jacobs 2017). The impacts of these effects can either be seen immediately following the completion, or over time as pressure through investigation of the wells production and pressure history. These occurrences of frac-hits typically cause a negative impact on the overall production performance of a wells life. For example, in 2017 Range Resources stated that wells in a shale asset that were underperforming due to frac-hits to offset production amounted to a financial loss of up to 75 million dollars per day (Jacobs 2017). Therefor methods of preventing or limiting frac-hits have the potential to save companies billions of dollars over the course of a fields lifetime. However, prior to preventing this phenomenon, it must first be consistently detected.

While frac-hits have been detected through investigations of pressure, flowrates, and models, few recent studies have been completed an in-depth study on this topic.

Generally, frac-hits have been detected by monitoring the sharp changes in the pressure and flow rate, in parent wells (existing producing wells), during the stimulation and production period from the child wells (new wells completed) through graphical visualizations of field data and some computerized model creation (Esquivel et al. 2017). This approach becomes complex as number of wells in the field increases and it becomes difficult to deconvolute the combined effects of child wells on the parent wells. In the absence of efficient tools and methodologies for frac-hit detection, optimum recovery of hydrocarbons form shale resources remain unlikely since these events can cause decline in production rates.

In 2015 a procedure was created to identify and quantify interference from pressure response observations with a shut in well and changing rates with neighboring wells in a multi-well pad (Awada et al. 2016). Interference was determined and classified through use of buildup tests, rate and pressure vs time plots, interference diagnostics, interference time, and history matching through the use of models (Esquivel et al. 2017). This analysis of hydraulic interference was completed through use of production data. Their study used parent and child wells in neighboring areas to evaluate production rates and determine the effect and corresponding magnitude between frac-hit cases, which vary from field to field. Also, mathematical models were created to assess the degree of interference in multi-well pads due to frac-hits where numerical simulations created varying frac-hit situations to determine the accuracy of the model (Molina et al. 2017). These studies used different methods of analyzing frac-hits but use various assumptions to create and analyze their data.

Recently, artificial intelligence (AI) has been becoming more prominently used to reduce human bias, assumptions, and errors in cases like those mentioned above. For example, AI has been used to aid in predicting average oil flow rates in multilateral wells (Buhulaigah et al. 2017) through use of surface and reservoir parameters. This AI model was able to predict oil flow rate with an error of around 8% while previous models from Borisov's correlation averages an error of greater than 50%. Also, in 2017 Amer et al. used artificial neural networks (ANN) to determine ROP predictions in the Nile Delta Area. Lithology changes, drilling parameters and bit data were the inputs to predict the ROP changes and were able to accurately predict this with an accuracy of 90% in the training section. In 2018, Zborowski noted the concerns and efforts of Baker Hughes, a GE company(BHGE). This stated that this phenomena is becoming a primary concern and that a race of innovation has begun for solutions to this issue with BHGE further advancing the use of machine learning to diagnose fracture interference. This was completed using input data such as pumping schedules, well spacing, and proppant concentrations to predict time-series pressure data on offset wells. Specifically, this method uses clustering analysis to determine why frac hit occurs but is currently not capable of determining the hit at the time of occurrence. These prior works demonstrations that machine learning is a viable method to use to further investigate fracture interference.

Artificial Neural networks (ANN) provide an accurate method where the data is analyzed, weights are assigned to each input parameter, and through the weights an output of your choosing can be determined, which in this case is daily gas rate. This is chosen due to the production being noted as the most impacted and observed through the previous referenced studies on fracture interference.

AI provides a perfect base for analyzing frac-hit issues in the field while reducing the number of assumptions made in the process and decreasing the time needed to determine a frac-hit has occurred. In this work, ANN are applied to a dataset provided by an operator in Marcellus Shale (more than 200 wells) to validate a technology that identifies a parent wells instance of frac-hit as well as a localized time period when this occurs. Therefore, child well affecting the production of the parent well can be identified and the stimulation and design parameters can be accessed

Artificial Neural Network Background

As society begins to turn to machine learning to address complex situations, it is important to understand the basic concepts of how this process works so the potential in the oil and gas industry can truly be understood. Machine learning with ANNs can be described as a way of determining a solution without have set methods and equations to determine the outcome.

Generally, there are many varying forms of ANNs that perform similar task with some technical differences with most using the same basic concept. The process begins by the selection of input parameters, such as tubing and casing pressures for oil and gas purposes or the distance it takes to stop a vehicle when traveling at a set speed and the amount of tread left on the tires for an automated vehicle that will be used to determine the output prediction. These parameters must not be correlated to one another so that an accurate model can be created (Figure 1) as a heavy correlation effects the models ability to train itself accurately on the data. Each line from the parameter to the hidden layer represents the path of information to the next section. The hidden layer, the location of neurons, is where the parameters information is sent to form an approximate function with weighted values that will be used in output prediction. The function and weights constantly vary between number of parameters selected, number of hidden neurons chosen, and the output that is being predicted. This is to ensure that the best prediction is made based of the given data. On a case to case bases a grid search is performed on the model, which the hidden layer's approximation and weights are iteratively changed as the model attempts to form an equation with the best accuracy between the predicted and known output. Once the user determined accuracy has been achieved, the model is saved and can be used again to predict outputs of similar cases with the same input parameters included.

