

# NetBackup Anomaly Detection Configuration

Understanding data clustering and  
designing test patterns to validate  
anomaly detection configuration.

## Introduction

Anomaly detection serves as a watchtower inside the perimeter of data protection. With this feature, you can augment your data trending and coordinate the initial detection of suspicious activity. Any malicious software introduced will take up space and attempt to manipulate files, which often causes downstream impacts to data protection that can be observed if you know what to look for.

NetBackup™ anomaly detection knows the signs of breach, using data clustering, complex math, and statistical analyses to detect anomalies during backup operations. By analyzing backup jobs and comparing them against initial ingests into NetBackup, Anomaly Detection is able to provide an early view into any deviations from normal operations. To function properly and establish what “normal” is requires training (test) data. This presents challenges when the NetBackup Anomaly Detection service is new to the domain.

This paper describes how to overcome these challenges and design an anomaly test pattern to help you understand the complexities of the mechanics and algorithms. The goal is to minimize false positives and ensure the results find more relevant deviations that will assist you in bolstering your awareness and cyber-responses.

## Data Clustering

It all starts with data clustering. At the center of most statistical theories is the idea of a Normal Curve. Substantial amounts of data support the Normal Curve theorem, which states that a population (in our case that's a set of data), when plotted, will be a certain distance from the mean, or the average of all the data, in a predictable pattern. Each vertical line designates the number of standard deviations from the mean, known as a Z-score. This type of statistical modeling is used for television ratings, voting predictions, health data, and much more.

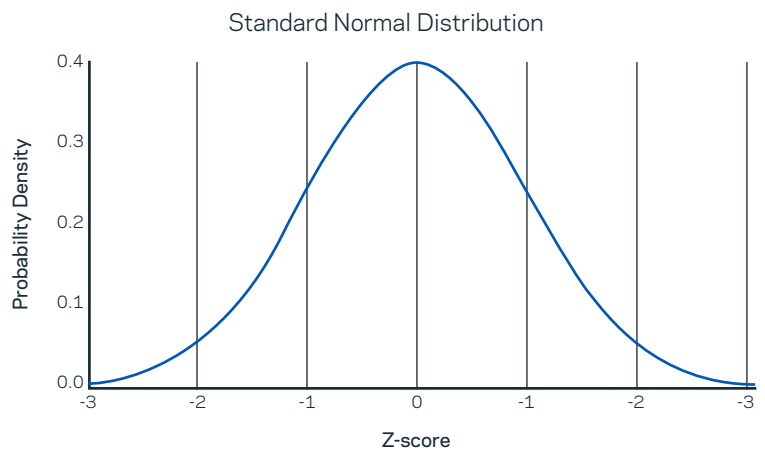


Figure 1. An Example of a Statistical Normal Curve with Standard Deviation Markers

NetBackup takes data clustering, which is a machine learning (ML) concept, and expands upon it to make it dynamic. The math is incredibly complex. The mean and standard deviation for each metadata point is constantly being recalculated to see where the new data compares to an established cluster and what its relationship is to other metadata points.

For this reason, an anomaly cannot simply be detected on the third or even tenth backup. There must be enough data to consider the learning period complete. Once deemed complete, the data cluster is bounded and can be used mathematically to determine what is outside those bounds and, therefore, a potential anomaly. The further away that data is from the center of the data cluster, the higher the anomaly score. For NetBackup, less than 10 is low, values between 10 and 15 are medium, and 15 or greater is considered a high anomaly score.

## Anomaly Detection

NetBackup anomaly detection must be properly enabled – choose Enable anomaly data gathering, detection services, and events. This will notify you of anomalies in backup jobs. In previous NetBackup versions, anomaly detection was performed using a configuration file, but this is now in the NetBackup WebUI (Web User Interface) to remove the possibility of introducing typos, unsupported values, or bad syntax (see Figure 2).

Figure 2. Anomaly Detection Service Configuration from NetBackup WebUI

## Anomaly Detection Details

An anomaly is comprised of one or more elements, data points, or observations from any of the following:

1. Backup images size
2. Data transferred
3. Deduplication rate
4. Total time for job completion
5. Number of files

The anomaly score is measured by the distance from the center of the data cluster to the normalized data point based on a normalized mathematical calculation. This is similar to converting different units to the same degree of measurement for relative comparison.

