Masquerading Malicious DNS Traffic

Bayesian Inference, Rainier, Spark

David Rodriguez
March 28, 2019

The Outline

Masquerading
DNS
Traffic

Time Series
Modeling

Rainier
+
Spark

Anomaly
Detection

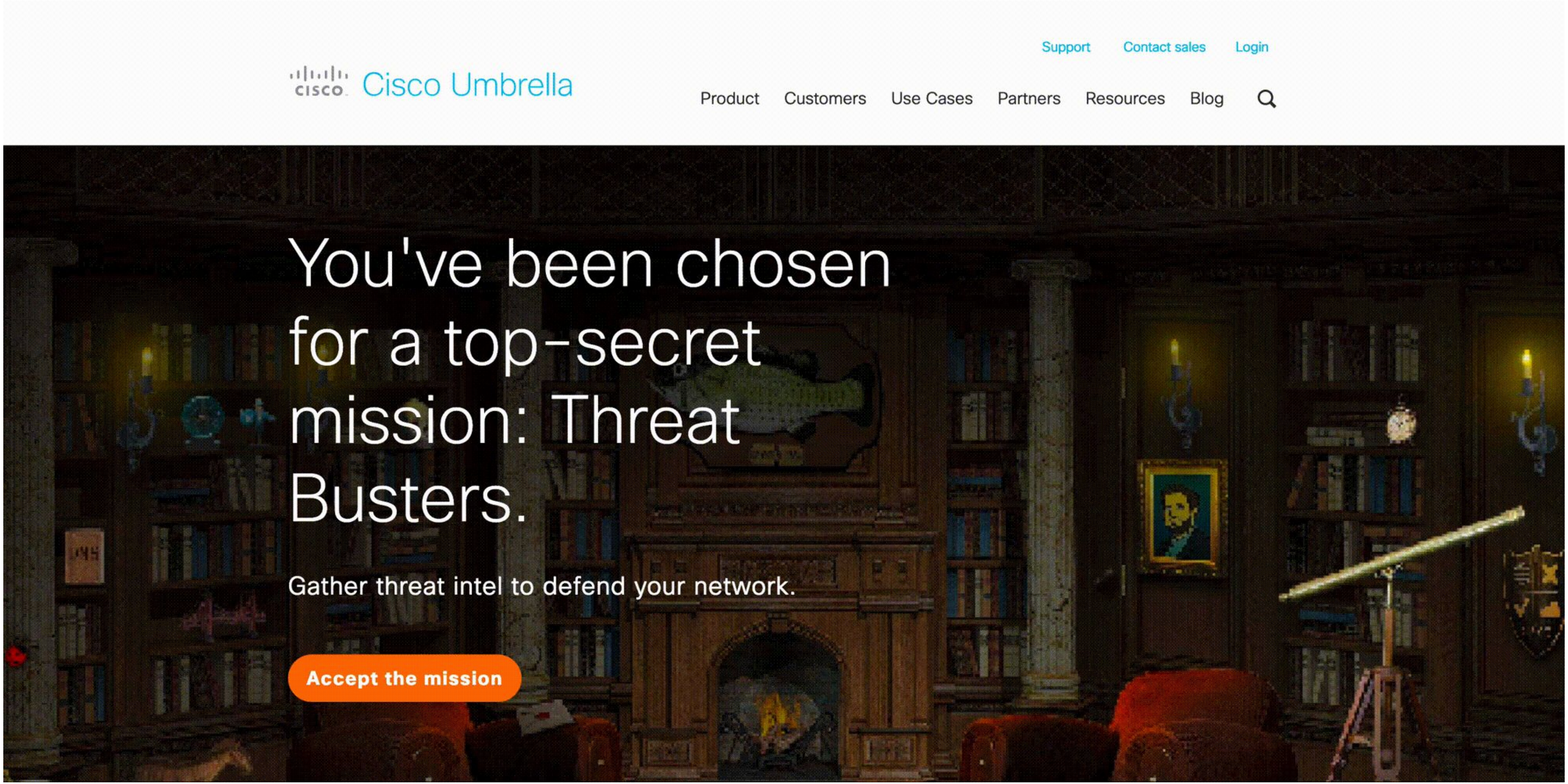
The Outline

Masquerading
DNS
Traffic

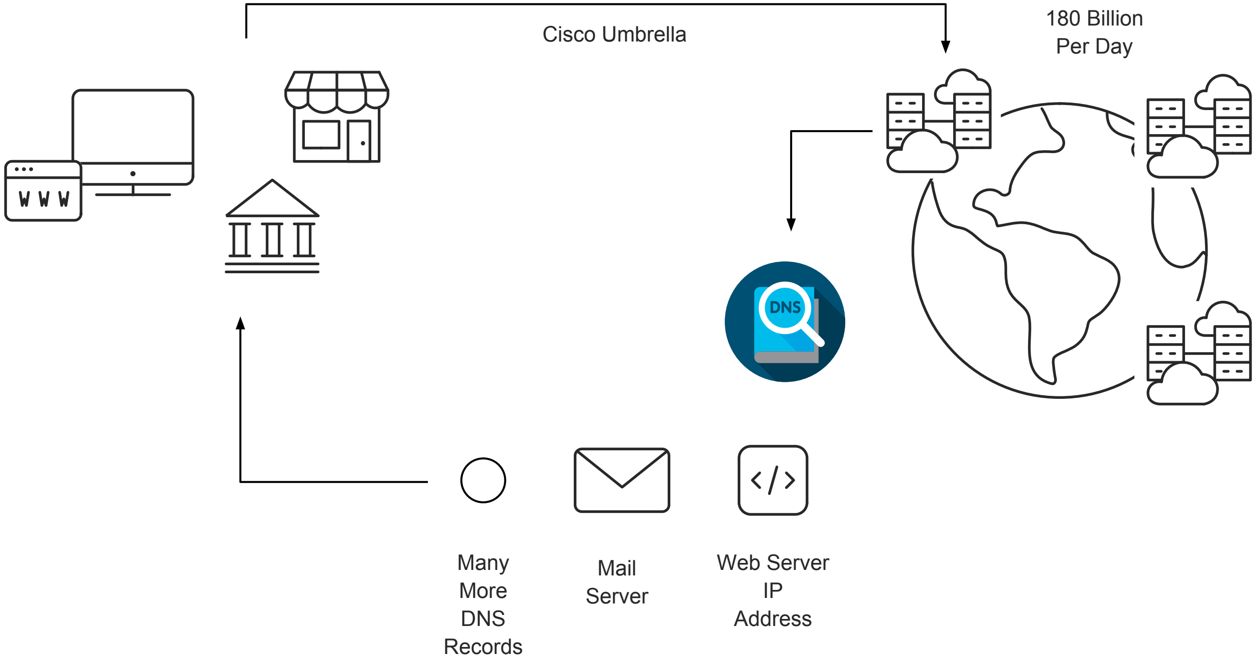
Time Series
Modeling

Rainier
+
Spark

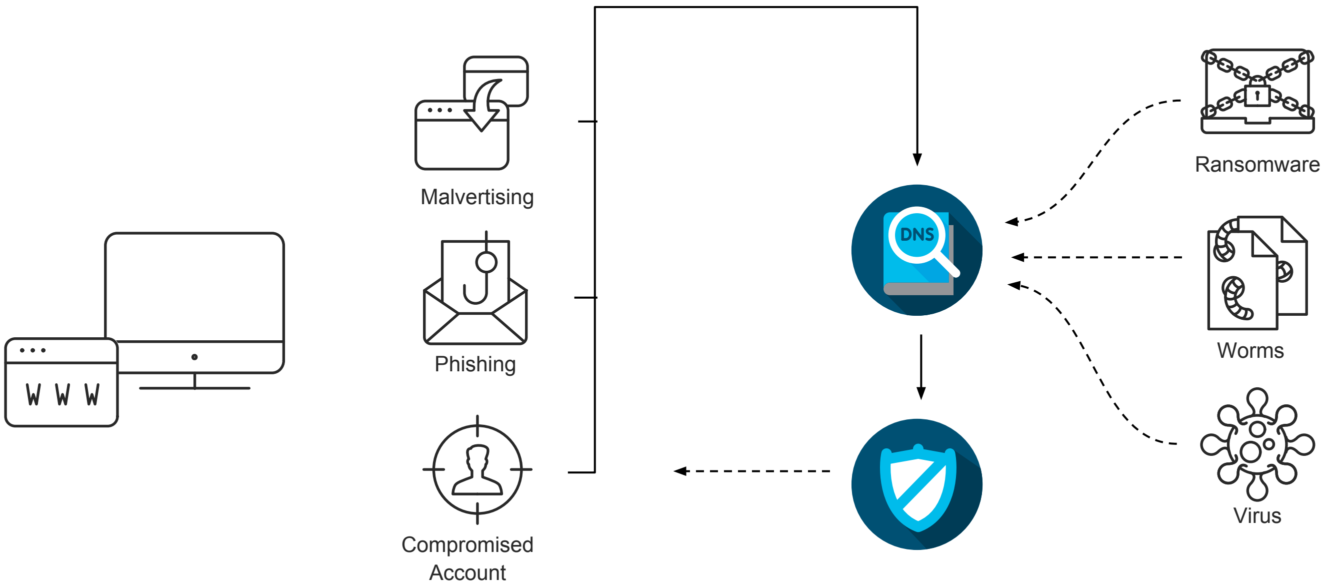
Anomaly
Detection


Support
Contact sales
Login
cisco Cisco Umbrella
Product
Customers
Use Cases
Partners
Resources
Blog
You've been chosen for a top-secret mission: Threat Busters.
Gather threat intel to defend your network.
Accept the mission