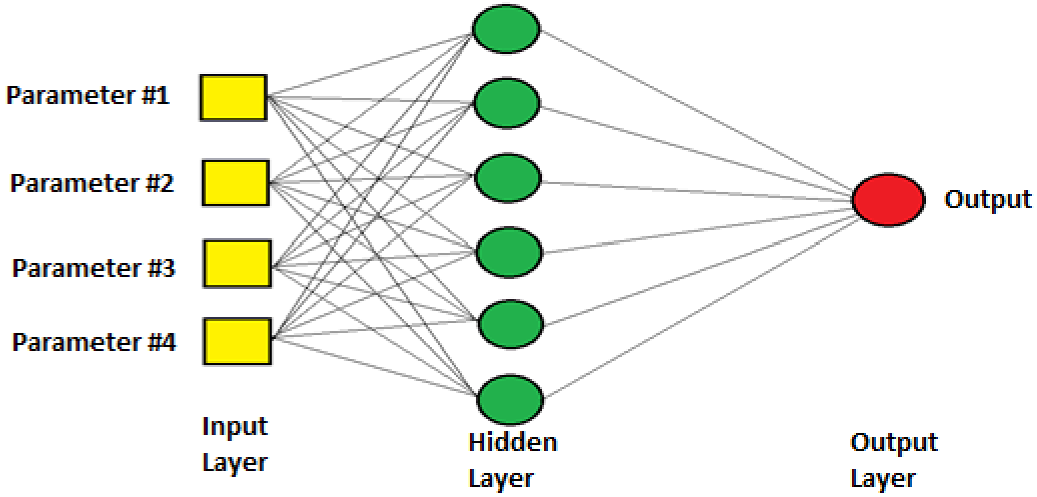


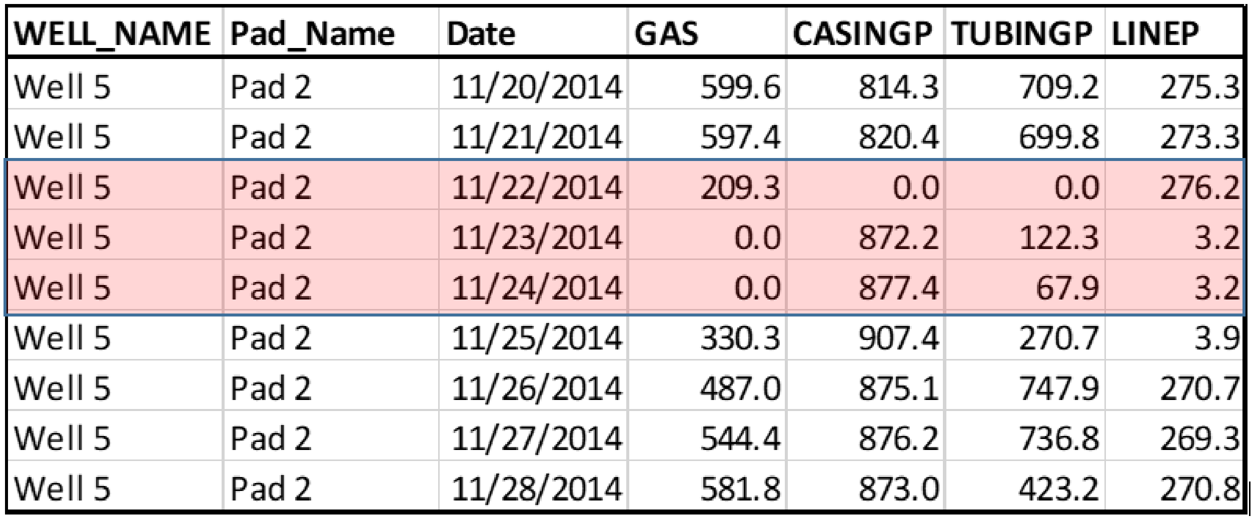
Figure 1. Artificial Neural Network Layers

Methodology

Data Mining and Sorting

In order to create a predictive model, the input and output data needed to be formatted and recorded correctly. The field data that was received consisted of daily production rates, operational, as well as some completion data for each well. This production data consisted of gas, water, and condensate production. The operational data contained casing, tubing, line, and sandface pressures. The field data in its raw format included various days with zero production along with anomalies within the data compared to the local average. This type of data, if trained within the neural network, could cause poor training of the mode. Therefore, this issue had to be resolved prior to sending the data through the created neural network.

Table 1. Raw Field Data



WELL_NAME	Pad_Name	Date	GAS	CASINGP	TUBINGP	LINEP
Well 5	Pad 2	11/20/2014	599.6	814.3	709.2	275.3
Well 5	Pad 2	11/21/2014	597.4	820.4	699.8	273.3
Well 5	Pad 2	11/22/2014	209.3	0.0	0.0	276.2
Well 5	Pad 2	11/23/2014	0.0	872.2	122.3	3.2
Well 5	Pad 2	11/24/2014	0.0	877.4	67.9	3.2
Well 5	Pad 2	11/25/2014	330.3	907.4	270.7	3.9
Well 5	Pad 2	11/26/2014	487.0	875.1	747.9	270.7
Well 5	Pad 2	11/27/2014	544.4	876.2	736.8	269.3
Well 5	Pad 2	11/28/2014	581.8	873.0	423.2	270.8

This problem was solved using a created Python script that eliminated rows of daily data if the production values didn't meet the selected criteria. For example, all attributes within the raw dataset that have missing or zero valued data, due to shut in periods or erroneous input, resulted in the row being eliminated from the data set. The outlying data, due to erroneous user input, used a different approach for removal. For this data each wells' parameters are graphed against time and the outliers are visually identified and removed do to the data's limited amount of outliers. To ensure no other outliers are missed, the data is cleansed again by using a rolling average of the parameters values on a well basis and eliminating any points that fell outside of the users given threshold of 30% above or below the current rolling average of the data.

Also, to ensure no other outliers are present in the data, a method of data cleansing known as Z-score is used which is similar to the rolling average used above. This method measures that significances of the input parameters to identify which of the values are within the acceptable range of values. The identified values that are outside of the given range can then be marked as outliers. This flagged outliers can then be confirmed by graphing the data in a cross plot against other input parameters as well as the target output parameter to ensure the correct selection has been made. The outliers seemed to be limited as the data was recorded on a daily basis and was prescreened by the company which provided the data for the study.

These processes are performed due to the limitations of ANN training and it ability to handle missing, incomplete, or erroneous data (outliers). The elimination of invalid data rows allows the NN to properly develop trends within the data for accurate predictions. If the data would be left in the study, the model would create inaccurate weights to the variables causing a higher error within the test section of the results.

Table 2. Cleaned Field Data

WELL_NAME	Pad_Name	Date	GAS	CASINGP	TUBINGP	LINEP
Well 5	Pad 2	11/20/2014	599.6	814.3	709.2	275.3
Well 5	Pad 2	11/21/2014	597.4	820.4	699.8	273.3
Well 5	Pad 2	11/25/2014	330.3	907.4	270.7	3.9
Well 5	Pad 2	11/26/2014	487.0	875.1	747.9	270.7
Well 5	Pad 2	11/27/2014	544.4	876.2	736.8	269.3
Well 5	Pad 2	11/28/2014	581.8	873.0	423.2	270.8

Data Normalization

Once the data mining and sorting process is complete, the last step is to transform the raw data. For this process, Feature Scaling is used to reduce the dimensionality of the values of data. The minmax scaling process is used to scale the data between zero and one.

$$V_{Transformed} = \frac{V_{initial} - V_{min}}{V_{max} - V_{min}}$$

Equation 1. MinMax Scaling

This process allows multiple advantages over simple leaving the data as is. First, this allows the ANN training to be performed more quickly than without scaling while minimizing the risk of the model being stuck in a local maxima of optimization, therefore hampering the chances of the model reaching the optimal training. However, this is not the driving factor of normalizing data. Most ANN systems implicitly assign greater weight to features with large ranges rather than those with small ranges, therefor Feature Scaling is essential so that all parameters of the neural network will have the same range of scaling from 0 to 1. Thus ensuring they have the capability to equally contribute to the final weights of the ANN (Aksoy et al 2001). The results will then be scaled back to the original parameters once the modeling, training, testing, and prediction has been completed to allow proper visualization of the model.

Preprocessing

Once all data has been organized, cleaned, and scaled, feature selection can be performed. Feature selection are measures used to evaluate the intended outcome of the research, in this case frac-hits. For instance, the data had some categories of data, multiple missing section, some with low variance, and low impact, which would result in the neural network results to be inaccurate. Therefore, the parameters to be excluded are determined by determining the variance of each attribute and when minimal impact was suspected, the parameter would be removed from the study.

Since the data is trained on a well by well basis, the reservoir property data, baring a change in completion, was consistent for each of the daily values. Therefore, is was removed with a low variance feature selection tool which removes all feature whose variance doesn’t meet a given threshold or has no variance using the following algorithm.

$$Var[X] = p(1 - p)$$

Equation 2. Feature Selection

The sections with missing data are handled differently than the above variance method. For these cases the parameter was removed if it did not contain more than 90% of the data on a daily basis for the life of the well that is being trained and tested.

Supervised Neural Network Setup (ADAM)

To train the NN certain distinctions are made to ensure accuracy and limit the training time of the model. Due to the large amount of data that was available a decision of what type of ANN (supervised or unsupervised) as well as the solver was needed. Supervised methods such as adaptive moment estimation (ADAM), quasi-Newton methods, stochastic gradient descent, as well as unsupervised methods such as KMeans and Density Based Spatial Clustering of Applications with Noise, also known as DBSCAN are considered and the options are weighed. The unsupervised methods (DBSCAN and KMEANS) would use cluster analysis to identify patterns within the data and allow groupings to be formed. DBSCAN's process is for each data point to be identified in a cluster and given a radius that contains as least an (x) minimum number of points. The shape of this neighborhood is determined by the choice of a distance function between the two points which is the main group classifier for the process. KMeans methods allows the user to define a set number of clusters that the program will group all of the given parameters into, therefor labeling combinations of parameters into its determined set cluster. However, with the number of data points used in the study, as well as the complexity of this information, these methods would create false positive instances of frac-hits when clustered. This caused for further insight into supervised methods for the study.