Figure 3 describes how this process works. The anomaly data gathering services provide the training data. The anomaly detection and alert services provide a closed loop, alerting the user and incorporating feedback into the training data. If the data gathering service is the only option enabled, the training data is populated, but no detection or alerting will occur. Likewise, if only data gathering and detection are enabled, no alerting events will be generated.

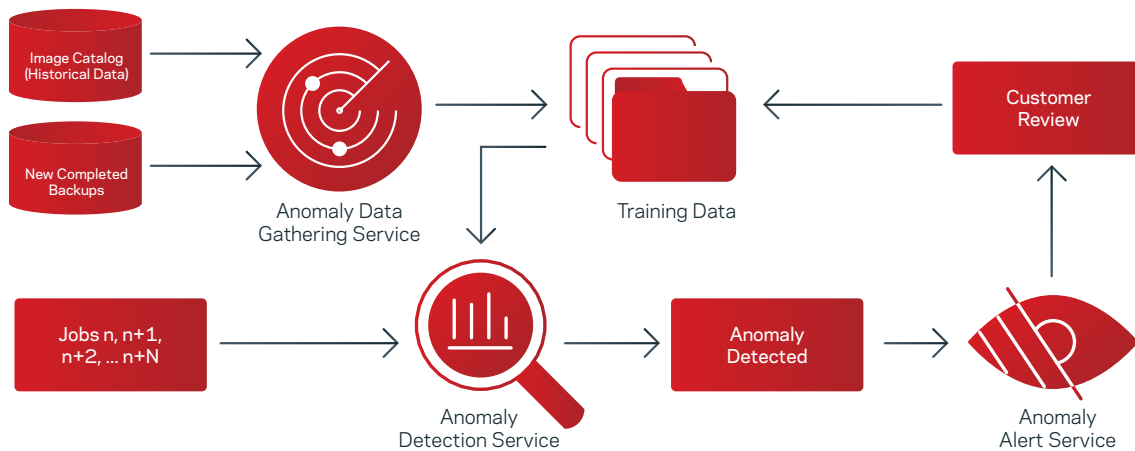


Figure 3. Anomaly Detection Process Flow

## Anomaly Data Gathering - Learning Cycle

The learning process for a workload is the last 90 days of backups, grouped by client, policy, policy type, schedule type, destination storage, and the unique asset ID in the NetBackup database. In the second phase of the learning process, training data is tested against the historical data. The next sequence of backups with the same attributes is used until 30 observations have been collected. Notably, incremental and full schedules will not be compared to each other. Since full schedules run less often than incremental schedules, it will take longer to establish that full schedule baseline under normal circumstances. If there are no historical backups to consider, the first 30 observations will constitute the entire set of training data. Once training data has been validated, new backup jobs will be compared to detect anomalies. Feedback allows the training data to be modified accordingly.

## Anomaly Data Gathering and Detection – Configuration

The sensitivity, data purge interval, and data gathering interval can be tuned for basic configuration items (see Figure 4). The anomaly detection sensitivity scale ranges from -100 to 100, with a default of 0 as the center point. Positive sensitivity values increase the data cluster size, while negative sensitivity values decrease the data cluster size. Points outside the data cluster, in either case, are detected and identified as an anomaly. Note, a point outside the data cluster for a particular positive sensitivity value may be inside the data cluster when the sensitivity value is skewed negative. This is similar to increasing or decreasing the search radius on a map.

The anomaly detection data gathering service can be tuned with a frequency as low as 15 minutes, but no more than 120 minutes. A frequency of 15 minutes is suitable for most circumstances. There could be instances that require more processing time, so the interval can be set to be less frequent.