Part 1 DNS Resolution



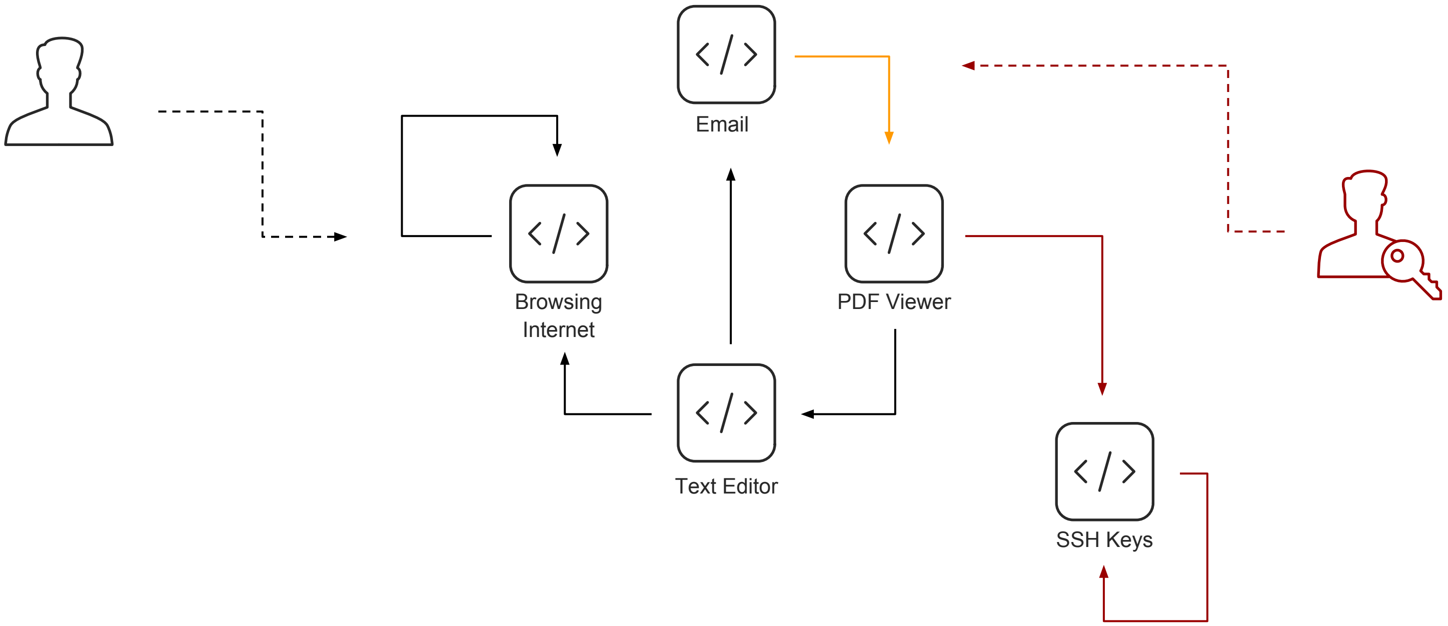
Part 1 Protection 101



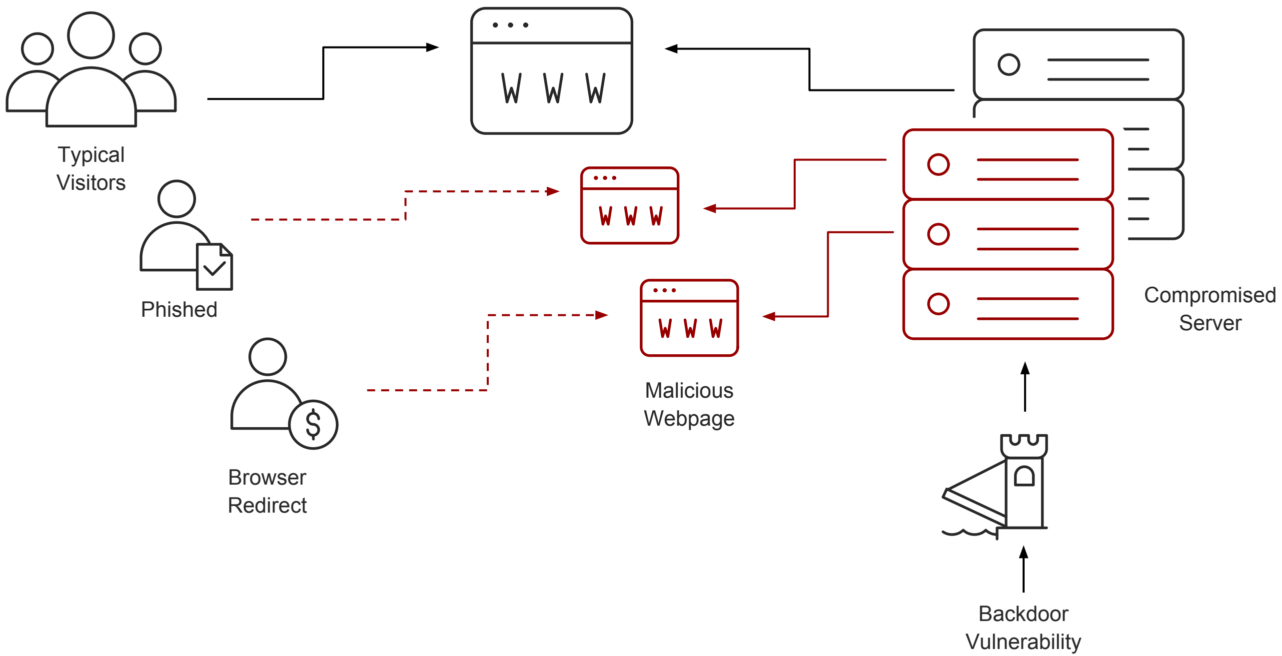
Part 1 Definition

Masquerading Traffic = Masquerading Users +
Compromised Websites

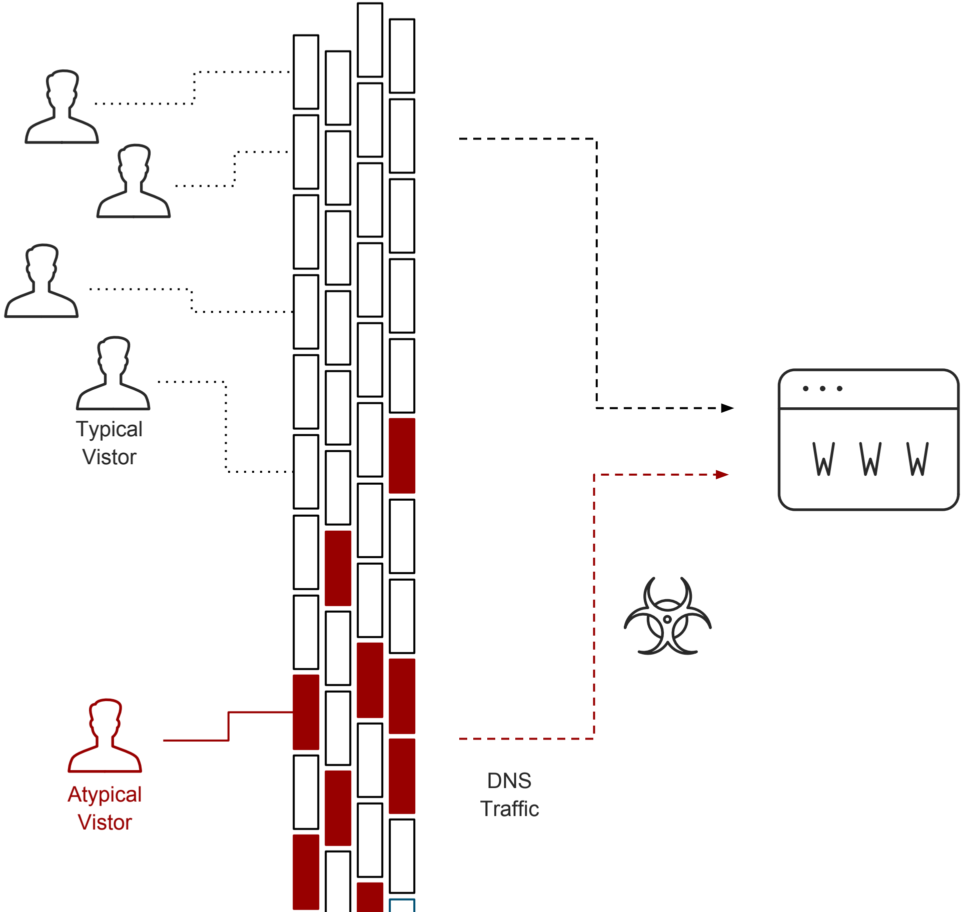
Part 1 Masquerading Users



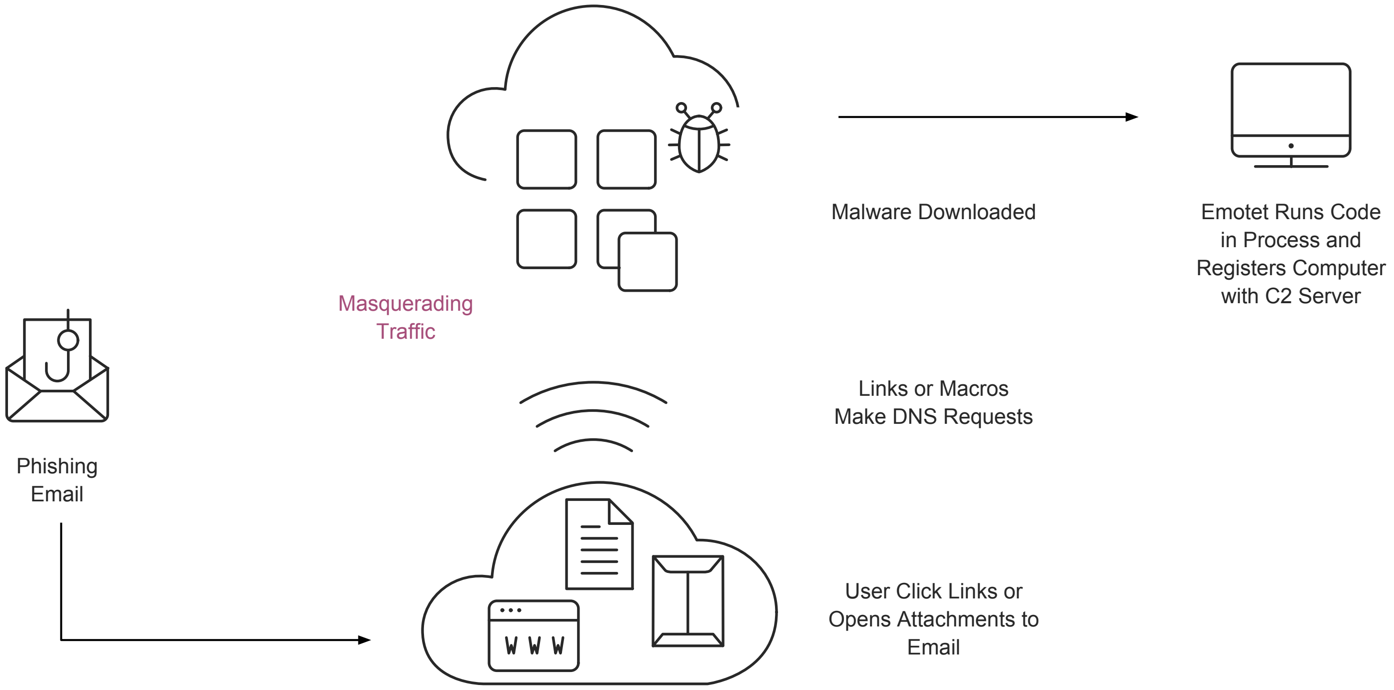
Part 1 Compromised Websites



