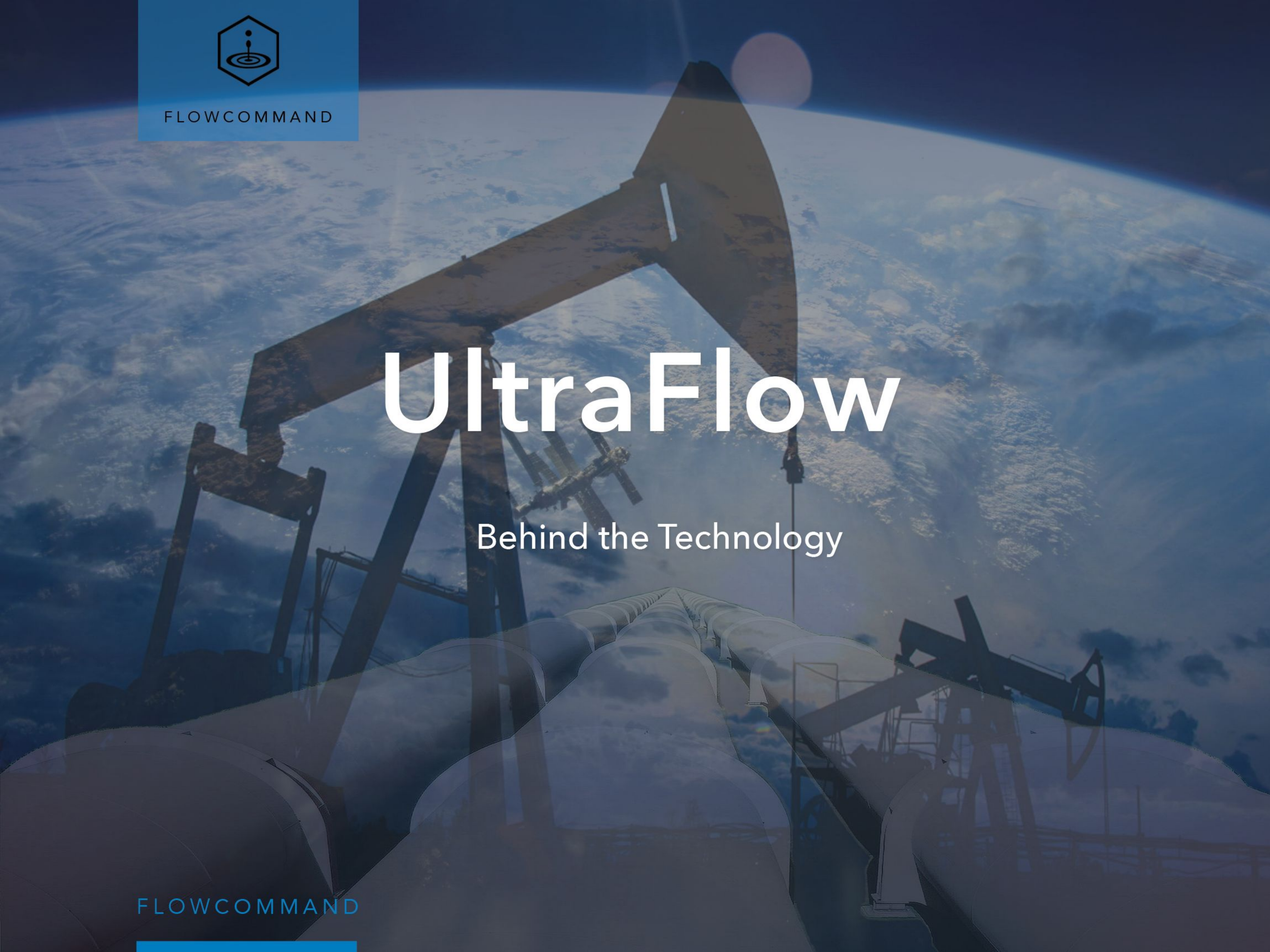
FLOWCOMMAND



UltraFlow

Behind the Technology

FLOWCOMMAND

An evolution in sensor engineering

FlowCommand is focused on reducing customer operating costs through the market's most advanced technology to measure fluid rates. Using a combination of machine learning, satellite telemetry, and proprietary hardware – FlowCommand systems allow oil and gas operators to track fluid behavior inside of any pipe. The unique combination of these features allow us to make a flow sensor that anyone can install with five minutes and a screwdriver.

The measurement process *is not driven by* a sensor that measures a single variable and performs basic physics equations. It is instead a multistep process that collects incredibly large amounts of acoustic data and transmits that data to several layers of machine learning algorithms that have been trained on billions of data points. By leveraging this dataset, we can determine the flowrate wherever a FlowCommand transducer is installed on a pipe.