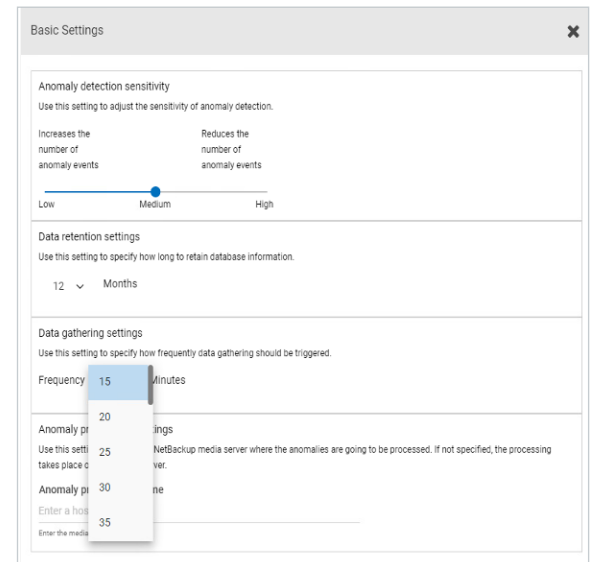


Figure 4. NetBackup WebUI Anomaly Detection Settings

By entering the media server hostname in the Anomaly proxy server, a single media server may be designated in the **Anomaly proxy server** box. Upon entering the media server, you can stop the anomaly services on the primary server, which will automatically start the service on the proxy host. Additionally, you can copy the existing anomaly database to the same location on the proxy host.

This is a specific use-case where the anomaly processing and anomaly database are on a NetBackup server, not the primary server. This is only needed for extremely busy NetBackup domains where primary server resources are a concern and anomaly detection has been identified as a potential contributor to resource constraints.

## Anomaly Data Gathering and Detection – Advanced Configuration

Advanced settings offer granular control options between specific hosts or policy types in your environment and anomaly detection services. Since every organization's environment is unique, configuration flexibility allows you to define what constitutes an anomaly to avoid false positive alerts.

One of the advanced settings available is the ability to add clients to an exclusion list (see Figure 5). The data of these clients' jobs is still available in the Activity Monitor, but this option completely ignores all observations from the selected client for anomaly detection. This setting is beneficial for clients whose data is known to be volatile, regardless of the policy type.

You can also exclude observations for certain policy types, which is helpful if a policy type consistently raises false-positive anomalies on a specific attribute. As shown in Figure 6, each observation can be excluded for everything from a particular policy type.

Further advanced options are available by modifying the configuration file on the Anomaly Detection server, the NetBackup Primary by default, or the media server acting as the proxy:

```
#/usr/opencv/var/global/anomaly_detection/anomaly_config.conf
#[install location]\Vetritas\NetBackup\var\global\anomaly_detection\
anomaly_config.conf
```

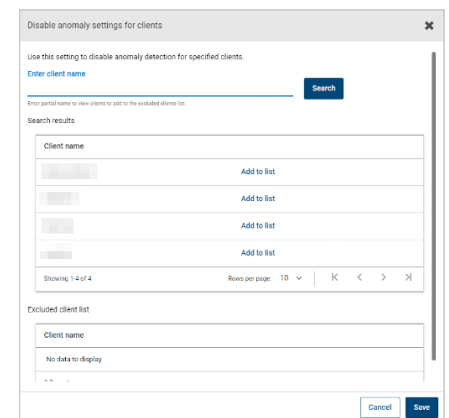


Figure 5. NetBackup WebUI Anomaly Detection Advanced Client Exclude Settings

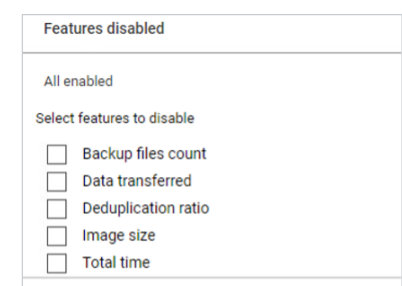


Figure 6. NetBackup WebUI Anomaly Detection Policy Type with Feature(s) exclude settings

First, to enable automatic scans, set:

```
ENABLE_AUTOMATED_SCAN=1
SCAN_HOST_POOL_NAME=[the scan pool to use for all automatic scans]
```

There are further options to enable batches of clients with specific scan pools for automatic scans included in the NetBackup Security and Encryption Guide.

To trigger automatic scans for LOW scores only, set:

```
TRIGGER_SCAN_FOR_LOW_SEVERITY=1
```

To trigger automatic scans for MEDUM scores only, set:

```
TRIGGER_SCAN_FOR_MEDIUM_SEVERITY=1
```

These above settings allow the user to add lower scoring tiers to automatic scans, and can be used together.