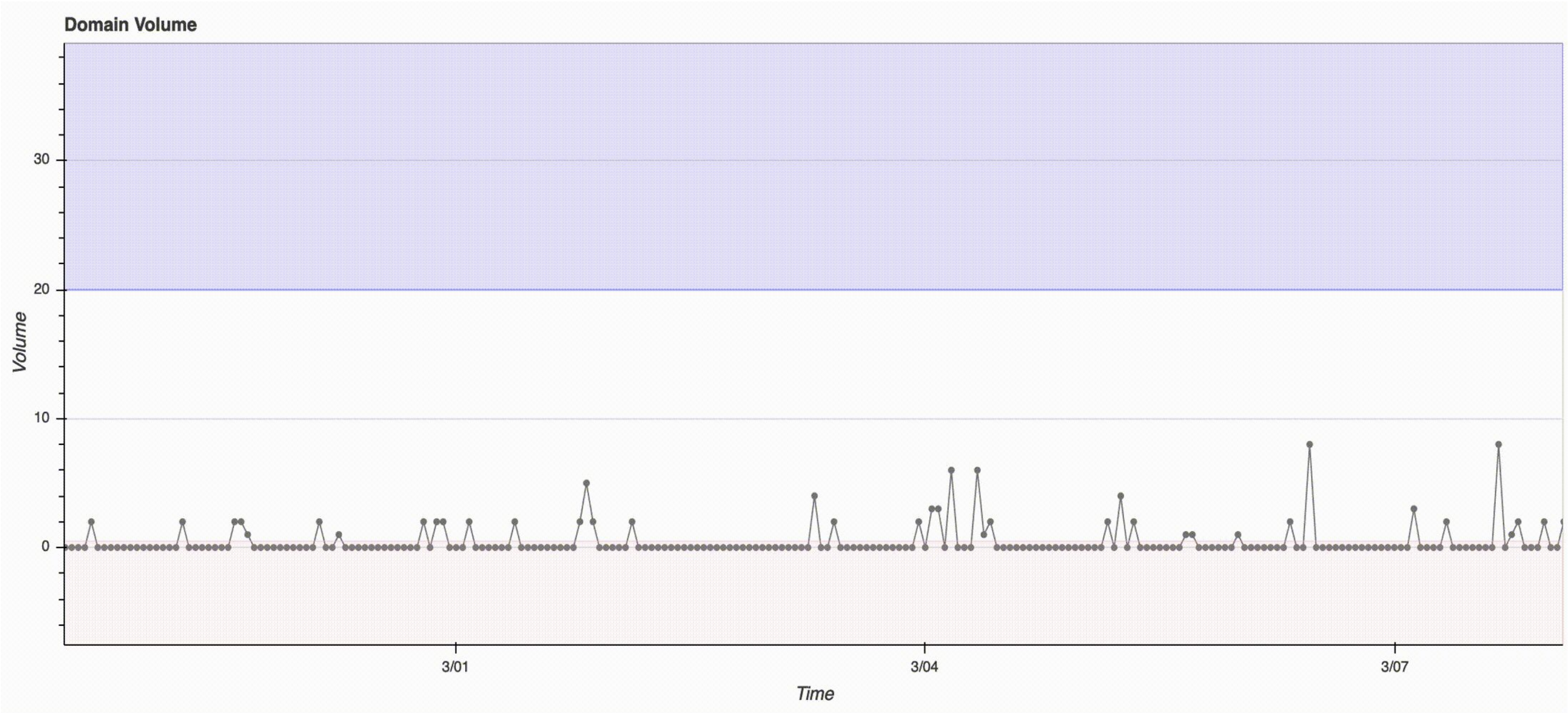
Part 1 Masquerading DNS Traffic



Part 1 Emotet Campaign



Part 1 Emotet Campaign



The Outline

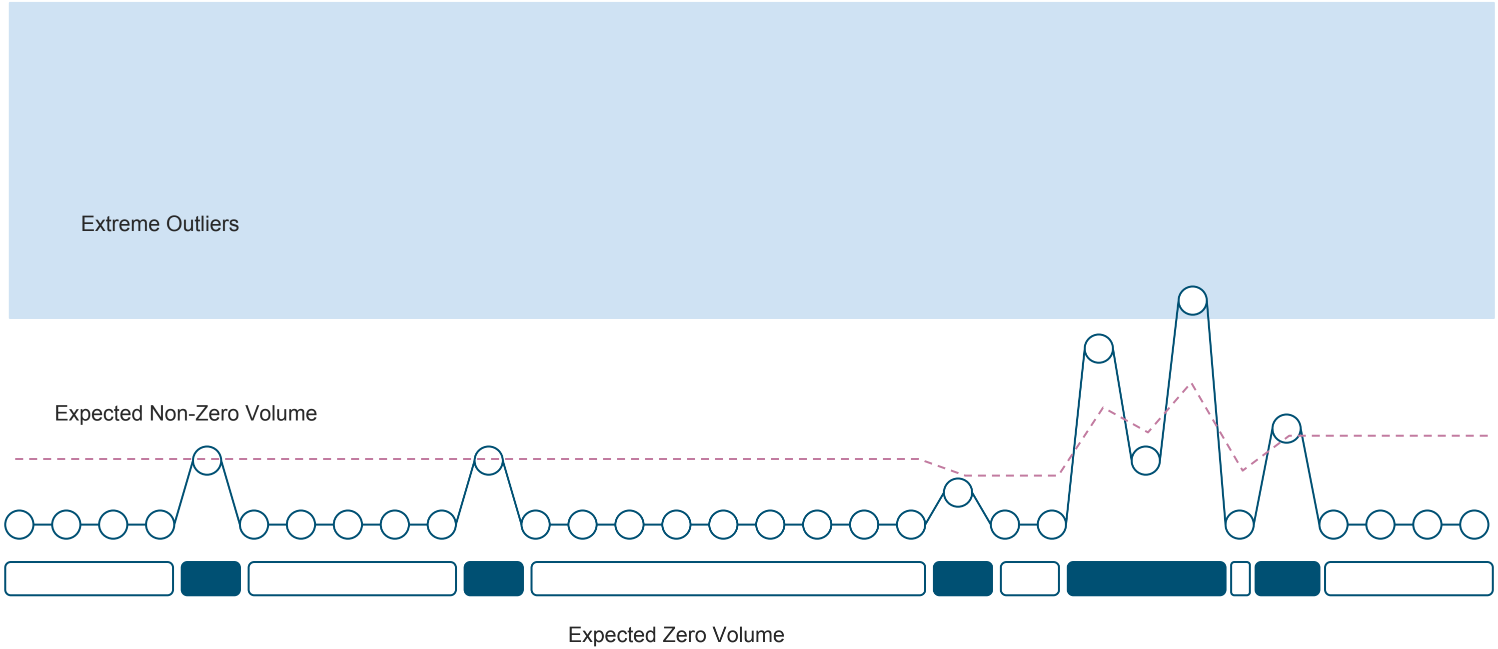
Masquerading
DNS
Traffic

Time Series
Modeling

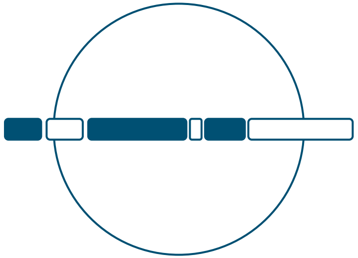
Rainier
+
Spark

Anomaly
Detection

Part 2 Time-Series Analysis