Sensors that measure cooperatively

The precise engineering and computer science driving this solution involves a multi-stage framework. **There are three steps involved in our measurement process, each broken into smaller steps in the below outline:**

Data Acquisition	**Data Transmission**	**Data Process**
Transducer sends waves through pipe to reflect off of fluid	Sends to satellite network	Processed by model
Sends wave reflection data to CPU	Sends to software	Displays to customer

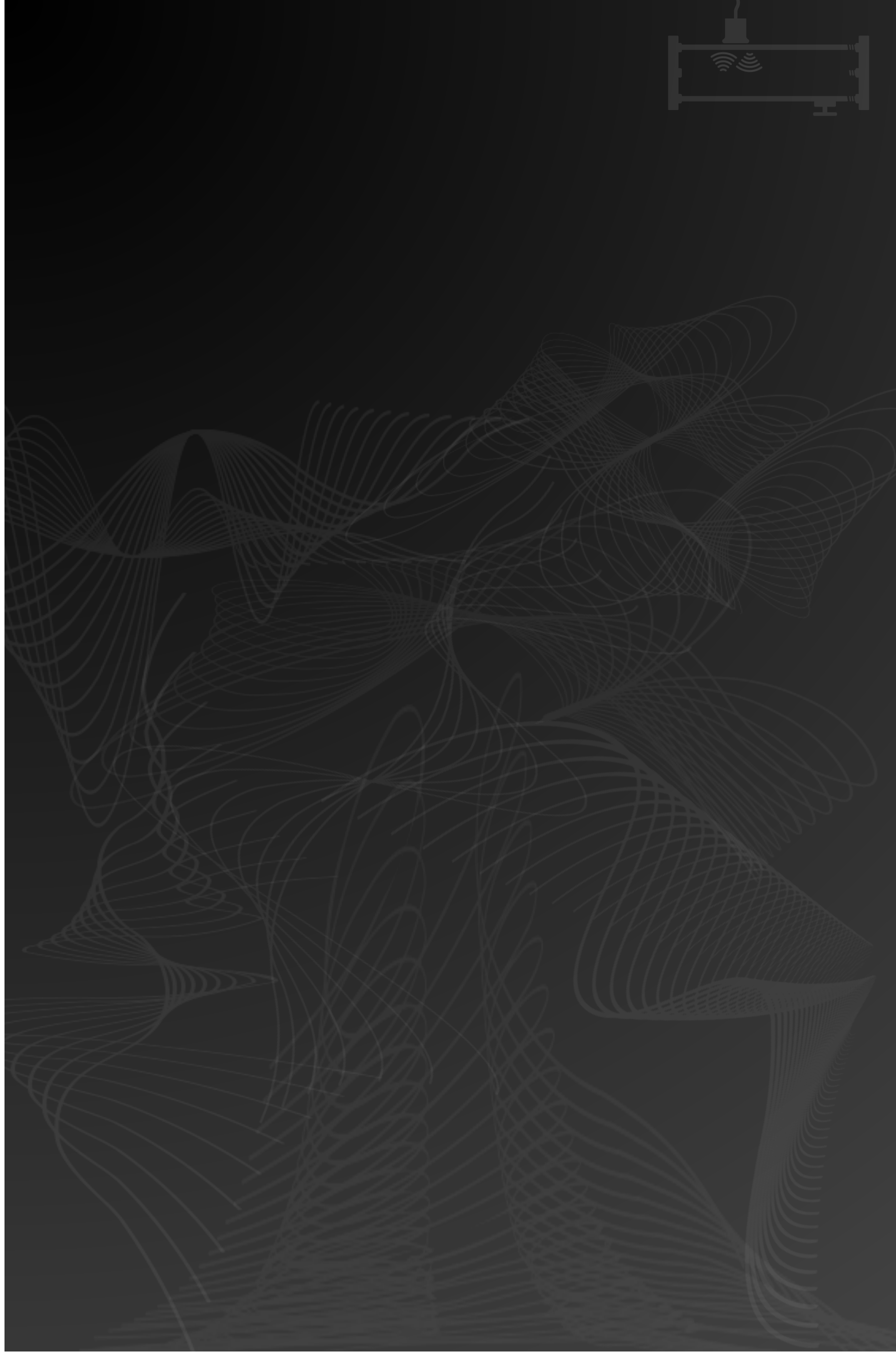
FLOWCOMMAND

Acquiring raw ultrasonic data

FlowCommand's transducer emits ultrasonic waves into the pipe. Those waves are then reflected off microscopic particles and bubbles in the fluid. Millions of these data points are recorded every second and are sent to our computing module to transmit to our software.

As with all modern machine learning and artificial intelligence techniques, the FlowCommand system extracts many important variables from the ultrasonic waves that describe the shape, size, and movement of the waves. In standard machine learning terms, these variables are called **features**.