Quasi Newton methods have been known to work well with smaller datasets, less than 1000 data points, so due to the size of data being presented, it is not a feasible option for the study. Stochastic gradient decent updates weights in the ANN based on minibatches of data rather than the entire batch (or given training set) and can be very used since it is computationally less demanding than other methods for cases where computer performance could limit the outcome of the study(Stanford 2018). However, since this was not a limiting factor, adaptive moment estimation (ADAM) ANN solver was used. ADAM (Kingma et al 2015) is an algorithm for first order gradient-based optimization of stochastic objective functions which adaptively estimates lower order moments which runs quicker and more accurately on large datasets than any other current method. This was selected since the dataset contained over 10,000 rows of time-based data. Based on the size of the data in the research, ADAM could in theory provide higher accuracy, as well as more precise results than stochastic gradient descent or quasi-Newton methods.

Once the solver is selected, the specifications of the NN can be set. These specifications include the number of neurons, activation function, learning rate, maximum number of iterations(epochs), and the tolerance for the optimization. This number of neurons is reference to the number of neurons within the hidden layer of the ANN. This layer of neurons is where the approximate function with weighted values that will be used in output prediction is formed. Since there are always various numbers of input parameters used within a NN, there is no set number of neurons that should be used for this layer. However, from experience, there should be at minimum one neuron for each of the input parameters in the data. Past this, the number of neurons to be used comes down to user preference at their experience with ANNs and their dataset which lead to the selection of one and a half times the input parameters.

Next, the activation function is chosen. The chosen function aids in the output of the neurons by molding the results to the selected function. In this case, rectified linear units classification is used. The method works by thresholding values at 0 using the following equation.

$$f(x) = max(0, x)$$

Equation 3. Activation Function

Therefore, if an output is less than 0, the function will output 0, when the model output is greater than 0 it outputs a linear function from zero to one (Agarap 2017). One is used for the maximum due to the normalization of the data which was discussed above.

Next, the learning rate is selected. The learning rate is a crucial hyper-parameter that controls the NNs' performance, as if the rate is too low, it hampers the speed of training causing a large increase in model runs. However, if the rate is too large, the training is accelerated but at the high risk of causing the ANN to diverge from the actual data (Ren et al. 2018). Similar to the number of neuron selection this value has no set universal standard and is open to interpretation of how the user would like the model to be implemented.

Finally, the maximum number of iterations and tolerance of the mode are selected. Tolerance is used in the optimization of the model. When the loss does not improve for a given number of iterations by the specified tolerance, the model is considered converged and training is complete. If continues to improve, the maximum iteration number is the only stopping point of the model. This ensures two scenarios are avoided. One being the model does not continue training when the improvement at such a small level that the results will not show gain any value. The second is when best model is created prior to the maximum iteration the model does not continue when no improvement will occur.

Once the solver and is specifications are selected an ANN model can be trained. The ANN method used for this training and test set is called backpropagation. Backpropagation is used to compute partial derivatives of the cost function with respect to any weight or bias of the ANN. These weights or bias are then updated based on the results of the results of the training and test data accuracy. This process calculates the error contribution of each weight for each iteration of data. The weights are then updated and the training is ran constantly through this process until the training data is unable to reach a higher accuracy without hampering the output of the test data. For example, as long as the training and test accuracy is increasing, the model continues to iterate and learn the patterns, storing the updated weights, the moment the test dataset accuracy begins to decrease, the previous iterations weights are used as the final results of the NN training.

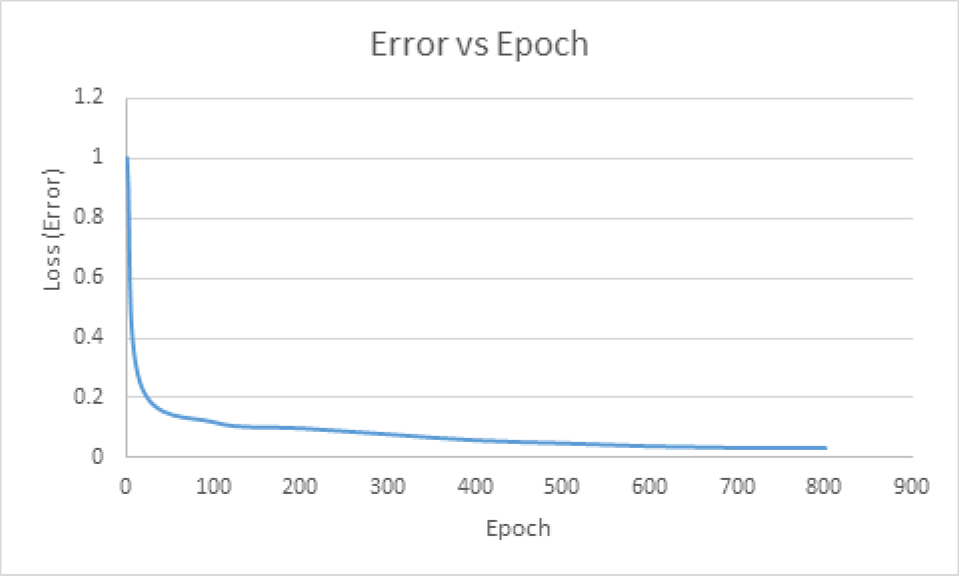


Figure 2. Error vs Epoch example

The training set of data consists of 40% to 50% of the data on average while the test dataset is composed of 10%, and the final 40% to 50% of the data is saved for use in model predictions. The training and test dataset can be composed in one of two ways. The first method is to allow the model to choose data points (taken from the same time period) at random from the entire dataset, leaving the remaining to be designated as prediction data. However, this would not allow for proper testing of a frac-hit due to the broken-up time frame of the data. Therefor the second method must be used. This method takes the 40% of the training dataset from the earliest 40% of the data, the following 10% for test, and the final 50% for prediction. For example, if there is 10 years of data from 2000 to 2010, years 2000 through the end of 2004 would be training, 2005 would be used for testing, and 2006 through 2010 would be used for prediction. Once the data is separated in the above fashion and the training and testing is complete, the model can be evaluated to determine if the accuracy is high enough to continue the process and move on to prediction of the dataset.

The test results, assuming the training set was accurately calibrated, should show similar predictions as the output values recorded in the dataset and have an accuracy of 80% or higher. This signifies an accurate model creation that can be used to predict the final 50% of the data. Once the model shows adequate training and test results and the outputs are predicted; the results are saved, and the process is repeated until 100 ANN models have been created. This means the above process is completed 100 times to inspect how the prediction of the model is affected by a random initial sequence.

The process of running the ANN multiple times is essential for analyzing the models capability of producing reproducible results. Each run of the ANN beings with a random selection of initial weights. This random initial selection causes the training and testing of the ANN to vary for each run. This is due to the random weights applied to the model prior to the first epoch of training. These weights, in turn effect the models accuracy when comparing the results to that actual field measurements. In some instances, a poor initial selection of weights for the parameters can cause the model to have a low accuracy. If the model is only be ran once, and this poor initial weighing would occur, the results for the case would be poor and it could not detect if a frac-hit could have occurred Therefor, by running the model 100 times, the results show the effect of the random initial selection of weights on the model, as well as proving the model is capable of providing consistent, accurate results. With 100 ANN runs completed for each well, the mean of the models' runs are taken to check the models average consistency prior to moving to the next step of the study.

The mean of predictions made by these 100 models are used as the final prediction of flowrate for the well. This is done to increase the accuracy of the models since each model begins with slightly varying weights for each neuron, meaning that while accuracy may be similar, the weights can slightly differ. Therefore, taking the mean of the 100 different models can allow for consistent results, reducing the effects lower accuracy models and further validating the accuracy of neural networks.