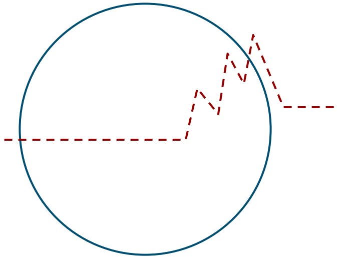


Part 2 Time-Series Analysis



ψ

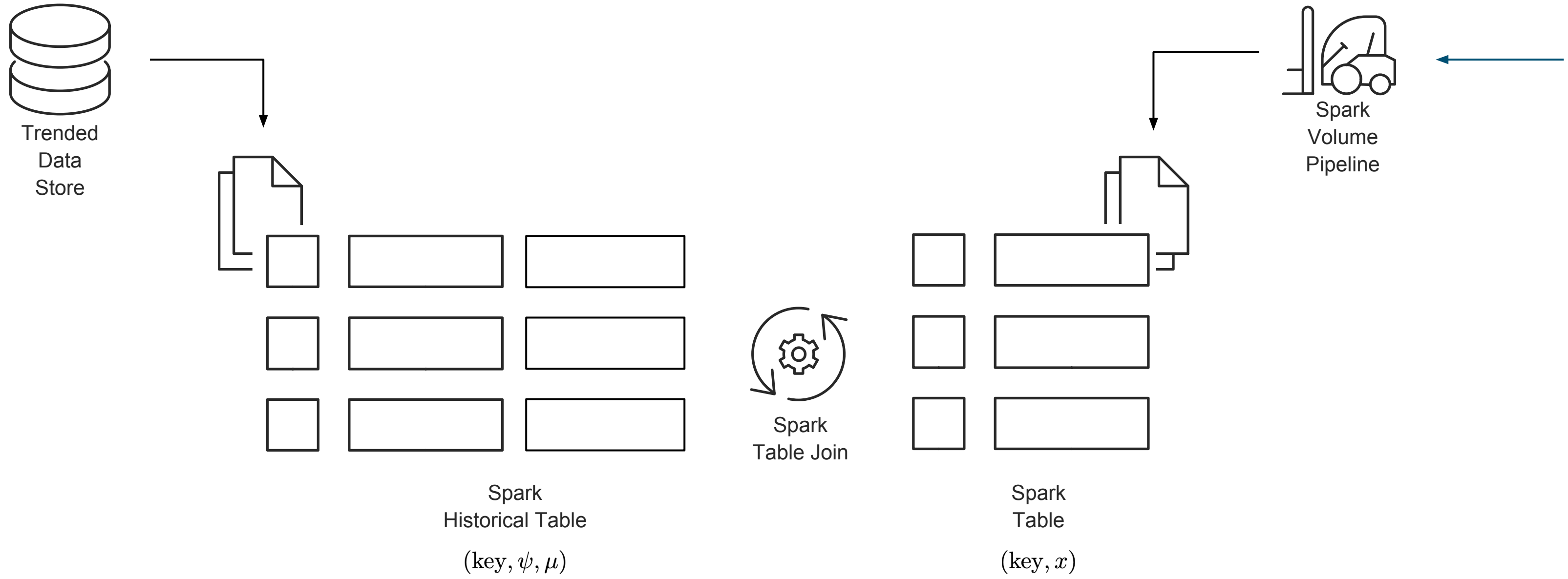
Probability of Demand



μ

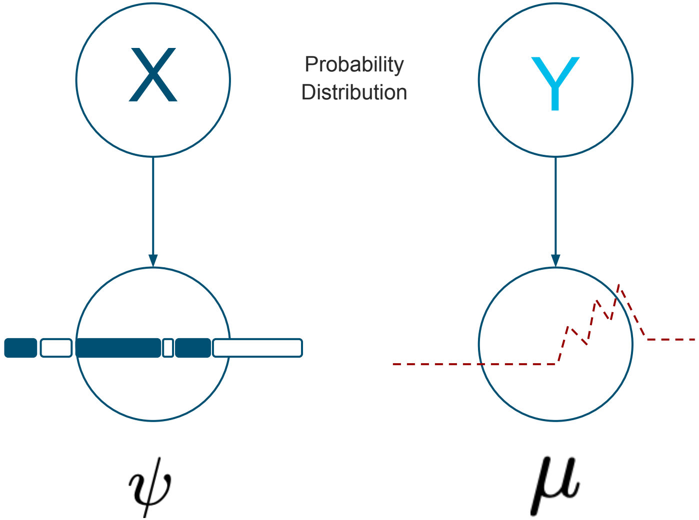
Expected Demand
when non-zero

Part 2 Croston's Method



Note : $\mu_{new} = \mu \cdot w + (1 - w) \cdot x$ $\quad$ $\psi_{new} = \psi \cdot w + (1 - w) \cdot \chi_{x>0}$