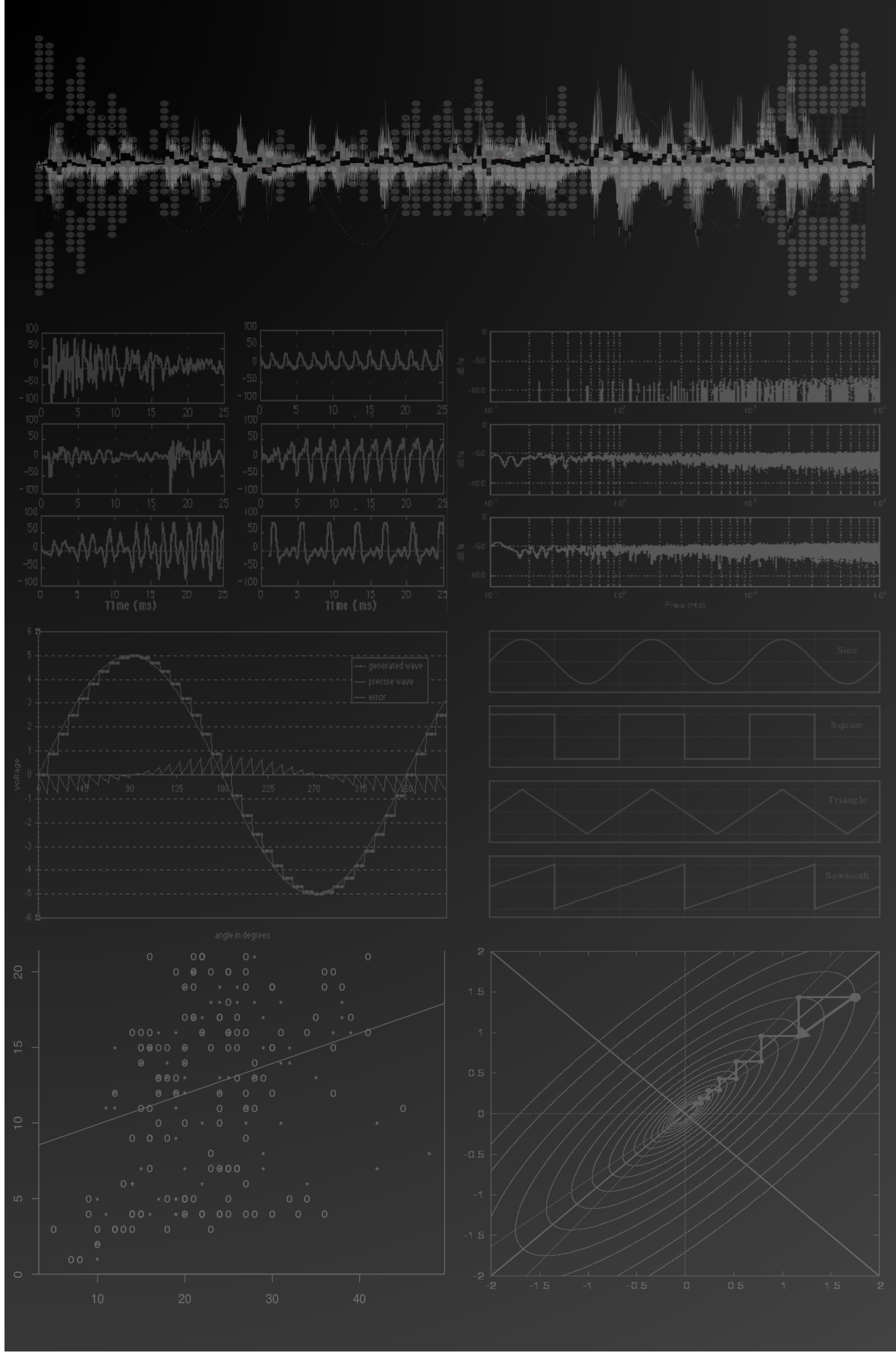
There are far too many features to list—many of them are protected under trade secret, while others are simply impossible to describe in plain english terms since they were identified by a machine learning algorithm itself.



Example:
Wave analysis

To help exemplify our solution's measurement process and algorithms, to the right is a representation of the statistical analyses our software automatically performs on properties of the ultrasonic waves emitted and received by the transducer. These properties are a small component of the inputs into FlowCommand Software.

Approximately one million waves are sent and received every second; thousands of variables are captured on each wave as well aggregated sets of waves.



Seamless data transmission

All of the data related to the wave signal reflections is compressed and encrypted by our processor on the unit in the field. This encrypted data is then transmitted via a technology called SBD (Short Burst Data) to a private non-geosynchronous satellite network.

This data is automatically downloaded by FlowCommand servers and processed by a piece of FlowCommand software called an Ingestion Engine that lives on the cloud. This means that the Customer has access to this information from any device with an internet connection. This Ingestion Engine decompresses the data and breaks it into relevant pieces to be processed by our algorithms.