Alternatively, you can change the threshold score that triggers an automatic malware scan for compatible workloads. This setting will accept a decimal value that establishes the 'greater than or equal to' criteria. As a note, anomaly scores are rounded to the nearest hundredth place. The default value for a high score is 15:

```
TRIGGER_SCAN_FOR_SCORE_GREATER_THAN=##.##
```

Changes to this configuration file require that the anomaly detection services are restarted.

## Manually Starting and Stopping Anomaly Detection Services

For any need to restart, manually stop or manually start the NetBackup Anomaly Detection service, here are the CLI methods:

To manually stop the NetBackup Anomaly services:

UNIX/Linux:

```
# /usr/opensv/netbackup/bin/nbanomalygmt -stop
```

Windows

```
# [install path]\Veritas\NetBackup\bin\banomalygmt -stop
```

To manually start the NetBackup Anomaly services:

UNIX/Linux:

```
# /usr/opensv/netbackup/bin/nbanomalygmt -start
```

Windows

```
# [install path]\Veritas\NetBackup\bin\banomalygmt -start
```

## Testing Anomaly Detection

Designing a test requires preparation. Anomaly detection services only operate on backup jobs. Other types of jobs, such as restores, imports, duplications, or replications, do not share these metrics for data clustering algorithms. A backup policy with a consistent client, schedule, and storage unit is required to establish a baseline.

A common mistake would be to generate purposeful anomalies during testing before the learning and training cycles are complete. By introducing what would normally be an anomaly, while all data is being considered for training. This ends up adding the anomaly to the baseline, which means it will most likely not be detected as an anomaly going forward.

These are the logging directories associated with NetBackup anomaly detection:

UNIX/Linux

```
# [install path]/netbackup/logs/nbanomalygmt/  
# [install path]/netbackup/logs/nbanomalyalert/  
# [install path]/netbackup/logs/nbanomalydetect/
```

Windows

```
> [install path]\NetBackup\logs\nbanomalygmt\  
> [install path]\NetBackup\logs\nbanomalyalert\  
> [install path]\NetBackup\logs\nbanomalydetect\
```

These logging levels are controlled by the global logging level in the primary server's host properties, or `VERBOSE = {#}` in `bp.conf` on the primary server. An exception to this location would be when an anomaly proxy server is being used, in which case the services will run and log on that media server host.

Anomaly detection settings are updated according to the data gather interval. We can verify that settings are in place by checking the logs:

```
# [install path]/netbackup/logs/nbanomalygmt/  
  
{time} [#pid#] <2> DATABASE_TRACE: SELECT VALUE FROM CONFUGURATION_BASIC_SETTINGS WHERE CONFIG = "LIVE_  
DATA_GATHER_FREQUENCY_IN_MIN"  
{time} [#pid#] <2> DATABASE_TRACE: SELECT VALUE FROM CONFUGURATION_BASIC_SETTINGS WHERE CONFIG = "PURGE_  
DATA_INTERVAL_IN_MONTHS"  
{time} [#pid#] <2> nbanomalygmt: Using gather frequency as 15 mins  
{time} [#pid#] <2> nbanomalygmt: Using purge interval as 12 months
```

When anomaly detection services flag an anomaly, the alert service will find the flag and trigger the alert. We can verify the alert status of a detected anomaly by checking its ID in the logs:

```
# [install path]/netbackup/logs/nbanomalyalert/  
  
{time} [#pid#] <2> DATABASE_TRACE: UPDATE ANOMALY_DATA SET PROCESSED = 1 WHERE ANOMALY_UNIQUE_ID =  
'{###_#####}'  
{time} [#pid#] <2> DATABASE_TRACE: SELECT * FROM HIST_DATA WHERE CLIENT = '{someNBclient}' AND POLICY =  
'{somePolicy}' AND POLICY_TYPE = '{policyType}' AND SCHEDULE_TYPE = '{scheduleType}' AND SCHEDULE_NAME =  
'{scheduleName}' AND DESTINATION_STORAGE_NAME = '{storageunitName}' ORDER BY BACKUP_TIME ASC LIMIT 100  
{time} [#pid#] <2> DATABASE_TRACE: SELECT * ANOMALY_PER_FEATURE_DATA WHERE ANOMALY_ID = '{###_#####}'
```

```
{time} [#pid#] <2> nbanomalyalert: Executing API with path: netbackup/security/anomalies...
{time} [#pid#] <2> nbanomalyalert: Anomaly API execution done, return val =0
{time} [#pid#] <2> nbanomalyalert: Successfully sent alert message for anomaly id = {###_#####}.
```

With the mechanics of the anomaly detection services understood, let's design a testing scenario.