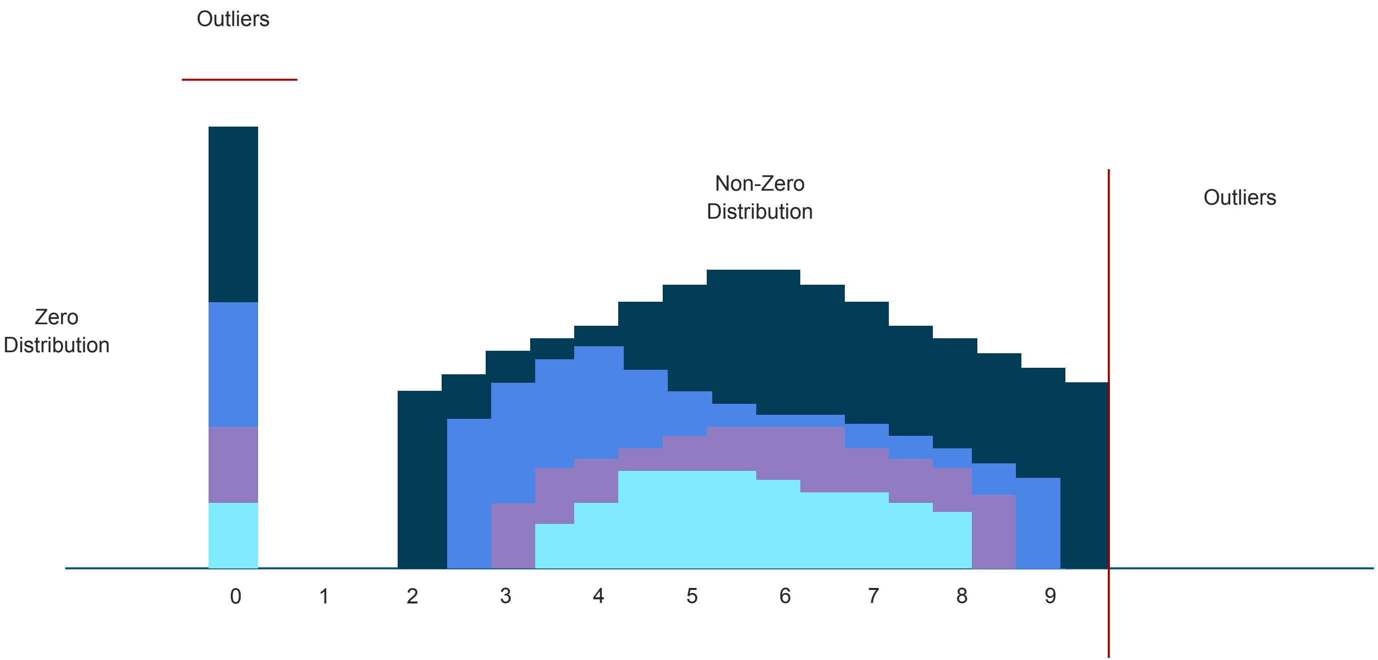
Part 2 Bayesian Approach



Probability of Demand

Expected Demand
when non-zero

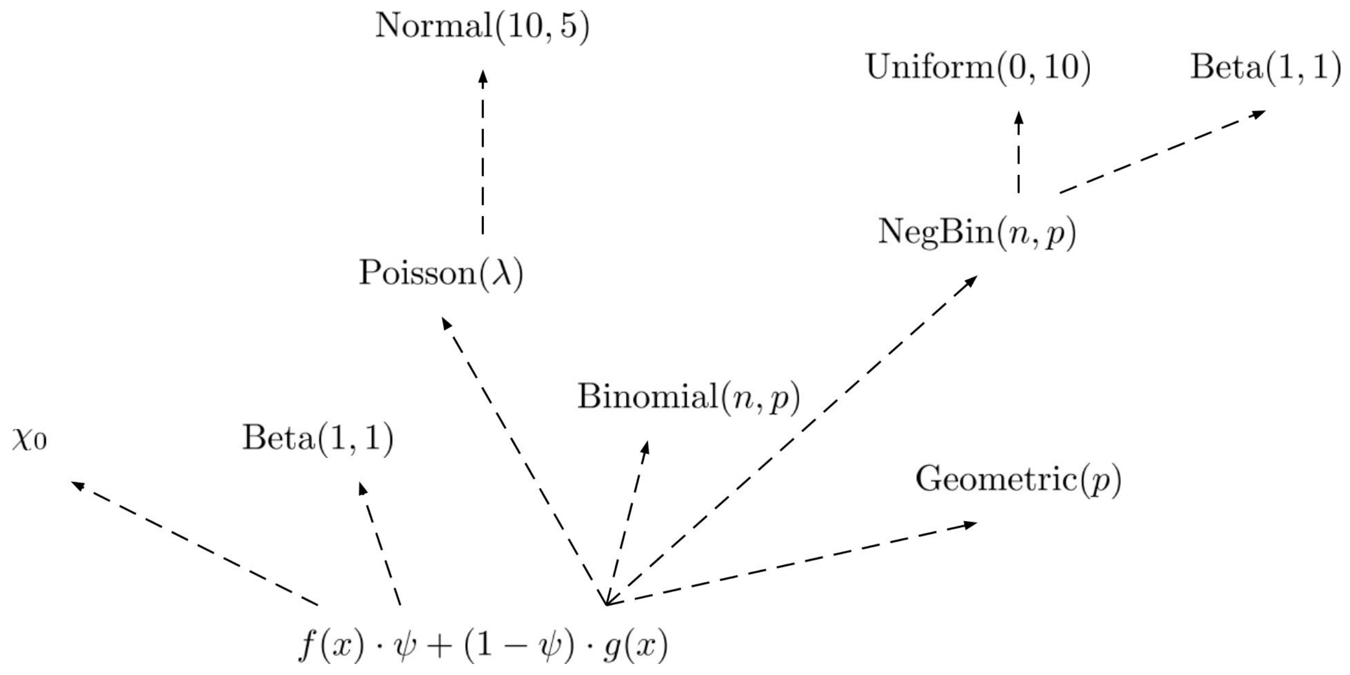
Part 2 Bayesian Approach



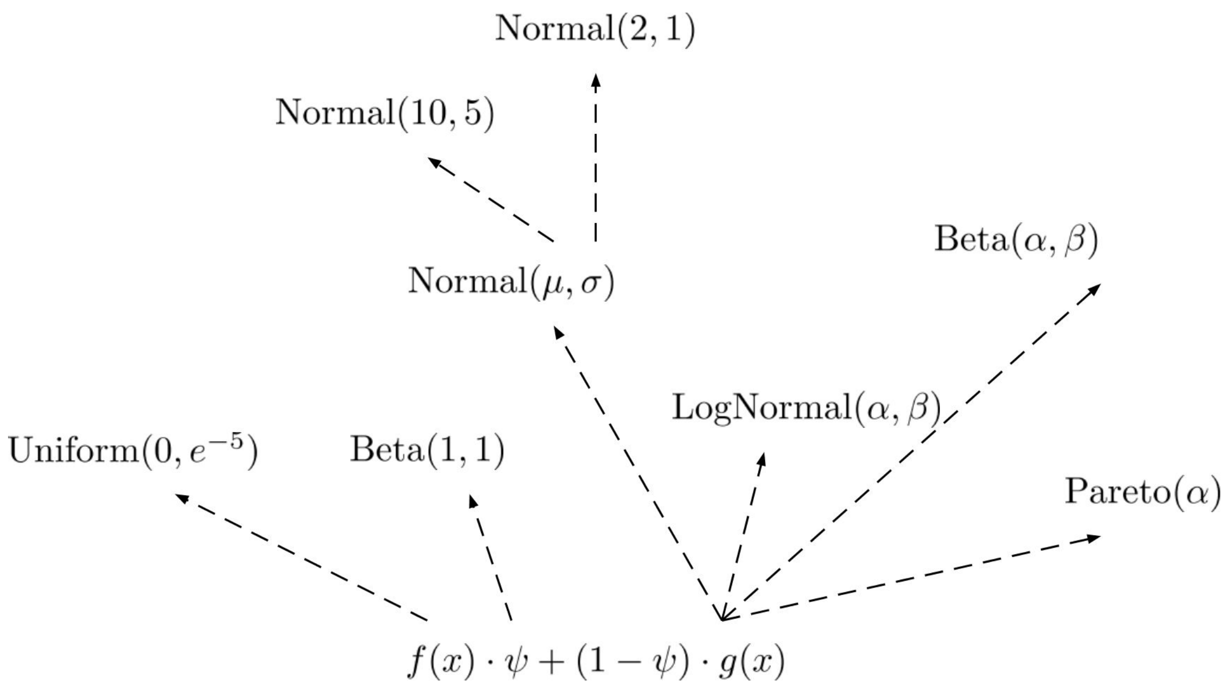
Part 2 Mixture Models

$$f(x) \cdot \psi + (1 - \psi) \cdot g(x)$$

Part 2 Discrete Models



Part 2 Continuous Models



The Outline

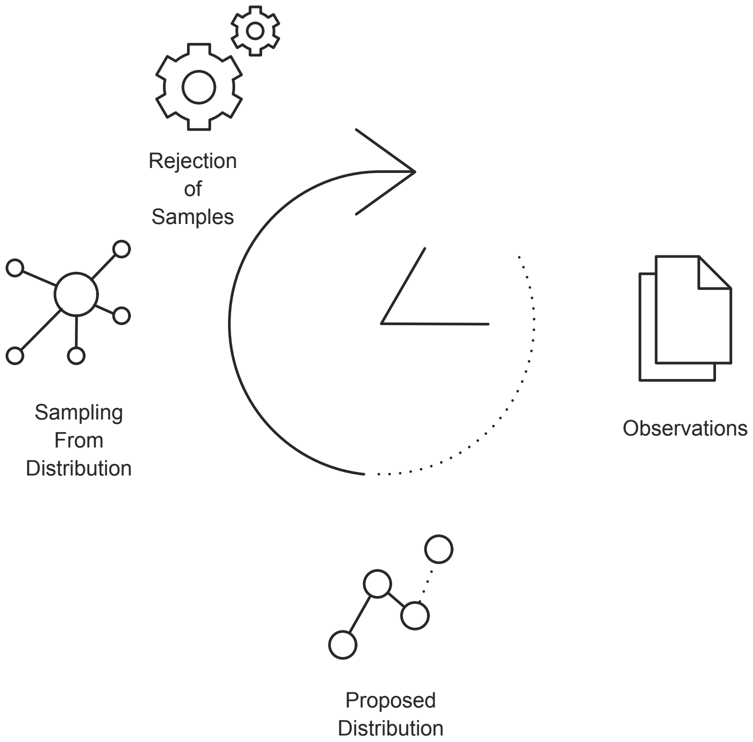
Masquerading
DNS
Traffic

Time Series
Modeling

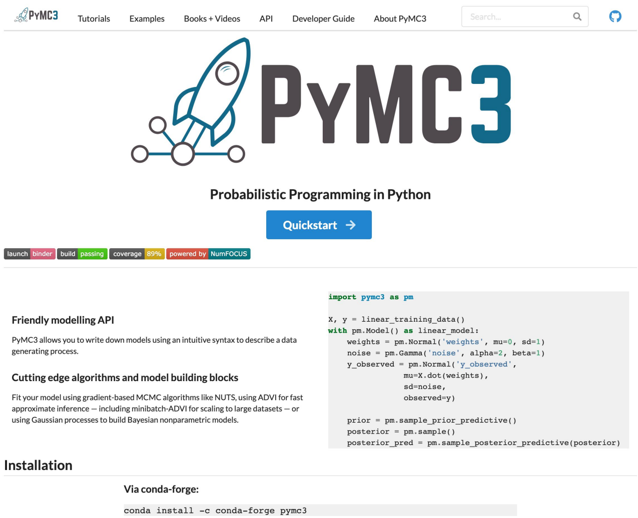
Rainier
+
Spark

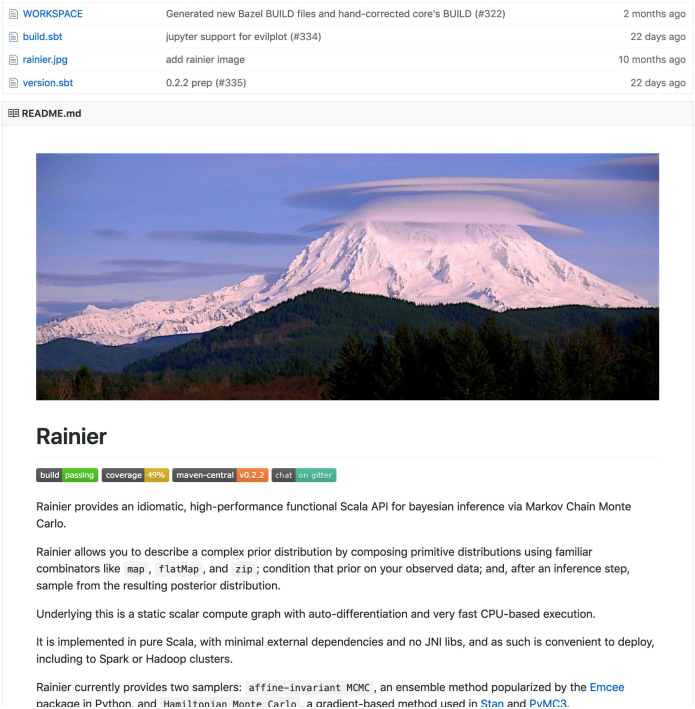
Anomaly
Detection

Part 3 MCMC Methods