Results (ANN)

Case 1: No Frac-hit Observed.

First, to ensure that the NN is correctly performing, a test case was ran on a well that was expected to have no frac-hit occurrences. This was confirmed by determining that the well was not located near other wells in the relative area. This well is located in the Marcellus Shale and is a horizontal well without nearby wells in the surrounding area. First, the data was split according to the specifications made within the methodology. Since the wells are analyzed on a case by case basis, the distance calculations are removed from the input parameters, seeing that they would be constant for the life of this well. The timeframe of data available for the well was from the beginning of production until 1750 days meaning 700 days are used for training, 175 days for test and the remaining data was used to compare the predictions to the real field data. Through 100 models of ANNs, results had an R-Squared combined training and test value of 98%. The results are able to confirm this as the predictions of all 100 models have a tight grouping to the field data (Figure 4). Taking the mean values further confirms that tis methodology can accurately predict the production data (Figure 5)

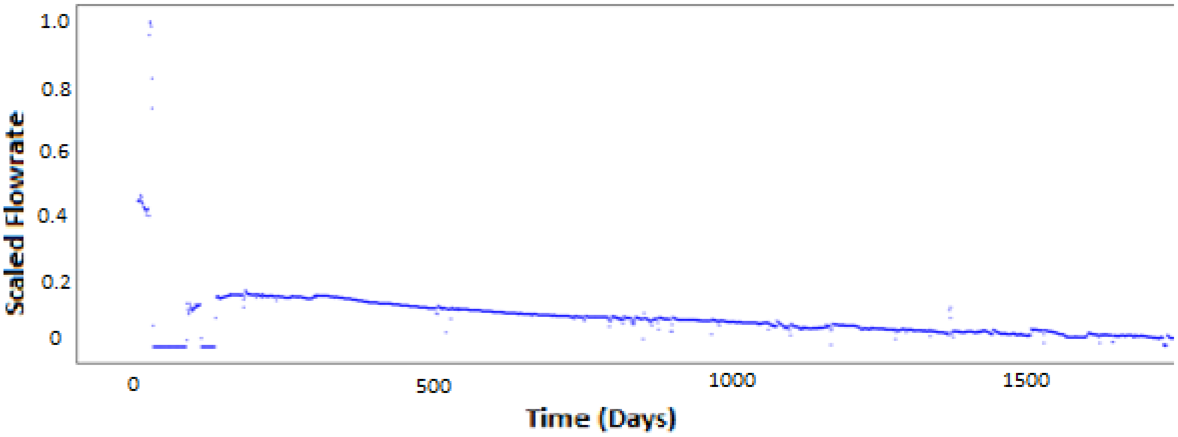


Figure 3. Scaled flowrate vs time before ANN NN has ran

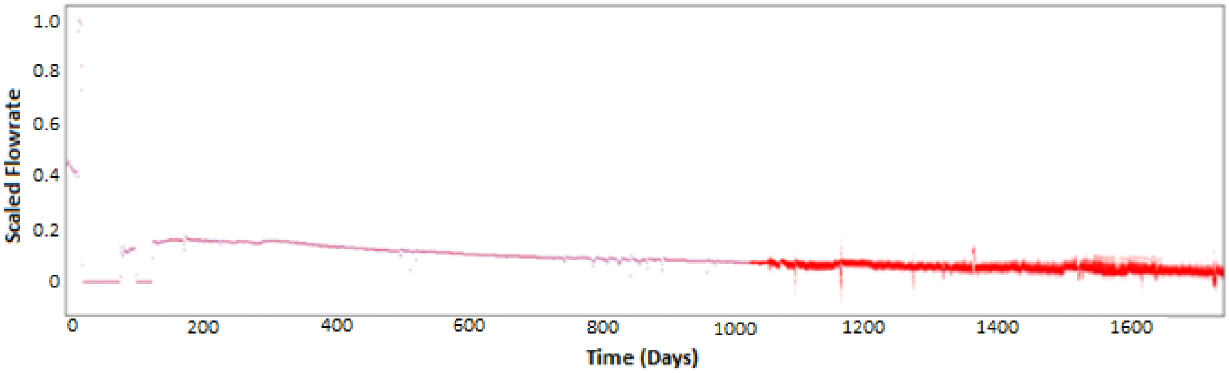


Figure 4. No Frac hit case. ANN results of 100 model runs.

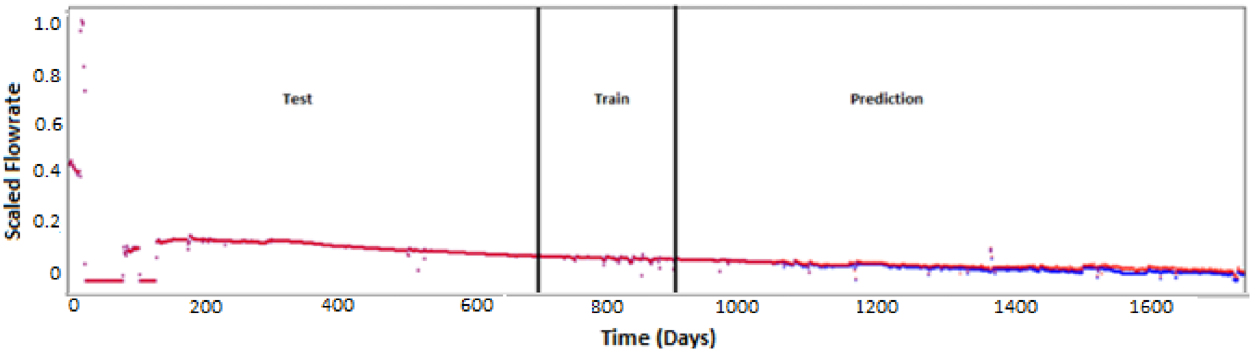


Figure 5. No Frac-hit Case Averaging Run. The red points represent the models' predictions while the blue represent the field data

Although no frac-hits are observed, a plot of the tubing pressure, gas flowrate, and cumulative production is graphed as a confirmation of model (Figure 6). There are no wells completed in the local areas around the well that have caused pressure or rate fluctuations that would indicate a frac-hit therefore the only well shown in the graph is the well that underwent ANN predictions.

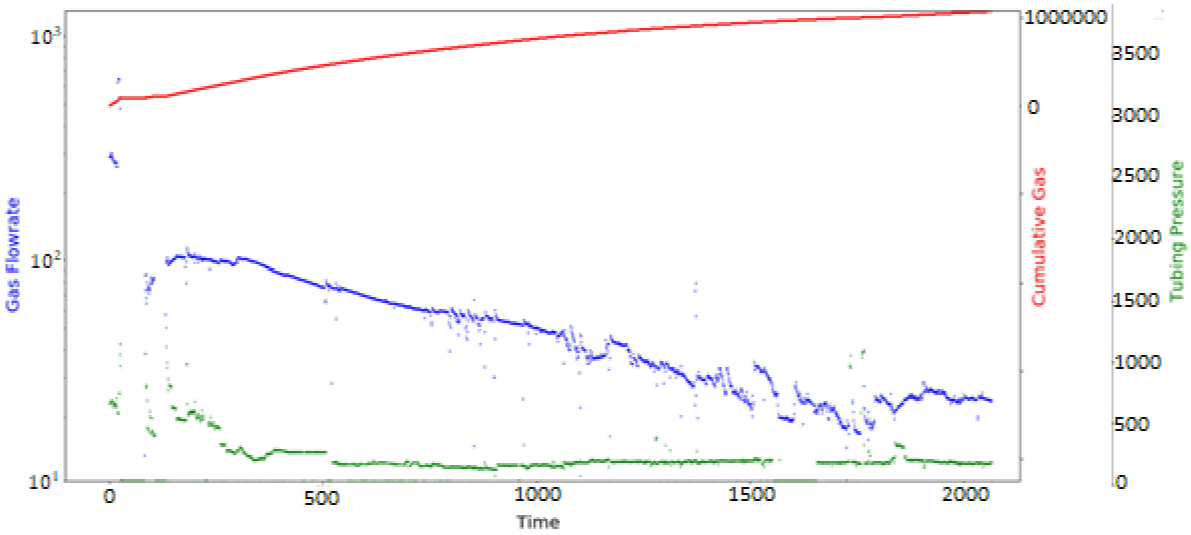


Figure 6. Case One. plot of gas flowrate, cumulative gas, and tubing pressures vs Time