Additionally, all FlowCommand units are equipped with two way communication abilities. This allows FlowCommand software to automatically send transmissions with updated programming or command functions to units in the field.

Measurement powered by AI

Once the FlowCommand data model has received the inputs from the ingestion engine, the model then automatically examines the data to determine which class of algorithm to implement. The selection process is based on the evaluation of thousands of data points, compared against a library of billions.

The FlowCommand machine learning model does not rely on any particular features; rather, it performs a variety of pattern recognition tasks that seek to understand the thousands of features, across millions of data points, in their totality to determine a flow rate. The key factor in giving our algorithm such accurate performance is that every sensor is constantly communicating with our library of data points and comparing what it hears with every piece of data that every other sensor has ever collected.

The reason for using this level of complexity is that most **flow regimes in the oilfield are complex**. They often have turbulence, varying fluid makeups, and other complexities that require evaluation techniques that are **far beyond the typical physics model** or single variable evaluation.

```
linear = linear_model.LinearRegression()
trainX =
np.asarray(df.X[20:len(df.X)]).reshape(-1, 1)
trainY =
np.asarray(df.Y[20:len(df.Y)]).reshape(-1, 1)
testX = np.asarray(df.X[:20]).reshape(-1, 1)
testY = np.asarray(df.Y[:20]).reshape(-1, 1)
linear.fit(trainX, trainY)
linear.score(trainX, trainY)
print('Coefficient: \n', linear.coef_)
print('Intercept: \n', linear.intercept_)
print('R² Value: \n', linear.score(trainX,
trainY))
predicted = linear.predict(testX)

logistic = LogisticRegression()
X = (np.asarray(df.X)).reshape(-1, 1)
Y = (np.asarray(df.Y)).ravel()
logistic.fit(X, Y)
logistic.score(X, Y)
print('Coefficient: \n', logistic.coef_)
print('Intercept: \n', logistic.intercept_)
print('R² Value: \n', logistic.score(X, Y))

from sklearn.cross_validation import
train_test_split
decision =
tree.DecisionTreeClassifier(criterion='gini')
X = df.values[:, 0:4]
Y = df.values[:, 4]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
decision.fit(trainX, trainY)
print('Accuracy: \n', decision.score(testX,
testY))

from sklearn.cross_validation import
train_test_split
support = svm.SVC()
X = df.values[:, 0:2]
Y = df.values[:, 2]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
support.fit(trainX, trainY)
print('Accuracy: \n', support.score(testX,
testY))
pred = support.predict(testX)

from sklearn.cross_validation import
train_test_split
neighbors =
KNeighborsClassifier(n_neighbors=5)
X = df.values[:, 0:2]
Y = df.values[:, 2]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
neighbors.fit(trainX, trainY)
print('Accuracy: \n', neighbors.score(testX,
testY))
pred = neighbors.predict(testX)
```

```
from sklearn.cross_validation import
train_test_split
decision =
tree.DecisionTreeClassifier(criterion='gini
')
X = df.values[:, 0:4]
Y = df.values[:, 4]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
decision.fit(trainX, trainY)
print('Accuracy: \n', decision.score(testX,
testY))

from sklearn.cross_validation import
train_test_split
support = svm.SVC()
X = df.values[:, 0:2]
Y = df.values[:, 2]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
support.fit(trainX, trainY)
print('Accuracy: \n', support.score(testX,
testY))
pred = support.predict(testX)

from sklearn.cross_validation import
train_test_split
neighbors =
KNeighborsClassifier(n_neighbors=5)
X = df.values[:, 0:2]
Y = df.values[:, 2]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
neighbors.fit(trainX, trainY)
print('Accuracy: \n',
neighbors.score(testX, testY))
pred = neighbors.predict(testX)

from sklearn.cross_validation import
train_test_split
forest = RandomForestClassifier()
X = df.values[:, 0:4]
Y = df.values[:, 4]
trainX, testX, trainY, testY =
train_test_split( X, Y, test_size = 0.3)
forest.fit(trainX, trainY)
print('Accuracy: \n', forest.score(testX,
testY))
pred = forest.predict(testX)

from sklearn import decomposition
pca = decomposition.PCA()
fa = decomposition.FactorAnalysis()
X = df.values[:, 0:4]
Y = df.values[:, 4]
train, test = train_test_split(X,test_size
= 0.3)
train_reduced = pca.fit_transform(train)
test_reduced = pca.transform(test)
pca.n_components_
```

FLOWCOMMAND

FlowCommand Inc.
10606 Hempstead Rd.   Suite 112
Houston, TX 77092
T +1 713 714 5547

team@flowcommand.com
www.flowcommand.com

FLOWCOMMAND