Part 3 MCMC Methods





Part 3 Rainier

Depending on your background, you might think of Rainier as aspiring to be either:

“Stan, but on the JVM”

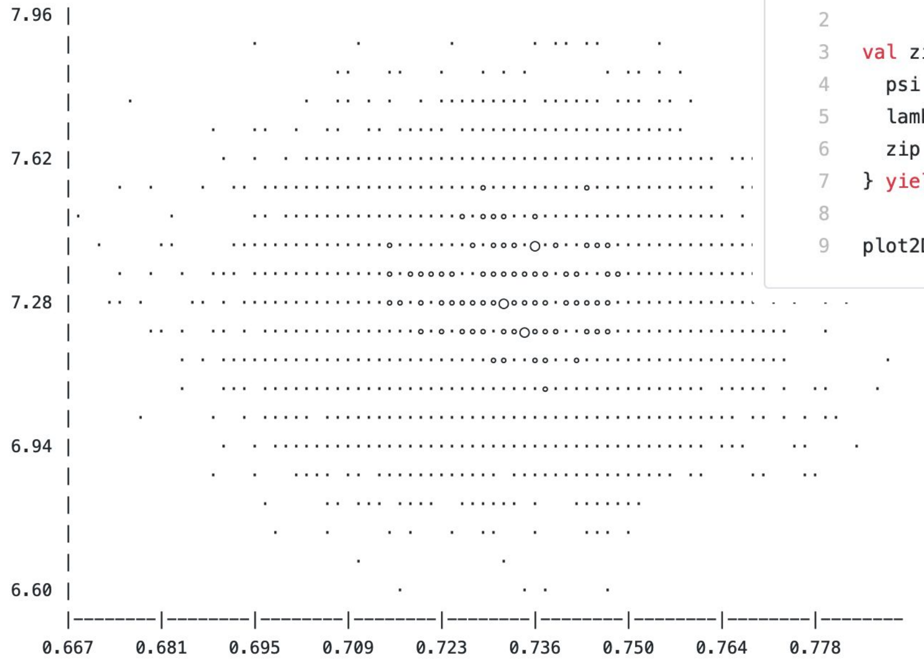
or

“Tensorflow, but for small data”.

~ README

Part 3 Rainier Methods

fit.txt

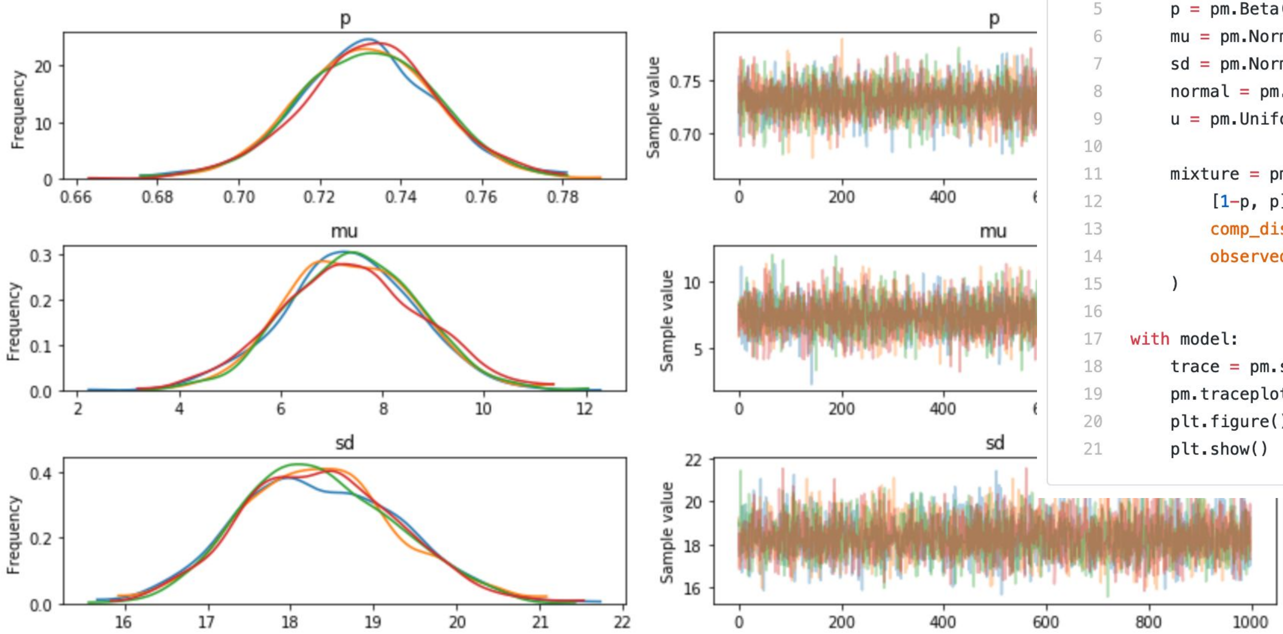


scala-repl.scala

```scala
val data = List(2,0,0,3,3,3,0,3,5,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0

val zipModel = for {
  psi <- Beta(1,1).param
  lambda <- Normal(5, 5).param
  zip <- Poisson(lambda).zeroInflated(psi).fit(data)
} yield (psi, lambda)

plot2D(zipModel.sample())
```