Case 2: Frac-hit Occurrence

Case 2 is located in the Marcellus Shale in a neighboring area to the well in case 1 and is a horizontal well with nearby wells in the surrounding area which have all been drilled and competed after the well in this case. Since the wells are analyzed on a case by case basis, the distance calculations are removed from the input parameters, seeing that they would be constant for the life of this well. The timeframe available is from the beginning of production to 1412 days. This means the data is split into 640 days of training data, 160 days of testing data, and the remaining days are used to test the model's prediction accuracy. The training and testing R-squared value of the predictions made by 100 models are determined to be 80% between the created model and the real data's training and test sections. The mean of the models' predictions gave a clear representation of the timeframe of fracture interference (Figure 8).

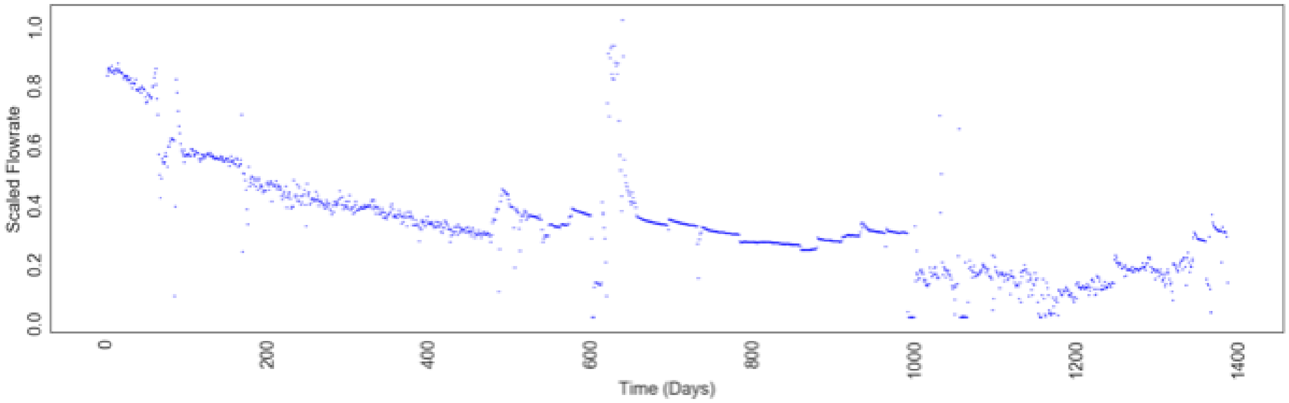


Figure 7. Initial Scaled Flowrate

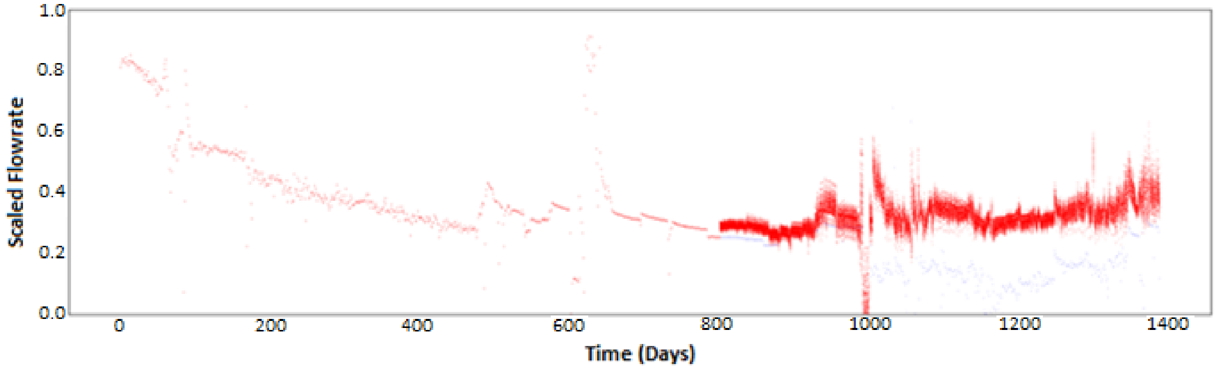


Figure 8. Frac-hit with 100 models' predictions

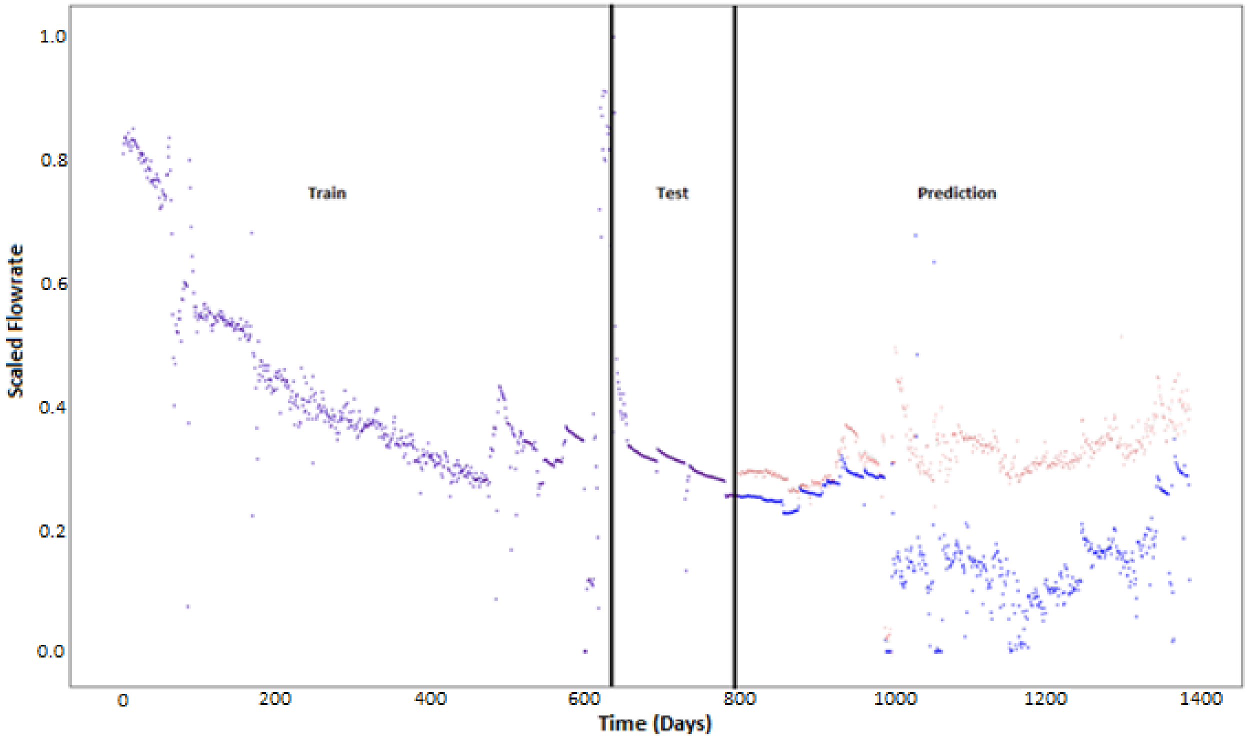


Figure 9. Frac-hit Average Run. The red points represent the models' predictions while the blue represent the field data

The predictions closely follow the field data fairly well until the time frame of 990 days where soon after there was a significant decline in the gas production while the model predicted higher values with the given input parameters. Further investigation into the production and completion data of the well at this time period lead to no evidence of why this occurred at this time period other than a possible frac-hit. Therefore, the code begins to look for new wells that was completed nearby between the timeframe in question. Thus, a plot is created to show the modeled well (parent) against the child well (Figure 11).