1. Assume the set of data is completely new, so that there is no historical data. It be from any static data set where you have the control to introduce changes that will be detected as anomalies, such as creating, modifying, or deleting files. For testing purposes, we recommend keeping this data set under 1 GB.
2. Create an entirely new backup policy with a single FULL schedule for the client with the data set. Use the same storage unit for all jobs. This can be any storage unit, but once you choose it, it must be kept consistent.
3. Next, run a backup of static data set at least 30 times. Each pass should be separated by a brief amount of time. The following command may be helpful to run many jobs in a script manually: `bpbbackup -i -p {policy} -s {schedule}`

The goal is to generate at least 30 identical backups, which will serve as the training data and ensure the 5 data points are closely clustered to reinforce a baseline. After the 30th backup, check the logs for “**completed getting training data set for detection.**” If this message does not appear, consider the data gathering interval and recheck the logs. If the message still does not appear, consider running five additional jobs and checking the logs again.

```
# [install path]/netbackup/logs/nbanomalydetect/

{time} [#pid#] <2> dbscan_detection: Getting training set for detection client - {someClient}, policy -
{somepolicy} and job - {someJobID}.
{timestamp} [#pid#] <2> dbscan_detection: Completed getting training data set for detection.
```

Next, in the logs, you can see the data is applied to the current job to see if it detects any anomalies. The DBSCAN function messages indicate the mathematical efforts being applied. In this example, the sensitivity is set to 0, so the optimal Epsilon value (Eps), which defines how close points must be to be considered as part of a data cluster, is not adjusted. The Eps value is not independently configurable.

```
{time} [#pid#] <2> dbscan_detection: Preparing input data for detection.
{time} [#pid#] <2> dbscan_detection: Completed preparing input data for detection.
{time} [#pid#] <2> dbscan_detection: Calling StandardizeData.
{time} [#pid#] <2> dbscan_detection: Completed StandardizeData activity.
{time} [#pid#] <2> dbscan_detection: Calculated eps = 4.000000
{time} [#pid#] <2> dbscan_detection: Using sensitivity parameter 0
{time} [#pid#] <2> dbscan_detection: Adjusted eps = 4.000000
{time} [#pid#] <2> dbscan_detection: Calling DBScan for detection for job {someJobID}
{time} [#pid#] <2> dbscan_detection: Done detection.
{time} [#pid#] <2> dbscan_detection: Anomaly not detected.
```

The following procedure introduces deliberate changes to multiple data points to generate an anomaly for testing.

1. Double or triple the number of files, by introducing unique files into the same location as the established backup selection in the policy.
  - a. This can be done by introducing a long string of random numbers and letters into the desired number of files.
  - b. This could also be done by introducing a different file type, such as an audio or image file.
2. Be sure you have confirmed that the training data is completed by checking the logs.
3. Do not make any changes to the test policy.

4. Make all these changes before running a new job from the same policy for the same client.
5. There will be at least a 15-minute delay before any anomaly is reported, since this is the data gathering minimum interval.
6. Observe the alerting mechanism in the NetBackup webUI.
7. Observe the score that your manually created anomaly returns.
8. If you wish to repeat this test, be sure to also confirm this as an anomaly, and consider using all new unique data for the next iteration of the test.

By increasing the number of files by this volume, you will significantly change the number of files in the backup. By further introducing unique data, you will increase the size of the backup, increase the data transferred, decrease the deduplication rate, and increase the total time for the backup to complete. It is not required to change all five data points to generate an anomaly. Still, for testing purposes, this designs a test that illustrates how multiple data points contribute to the anomaly, and demonstrates how it is presented in the NetBackup webUI anomaly detection display.