Part 3 PyMC Methods

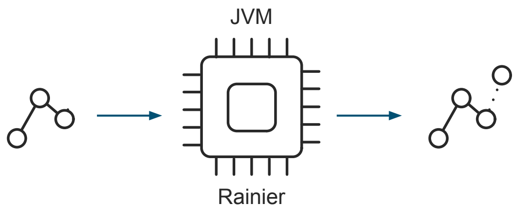
zero_inflated_normal.png

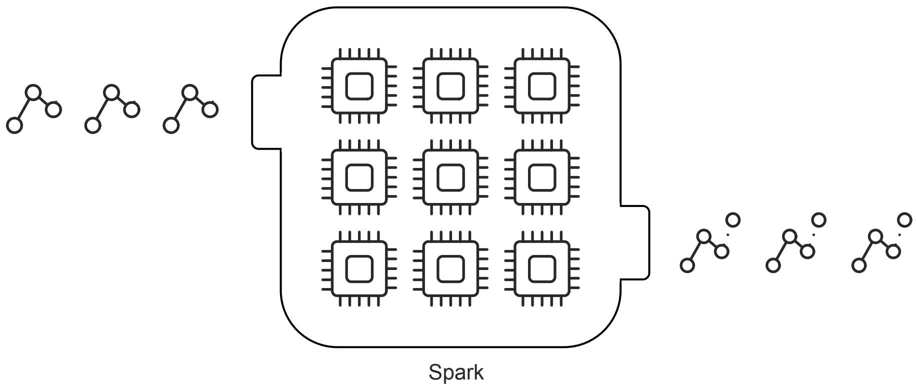


pymc-zero-inflated-normal.py

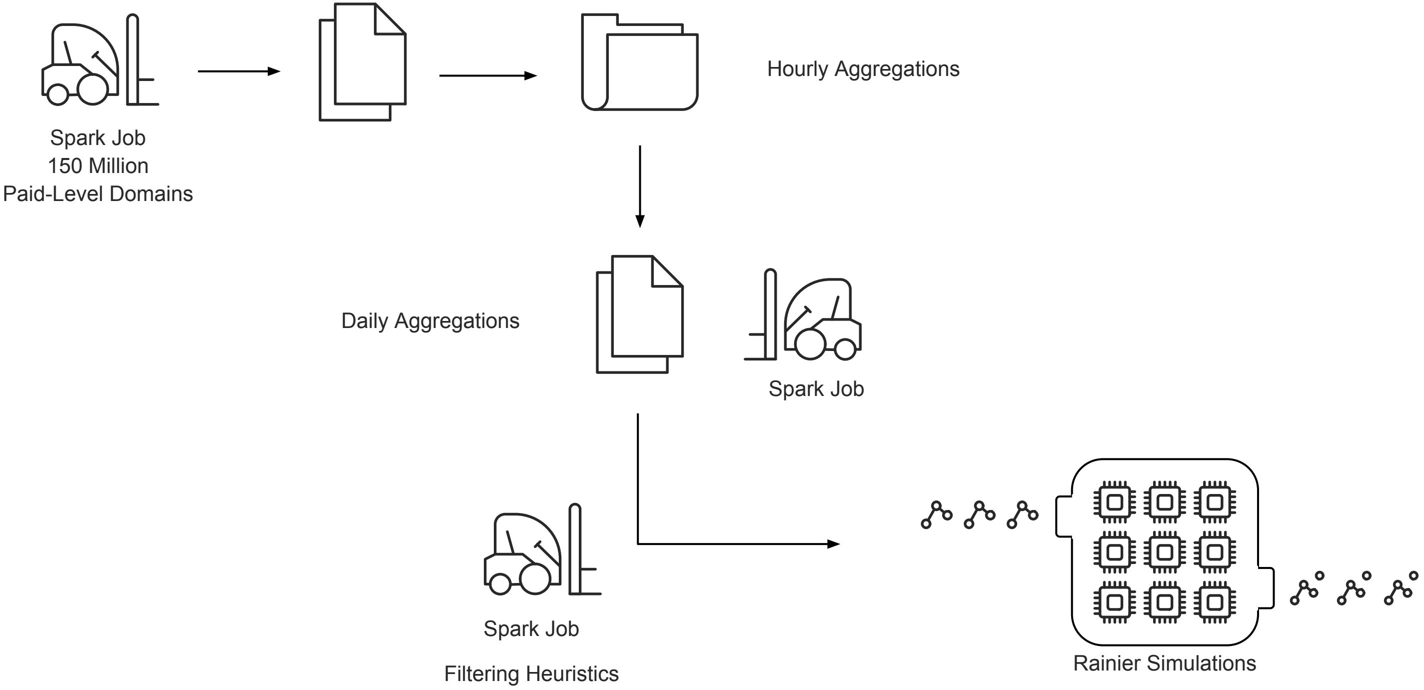
```python
data = [7.650241800665554e-05,3.923801883327712e-05,3.0,3.0,3.

model = pm.Model()
with model:
    p = pm.Beta('p', 1, 1)
    mu = pm.Normal('mu', 10, 10)
    sd = pm.Normal('sd', 10, 10)
    normal = pm.Normal('normal', mu, sd)
    u = pm.Uniform('u', 0, 1e-5)

    mixture = pm.Mixture('mixture',
        [1-p, p],
        comp_dists=[normal.distribution,u.distribution],
        observed=data
    )

with model:
    trace = pm.sample(1000, tune=1000)
    pm.traceplot(trace, varnames=['p', 'mu', 'sd'])
    plt.figure()
    plt.show()
```

Part 3 Rainier + Spark





Part 3 Rainier + Spark



[Rainier] Massive Bayesian Inference in Spark using Rainer

SparkRainer.scala

Raw

```scala
import com.stripe.rainier.core.{Normal, Poisson}
import com.stripe.rainier.sampler.{RNG, ScalaRNG}
import org.apache.spark.{SparkConf, SparkContext}

object Driver {
  implicit val rng: RNG = ScalaRNG(1527608515939L)
  val DROP_BURN_IN = 100

  /*
   Refer to StackOverflow Q, about serializing methods/objects:
         https://stackoverflow.com/questions/22592811/task-not-serializable-java-io-notserializableexception-when-calling-funct
  */
  def genMapper[A, B](f: A => B): A => B = {
    val locker = com.twitter.chill.MeatLocker(f)
    x => locker.get.apply(x)
  }

  def average(l: List[Double]): Double =
    l.size.toDouble / l.sum

  def dropBurnIn(dropBurn: Int)(v: List[Double]): List[Double] =
    v.drop(dropBurn)


  def fitPoisson(y: List[Int]): List[Double] = {
    val rate = for {
```