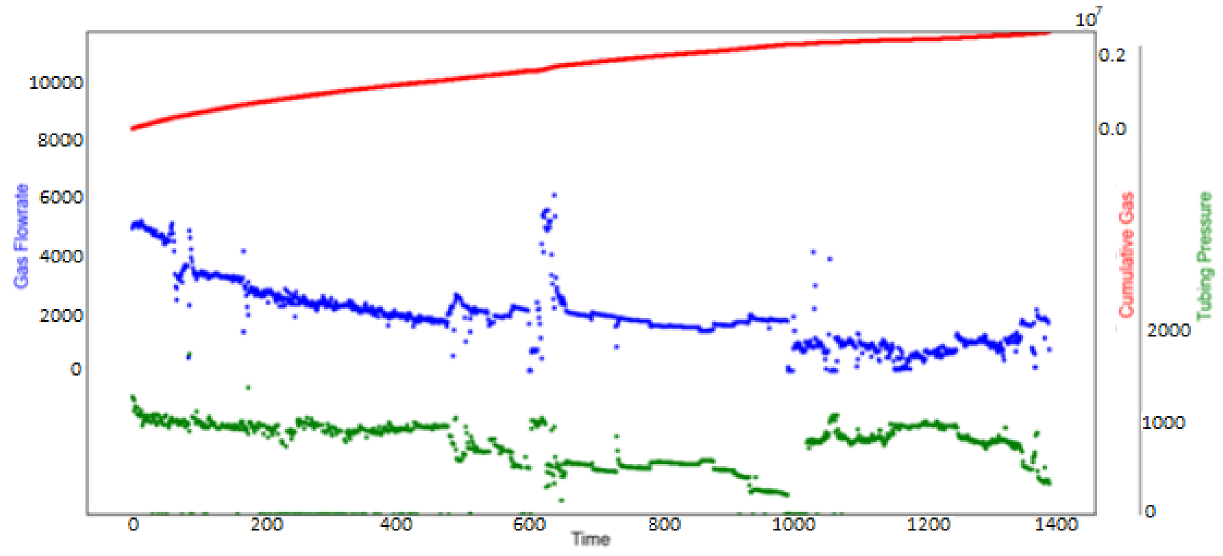


Figure 10. Parent well with Gas flowrate, tubing pressure, and cumulative gas

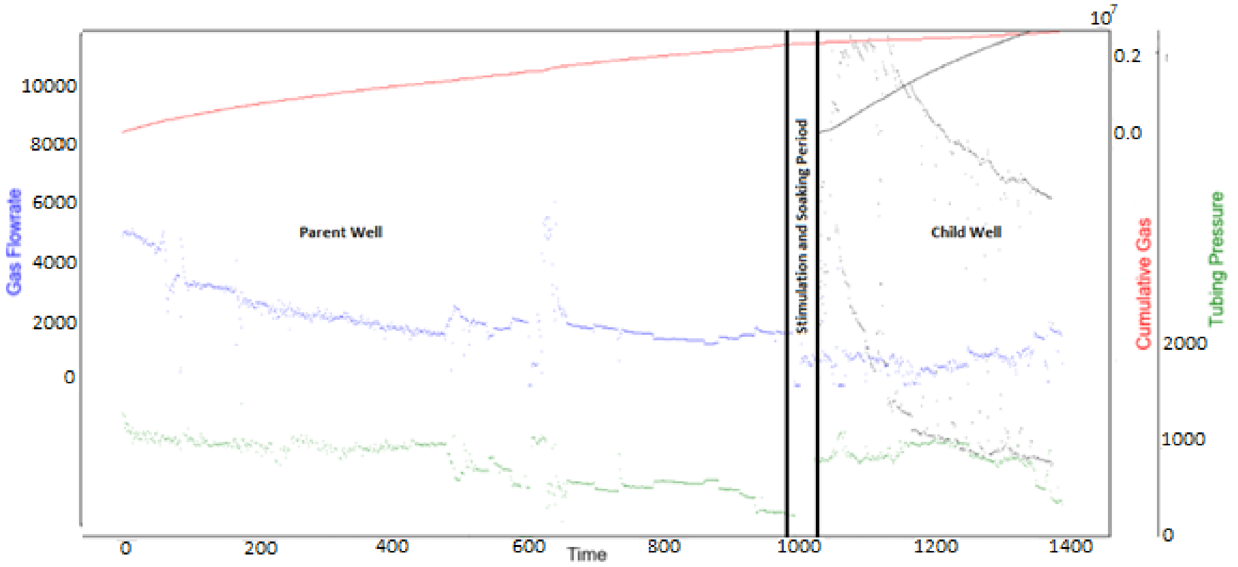


Figure 11. Frac-hit Parent Well vs Child Well. The child wells cumulative production, gas flowrate, and tubing pressures are all represented by black lines.

Through the graph it can be seen that the child well stimulation has an effect on the parent well, signifying that a frac-hit has taken place. This occurs around day 990 and a drop in the parent's wells production is immediately seen. The ANN prediction directly correlates to the stimulation of the child well. This confirms that the parent well is in communication with the child well at this point and aids in the validation of the model’s ability to predict a frac-hit location given the pressure and water production of a well.

With identification of frac-hit within the above well, wells with within the same pad are observed to determine if this effected one well, or the frac-hit shows an effect beyond its neighboring well. The timeframe available is from the beginning of production to 1740 days. This data is once again split into training data, testing data, and prediction sections. The model is ran 100 times and the results are plotted overtop one another. Once this process is complete, the mean of these runs are taken and plotted to determine the models outcome. (Figure 14).