## Anomaly Feedback Mechanism

Providing feedback on reported anomalies helps train the detection engine. For example, you can take the anomaly and Mark as **ignore**, **Confirm as anomaly**, or **Report as a false positive** (see Figure 7).

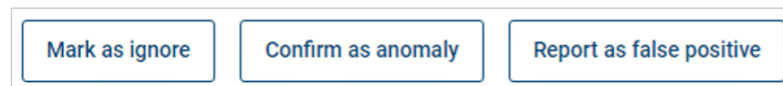


Figure 7. Anomaly Feedback Options

- **Mark as ignore:** by ignoring the data point, the detection and data gathering services ignore that data point for future data clustering. An example of when you may want to use this option is when testing new workloads, where the data is expected to be out of bounds and should not be considered as it moves into production.
- **Confirm as anomaly:** by selecting this option, you strengthen the data clustering algorithm. This feedback option should be used regardless of the nature of the anomaly's investigation, so future alerts are based on similar observations.
- **Report as false positive:** by marking an anomaly as a false positive, the data cluster is re-factored, or a new data cluster is created to include this new data point as if it were training data. Use this option to improve the acceptable range of data for a particular client + policy combination.

Always consider the feedback mechanism's impact on the training data. For example, while a bad actor or malignant software may not cause an alert, you want to be alerted to scenarios that are suspicious, so independent, secondary action can be taken. In addition, you may want to use it to implement the malware detection feature in NetBackup for supported workloads.

Data clusters are changing based on more data points falling closer to the central point of the cluster, and by anomaly feedback mechanisms. For example, suppose you have 90 days of backup history, plus 30 training observations from consistent backups always running a "Full" schedule type, where the values for each observation change only by insignificant amounts. In that case, the data cluster is very tightly packed - this is considered justified based on the math. If a more significant change in one or more of the observations were calculated, it will likely produce an anomaly. Conversely, if the historical and training data shows a wide range of acceptable values, an anomaly will be harder to detect, since the observable changes in the data during the training period established a broad set of acceptable values. This is why no single data point from a backup job is relied upon to identify an anomaly. Instead, five data points are used simultaneously for every backup job.



## Summary

Understanding what an anomaly is, how one is detected, and ways to train the detection engine with feedback mechanisms can improve your organization's ability to respond to threats more appropriately. Avoid the common mistake of introducing a purposeful anomaly during the learning cycle. As changes are made to existing production policy attributes, new observations are generated and incorporated into the data clustering algorithms, making feedback during this time especially important. With the addition of new workloads, the learning cycle must be respected to produce reliable results in the lifetime of that workload.

It is always wise to investigate detected anomalies in production. Even if there was no malignant cause, you still want to be alerted to similar circumstances in the future. Therefore, you should incorporate anomaly detection as an additional trigger to your cybersecurity response plan. Treat anomalies as potential symptoms of a broader range of infections. For example, in a medical context, a fever alone does not indicate a course of treatment. Still, it serves as a strong indicator to be considered along with other symptoms and warrants further investigation. Investigations can give you insight into your data and reveal potential malignant software or bad actors on the host that need to be remediated.

## About Veritas

Veritas Technologies is a leader in multi-cloud data management. Over 80,000 customers—including 95 percent of the Fortune 100—rely on Veritas to help ensure the protection, recoverability, and compliance of their data. Veritas has a reputation for reliability at scale, which delivers the resilience its customers need against the disruptions threatened by cyberattacks, like ransomware. No other vendor is able to match the ability of Veritas to execute, with support for 800+ data sources, 100+ operating systems, 1,400+ storage targets, and 60+ clouds through a single, unified approach. Powered by Cloud Scale Technology, Veritas is delivering today on its strategy for Autonomous Data Management that reduces operational overhead while delivering greater value. Learn more at [www.veritas.com](http://www.veritas.com). Follow us on Twitter at [@veritastechllc](https://twitter.com/veritastechllc).

# VERITAS™

2625 Augustine Drive  
Santa Clara, CA 95054  
+1 (866) 837 4827  
[veritas.com](http://veritas.com)

For global contact  
information visit:  
[veritas.com/company/contact](http://veritas.com/company/contact)