The Outline

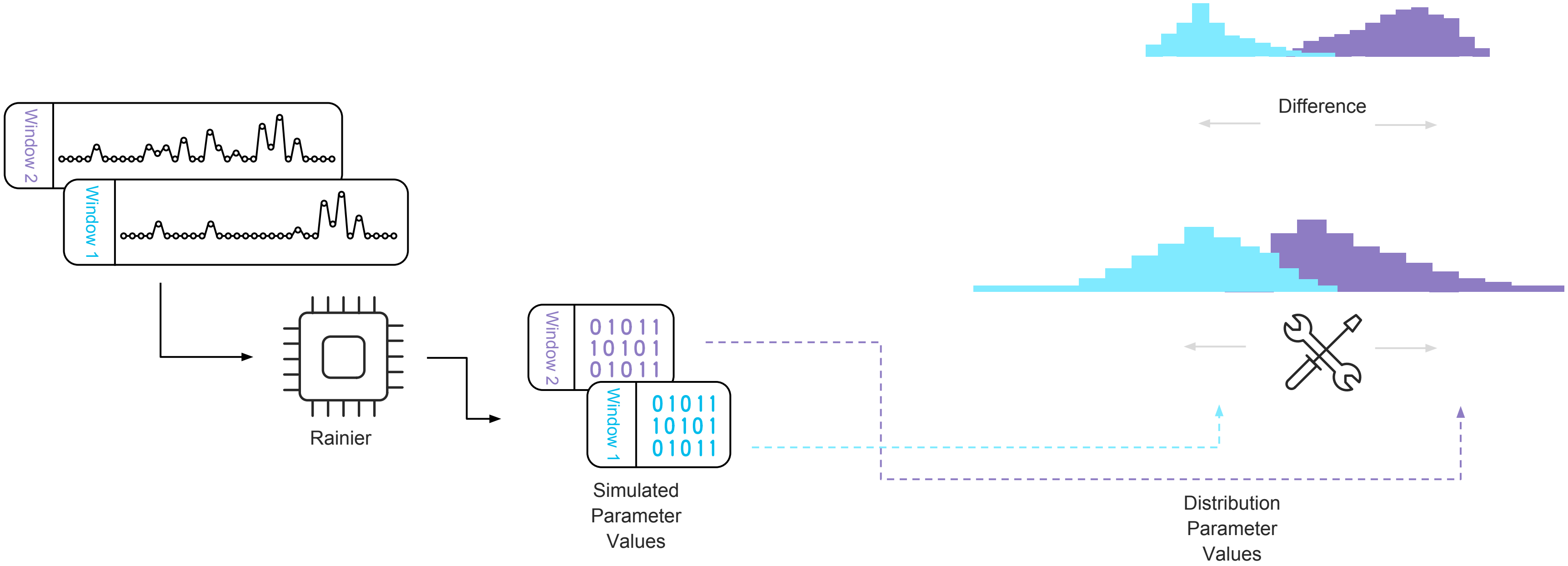
Masquerading
DNS
Traffic

Time Series
Modeling

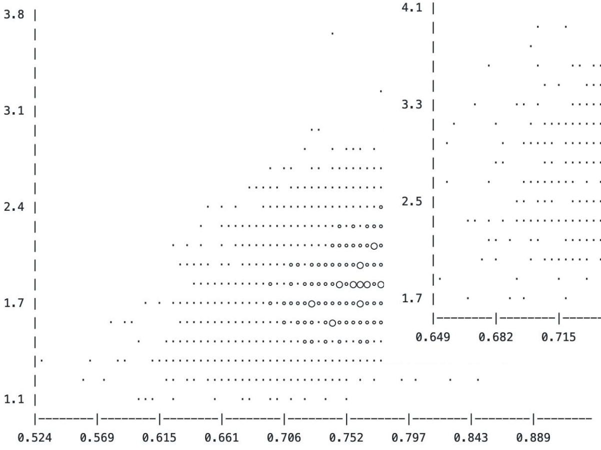
Rainier
+
Spark

Anomaly
Detection

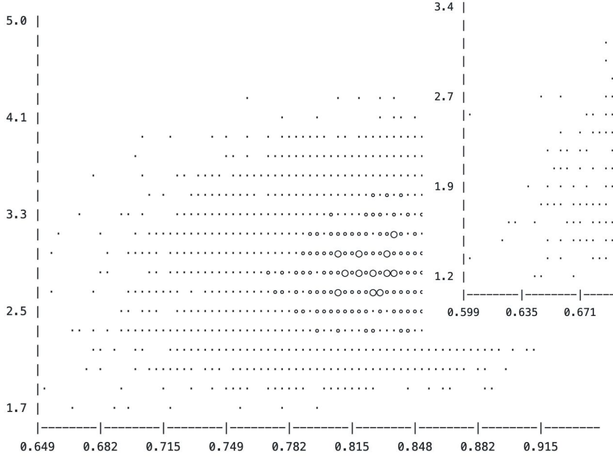
Part 4 Window Based



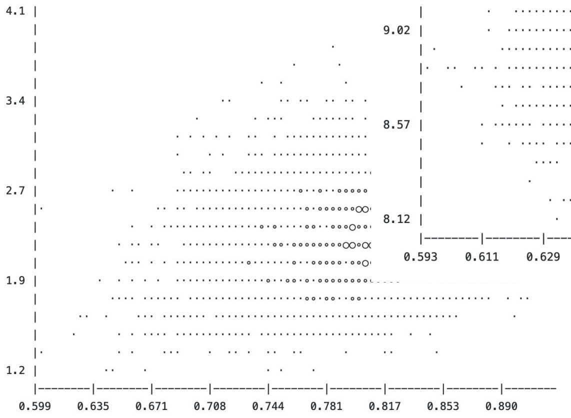
Part 4 Window Simulations



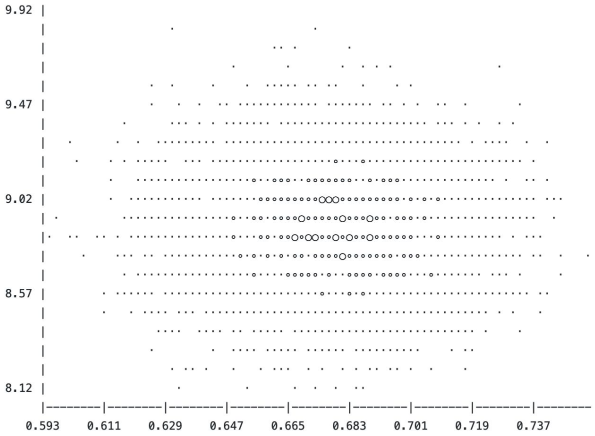
Week 1



Week 2

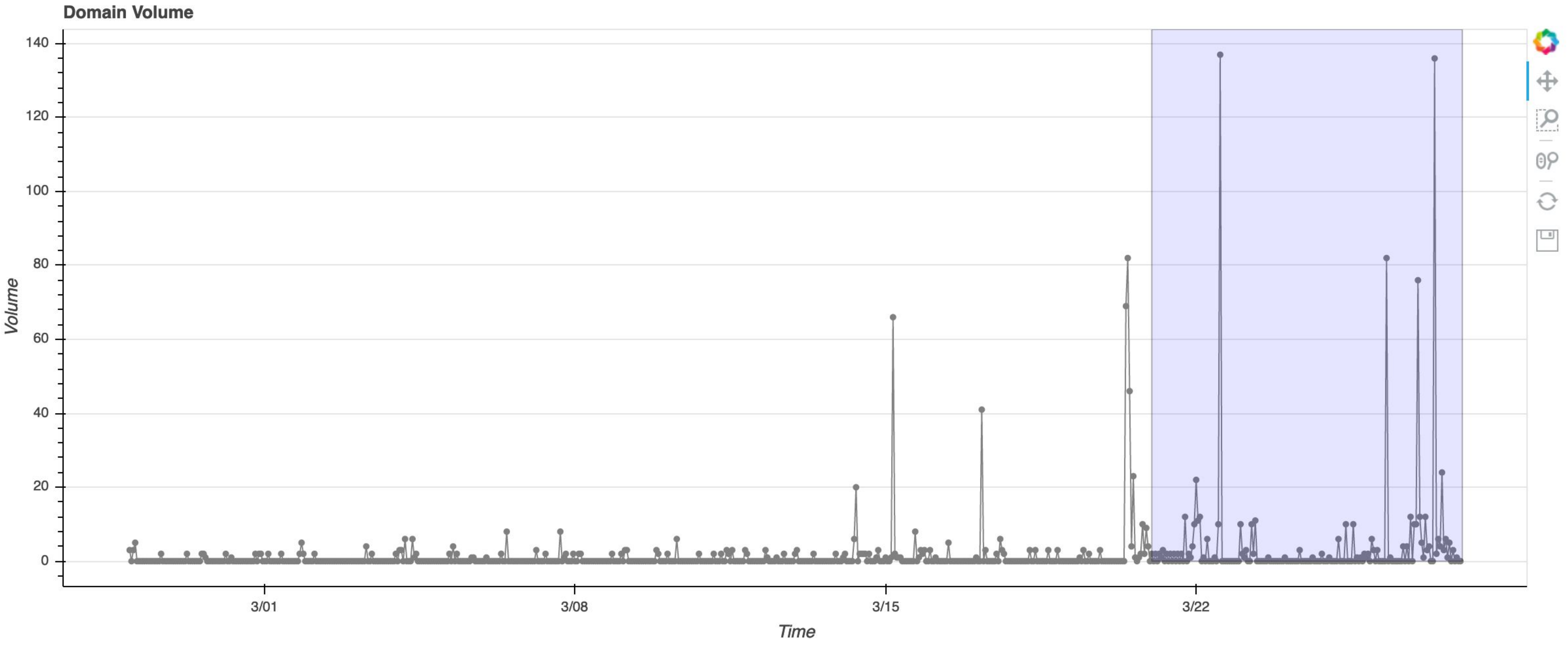


Week 3

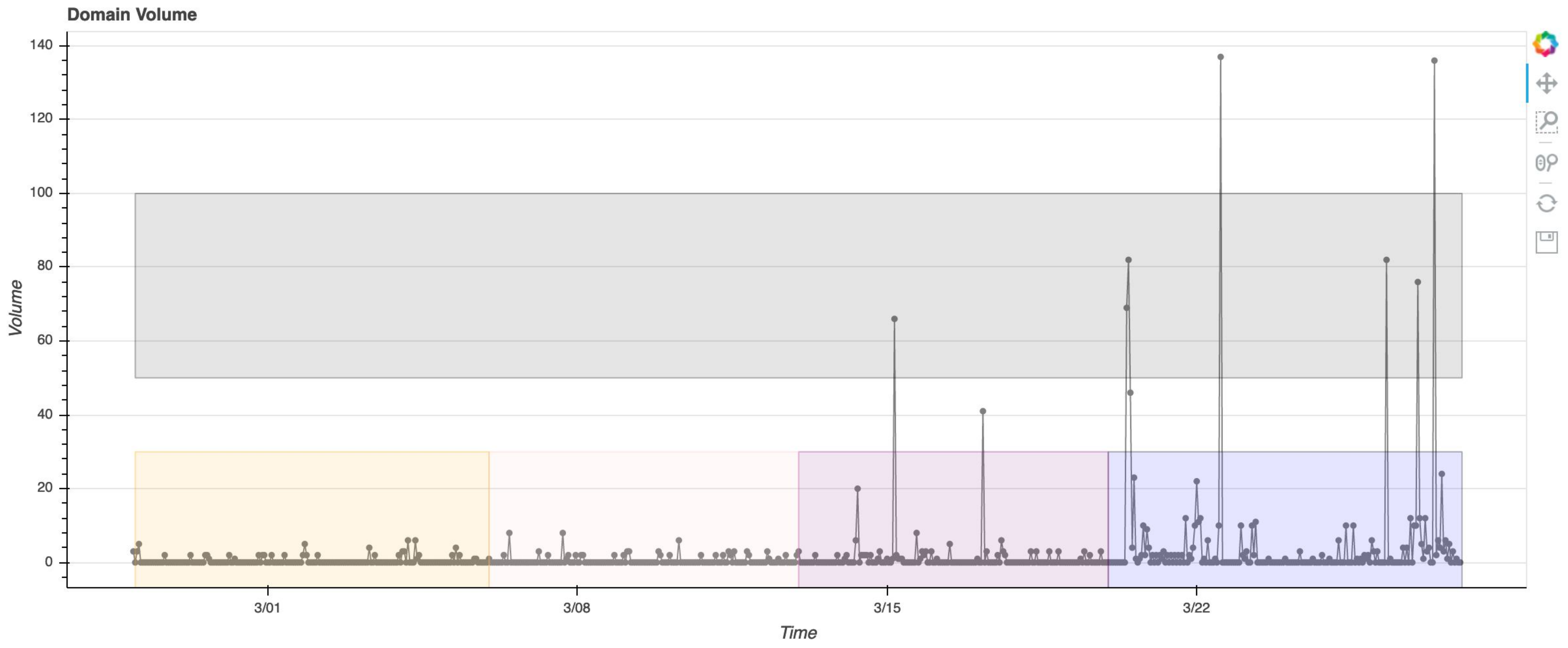


Week 4

Part 4 Outlier Window



Part 4 Local Outlier to Global



Closing Recap

Masquerading
DNS
Traffic

Time Series
Modeling

Rainier
+
Spark

Anomaly
Detection

Closing Glossed Over Details

Outliers

Goodness
of
Fit

Closing References

A Review of Croston's method for intermittent demand forecasting
https://www.researchgate.net/publication/254044245_A_Review_of_Croston's_method_for_intermittent_demand_forecasting

Rainier
https://github.com/stripe/rainier

PyMC3
https://docs.pymc.io/

Emotet
https://www.us-cert.gov/ncas/alerts/TA18-201A

Bokeh Plots
https://bokeh.pydata.org/en/latest/

Twitter Chill
https://github.com/twitter/chill

Closing Contact

Website
davidrdgz.github.io

Github
@davidrdgz

Twitter
@davidrdgz

Email
davrodr3 at cisco.com