



Predictive monitoring



From:

<https://pandorafms.com/manual/!776/>

Permanent link:

https://pandorafms.com/manual/!776/en/documentation/pandorafms/monitoring/10_other_monitoring

2024/06/10 14:34



Predictive monitoring

We are working on the translation of the Pandora FMS documentation. Sorry for any inconvenience.

Introduction

In addition to features such as remote, Agent-based or web monitoring, Pandora FMS offers advanced resources to improve monitoring. With these resources you can make estimates on historical data or create new Modules based on arithmetic operations of existing Modules.

Types of predictive monitoring

- Predictive monitoring:
 - *Capacity planning*: Makes a prediction based on the time window specified by the user, assuming a more or less linear behavior of the target module. This type of predictive modules allows us to know how many days we have left until the disk is full, or the number of requests to the database that we will have within a month, if we continue as before. These modules replace the old prediction modules.
 - *Service*: Recovers the value of a service to be able to display it in any Agent where it is necessary.
- Arithmetic monitoring:
 - *Synthetic arithmetic* : It is about being able to perform arithmetic operations (addition, subtraction, multiplication and division) with data previously obtained in other Modules.
 - *Synthetic average*: This involves taking an average of data previously obtained in other Modules.
 - *Trend* (Trending module): Compares the current average with the average of the previous period and returns the difference in absolute value or as a percentage. The Trending module makes the average of the last period at the indicated periodicity versus the average of the same period a previous day/week/month. For example, if you select a week, Trending module calculates the average of the last week and compares it with the average of the previous week.

Monitoring with synthetic modules

E

Synthetic Modules are Modules manufactured from data from other Modules, which can be on the same Agent or on different Agents. The operations that can be performed are arithmetic (add, subtract, multiply and divide) between Modules and/or with absolute values.

Synthetic Modules are managed by the [prediction server](#).

This subcomponent of the Pandora FMS server must be active and running. Likewise, the Agent that will contain the synthetic Modules must use a Prediction Server. Remember that you can also use a [High Availability Environment](#) and have load balancing on those servers.

In the administration section of an Agent in the Modules tab, access it by clicking on Create module and select Create new prediction server module and complete the requested fields.

For other logical operations (multiplication, subtraction, division) the order of the operators must be taken into account. Try the interface to learn how any arithmetic operation can be done between different Modules.

Anomaly detection (MADE)

Introduction to MADE

The final purpose of the Pandora FMS Anomaly Detection Engine (MADE) is the training and use of Artificial Intelligence models for automatic anomaly detection. To train these models, large amounts of input data are needed, which are obtained from Pandora FMS database. MADE keeps a copy of this data on disk to carry out retraining and resampling tasks in feather format, designed for efficient data storage.

Since models are loaded into memory and written to disk relatively frequently, trained models are stored on disk serialized with the data for simplicity and efficiency. The format in which they are stored may vary depending on the implementation details of each model. As we will see later, MADE also writes information related to anomalies and its own state to the database.

MADE generates as a result events in Pandora FMS, indicating whether it detects an anomaly in a specific monitor.

MADE configuration

Download links for MADE, for EL8:

https://firefly.pandorafms.com/centos8/pandorafms_made-0.1.0-1.el8.x86_64.rpm

For Ubuntu server:

https://firefly.pandorafms.com/ubuntu/pandorafms-made_0.1.0-2_amd64.deb

To activate and customize MADE, add the following configuration options to Pandora FMS server configuration file, `/etc/pandora/pandora_server.conf`:

```
# Enable (1) or disable (0) the Monitoring Anomaly Detection Engine (MADE).
madeserver 1

# Directory where models will be stored.
madeserver_path /var/spool/pandora/data_in/models

# Number of server threads for MADE.
madeserver_threads 2

# Model backend: 'prophet' or 'iforest'.
# 'prophet' is better suited for temporal series and supports forecasting.
# 'iforest' is faster and more efficient (cpu, memory...).
madeserver_backend prophet

# MADE will query the Pandora FMS database every makeserver_interval seconds
# to look for new data.
madeserver_interval 10

# Minimum number of data required to train a model (e.g., '7d' for seven days).
madeserver_min_train 7d

# Maximum number of data kept to train models (e.g., '90d' for 90 days).
madeserver_max_history 90d

# Model automatic retraining period (e.g., '7d' for seven days).
madeserver_autofit 7d

# Model sensitivity. A lower value triggers less anomalies.
madeserver_sensitivity 0.1
```

Help on MADE can be obtained by running the command:

```
pandora_made -h
```

MADE runs as a daemon managed by systemd. Installing the RPM or DEB package enables the service, but to start it without restarting the server it needs to be run:

```
systemctl start pandora_made.service
```

Either:

```
service pandora_made start
```

If the system restarts or crashes, systemd itself restarts the service.