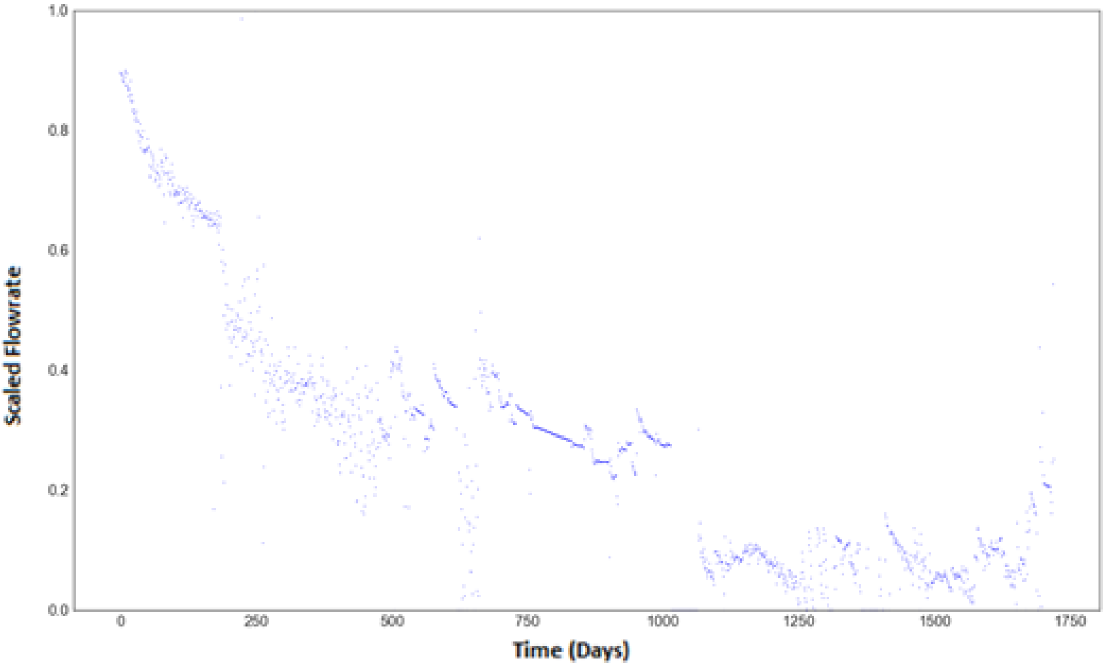


Figure 12. Base Scaled Flowrate

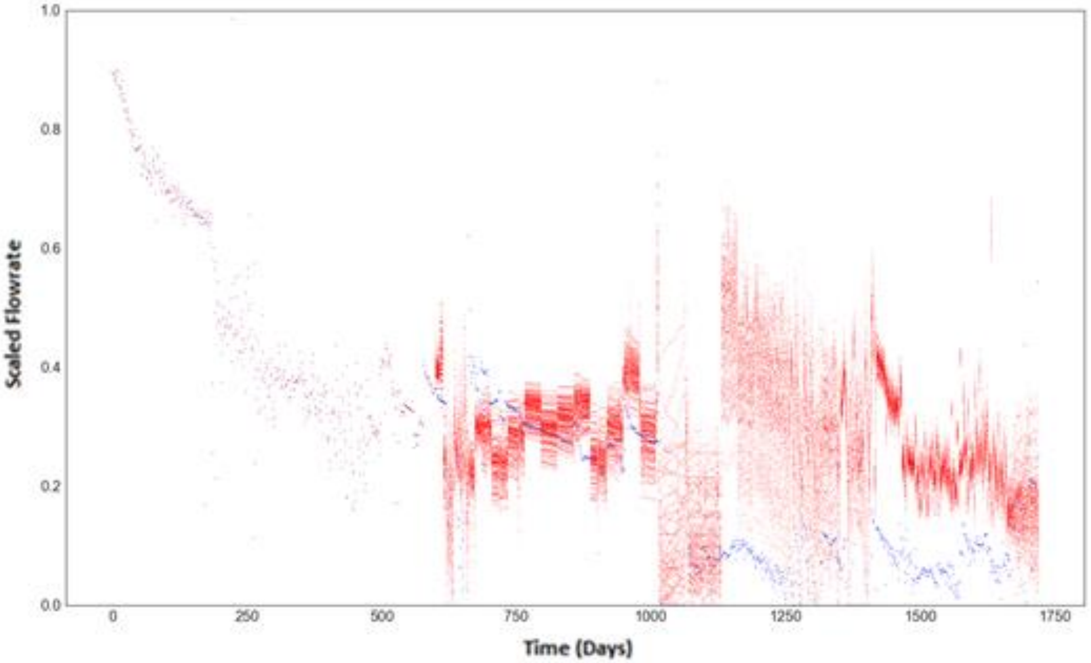


Figure 13. 100 runs of Model results

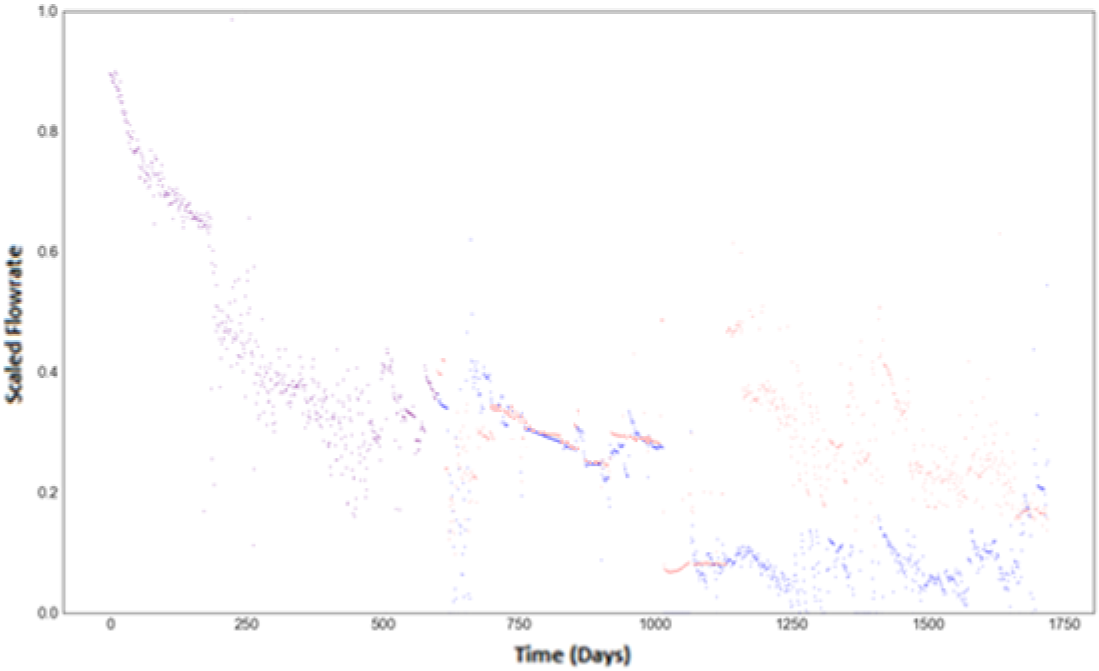


Figure 14. Average of Model runs

The predictions closely follow the field data fairly well until the time frame of 1150 days where soon after there was a significant decline in the gas production while the model predicted higher values with the given input parameters. Further investigation into the production and completion data of the well at this time period lead to no evidence of why this occurred at this time period other than a possible frac-hit. Therefore, a plot is created to show the modeled well and wells in the surrounding area to confirm the results.

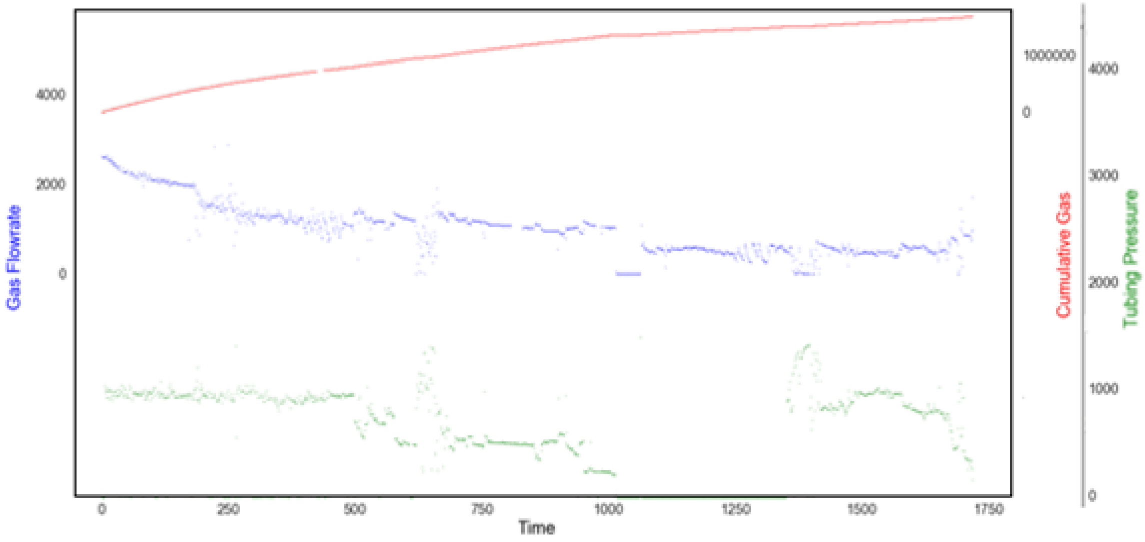


Figure 15. Case 2 parent well with Gas flowrate, tubing pressure, and cumulative gas