Model training may be forced using data previously acquired by Pandora FMS with the command:

```
pandora_made -c /etc/pandora/pandora_server.conf -t
```

It is also possible to force the training of a specific model, specifying the identifier of Pandora FMS module with `-m`:

```
pandora_made -c /etc/pandora/pandora_server.conf -t -m 1
```

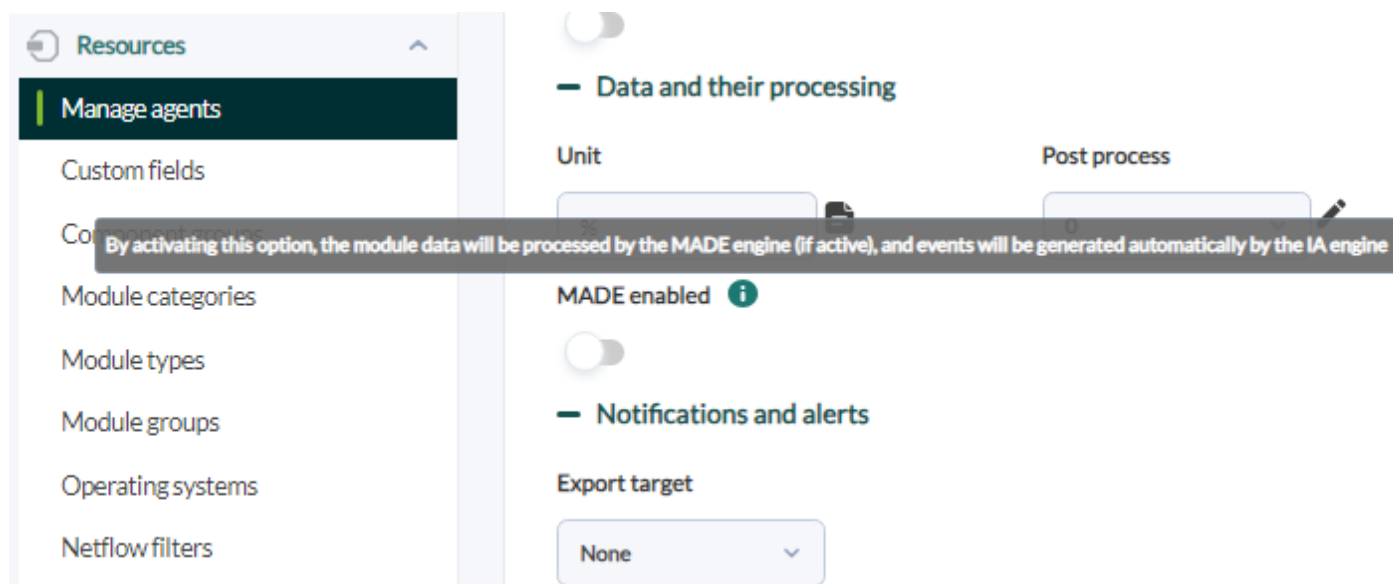
When retraining a model, MADE evaluates it and compares its performance with the current model, always keeping the best model. You may force the deletion of old models with the command:

```
pandora_made -c /etc/pandora/pandora_server.conf -d
```

You may find it convenient to run this command periodically from cron.

MADE configuration at module level






Once MADE has been installed and configured at a general level, in each numerical module there is the following selector to add that module to the data processing task:







The screenshot shows the configuration interface for a module. On the left, a sidebar lists various configuration options under the 'Resources' section, with 'Manage agents' selected. The main content area shows a configuration panel for 'Data and their processing'. A toggle switch for 'MADE enabled' is turned on. Below it, there is a section for 'Notifications and alerts' and an 'Export target' dropdown menu set to 'None'. A tooltip is displayed over the 'MADE enabled' toggle, stating: 'By activating this option, the module data will be processed by the MADE engine (if active), and events will be generated automatically by the IA engine'.

After a certain period of time and upon detection of an anomaly, MADE will publish its own **events** in a specific category:

Anomaly detected for module Connections opened: 212.0
✕

 General
 Details
 Agent fields
 Comments
 Responses

Event ID	#13566
Event name	Anomaly detected for module Connections opened: 212.0
Timestamp	October 26, 2023, 5:05 pm
Owner	
Type	SYSTEM 
Duplicate	No
Severity	Informative 
Status	New event 
Acknowledged by	N/A
Group	Servers 
Contact	N/A
Tags	N/A
Extra ID	N/A
Module custom ID	N/A

See also the [event alert system](#).

Anomaly detection

Once the service is installed and started, MADE works automatically. MADE reads data from Pandora FMS, resamples and rotates it when necessary, trains models when it has enough data, re-trains them periodically, and generates events when it detects anomalies.

Indicate in which modules you wish to activate anomaly detection. No further configuration is required other than activating it in each module, in the advanced settings section:

The system is intelligent and will perform model training for each data set and generate a detected anomaly event.

Such events can be captured like any other PFMS event to generate customized notifications through [event alerts](#).

Considerations on the different applied IA models

MADE is a useful tool to draw attention to certain patterns that would be very difficult for an administrator to detect or predict.

Prophet mode allows more robust models to be trained, which take into account the time characteristics of the data series and allow predictions to be made in the future, but they can be expensive to train in very large environments. It is the recommended default backend to use.

IsolationForest mode is much more resource efficient and has generated satisfactory results during testing, but this may vary depending on the environment and data. Its use is recommended when Prophet mode causes performance losses due to lack of hardware resources.

[Return to Pandora FMS Documentation index](#)