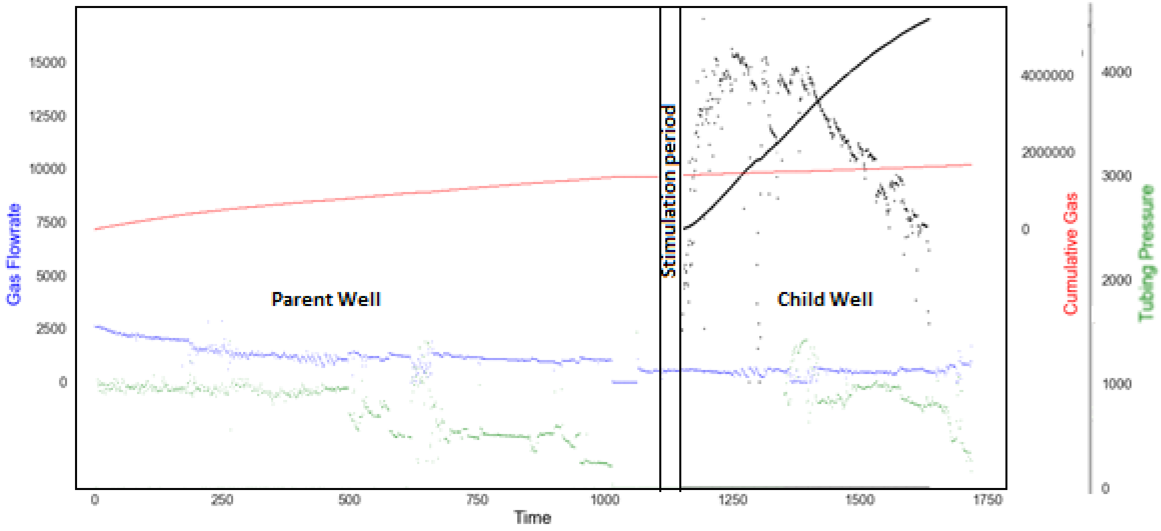


Figure 16. Case 2 frac-hit Parent Well vs Child Well. The child wells cumulative production, gas flowrate, and tubing pressures are all represented by black lines.

Through the graph above as well as a detailed look at Figure 14 it can be seen that the child well stimulation has an effect on the parent well, signifying that a frac-hit has taken place. This occurs around day 1150 and a drop in the parent's wells production is seen shortly after. The ANN prediction directly correlates to the stimulation of the child well. This confirms that the parent well is in communication with the child well at this point and aids in the validation of the model's ability to predict a frac-hit location given the pressure and water production of a well.

Summary and Conclusions

In this work, a methodology is introduced for early detection of hydraulic fracturing interference (frac-hits) in a shale asset. This includes a variety of data preparation methods such as data cleansing for shut-in periods and outlier removal methods such as Z-score. Once completed, the features with a low variance from one another, were removed from the study to increase the accuracy of the NN. At this point, various neural networks and their solving methods were reviewed until supervised methods with the ADAM solver were determined to best represent early cases of frac-hit. The results were then analyzed to determine if the use of the methodology and methods discussed could detect frac-hits

From the results, it is shown that ANN accurately predicted the output when the training and testing R-Squared values are deemed adequate based on the user's specifications. The mean of these models are then taken and used as the final predictions and can be validated by comparing the results to the field data. This was able to show that the model prediction accuracy on a well with and without a frac-hit. From there, this distinctively showed that in cases of frac-hits the model predictions differ from the field data at times of frac-hits and gives an accurate time frame of this phenomenon which allows the interfering well to be determined.

Recommendation

The power of machine learning and artificial intelligence can provide quick, accurate analyses of seemingly complex issues in the oil and gas field. The correlations that are made using supervised methods can be applied to other to other fields. Post research recommendations that this methodology be applied to other shale formations to allow a test of its effectiveness in other locations. Along with this, it would be recommended that semi-supervised and unsupervised methods, such as DBSCAN should be testing for the frac-hit phenomena as well. These methods in conjunction with supervised methods could yield higher accuracy and detection of frac-hits. In addition to this DBSCAN, unlike ANN, does not require a portion of the data to be used for training, meaning that is has the capability of detecting frac-hit within the first 40% of the given data range that ANN would not be able to predict.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLu). Retrieved from http://arxiv.org

Aksoy, S., & Haralick, R. (2001). Feature Normalization and Likelihood-based Similarity measure for image retrieval. *Pattern Recognition Letters, 22(5), 563-582* Retrieved from www.elsevier.nl/locate/patrec

Amer, M. M., Dahab, A. S., & El-Sayed, A.-A. H. (2017). An ROP Predictive Model in Nile Delta Area Using Artificial Neural Networks. Society of Petroleum Engineers. doi:10.2118/187969-MS

Awada, A., Santo, M., Lougheed, D., Xu, D., & Virues, C. (2016). Is That Interference? A Workflow for Identifying and Analyzing Communication Through Hydraulic Fractures in a Multi-Well Pad. Society of Petroleum Engineers. doi:10.2118/178509-PA

Buhulaigah, A., Al-Mashhad, A. S., Al-Arifi, S. A., Al-Kadem, M. S., & Al-Dabbous, M. S. (2017). Multilateral Wells Evaluation Utilizing Artificial Intelligence. Society of Petroleum Engineers. doi:10.2118/183688-MS

Esquivel, R., & Blasingame, T. A. (2017). Optimizing the Development of the Haynesville Shale - Lessons Learned from Well-to-Well Hydraulic Fracture Interference. Unconventional Resources Technology Conference. doi:10.15530/URTEC-2017-2670079

Jacobs, T. (2017). Frac Hits Reveal Well Spacing May be Too Tight, Completion Volumes Too Large. *Journal of Petroleum Technology, 69(11), 35-38* doi:10.2118/1117-0035-jpt

Jacobs, T. (2017). Oil and Gas Producers Find Frac Hits in Shale Wells a Major Challenge. *Journal of Petroleum Technology. 69(4), 29-34* doi:10.2118/0417-0029-jpt

Jacobs, T. (2018). In the Battle Against Frac Hits, Shale Producers Go to New Extremes. *Journal of Petroleum Technology, 70(08), 35-38* doi:10.2118/0818-0035-jpt

Kingma, D, Ba, J. Adam. (2015). A Method For Stochastic Optimization. The International Conference on Learning Representations (ICLR) Retrieved from http://arxiv.org

Molina, O., & Zeidouni, M. (2017). Analytical Model to Estimate the Fraction of Fracture Hits in a Multi-Well Pad. Society of Petroleum Engineers. doi:10.2118/187501-MS

Rassenfoss, S. (2017). Rethinking Fracturing: The Problems With Bigger Fracs in Tighter Spaces. *Journal of Petroleum Technology, 69(12), 28-34* doi:10.2118/1217-0028-jpt

Ren, G., Cao, Y., Wen, S., Huang, T., & Zeng, Z. (2018). A modified Elman neural network with a new learning rate scheme. *Neurocomputing, 286, 11-18* doi:10.1016/j.neucom 2018.01.046

Stanford University. (2018). Optimization: Stochastic Gradient Descent. Retrieved from http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/

Zborowski, M. (2018). Can Machine Learning Mitigate Frac Hits? Retrieved from https://www.spe.org/en/jpt/jpt-article-detail/?art=4762