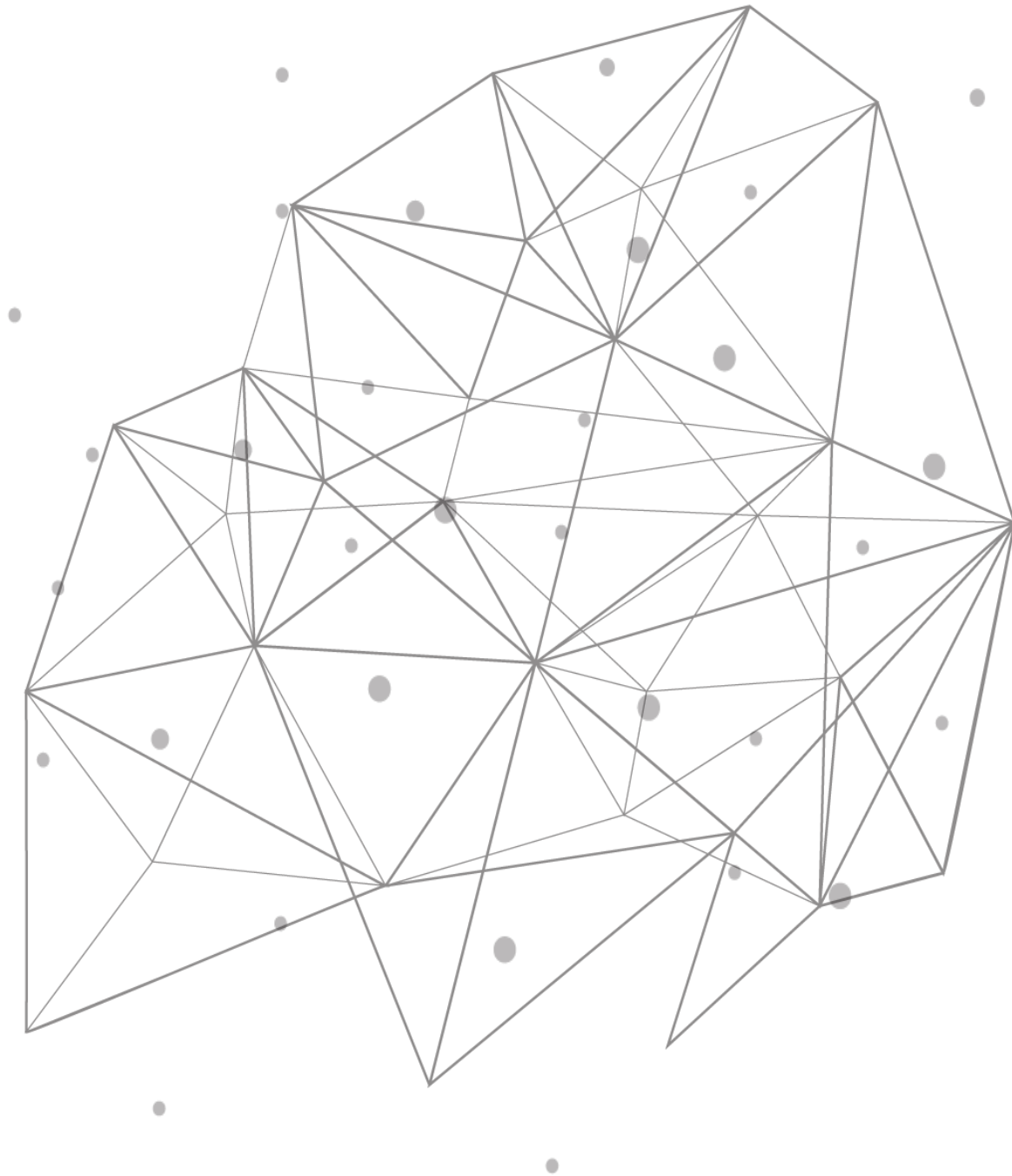


TCPWave DDI – XGBoost Model



Introduction

DNS is a central part of internet operations. The primary use of the DNS is to translate IP addresses to human-friendly names; this makes it a critical component of business operations. As a result, it has become the main target for malicious actors. The malicious actors have successfully deployed various DNS-based attacks, such as the application of Domain Generation Algorithms (DGA) to command and control a group of Internet of Things (IoT) or tunneling techniques. Despite the continuous progress in detecting DNS attacks, there are still weak spots in the network security infrastructure. Hence, it is imperative to implement new ways to analyze data, particularly machine learning (ML) algorithms. Various security teams urge to integrate ML-based DGA detection within the organizations that provide high-performance accuracy based on the data. TCPWave's DNS [TITAN](#) solution provides to combat and safeguards the DNS from the attacks. It uses In-House built tunnel detection Machine Learning (ML) algorithms trained using massive ~3.3M records and varied DNS data, thereby learning and detecting the malicious DNS traffic flowing through the DNS pathways in the enterprises. This white paper provides insights on how TCPWave's Threat Intelligence uses one of the ML models - XGBoost to detect and mitigate the DNS anomalies.

Machine Learning

It is one of the subsets of Artificial Intelligence (AI). It finishes the task of learning from data with specific inputs to the machine. There are several stages in the ML lifecycle, and a few of the major ones are:

Name	Description
Data Management	Collect, Clean, Visualize, and Feature Engineering the data.
Model Training	Process in which the ML algorithm is fed with training data and learns the best fit from the data features.
Model Evaluation	Process in which the ML model performance is evaluated based on specific metrics such as accuracy, precision, etc.
Model Deployment	Process in which the ML model is integrated into an existing application.

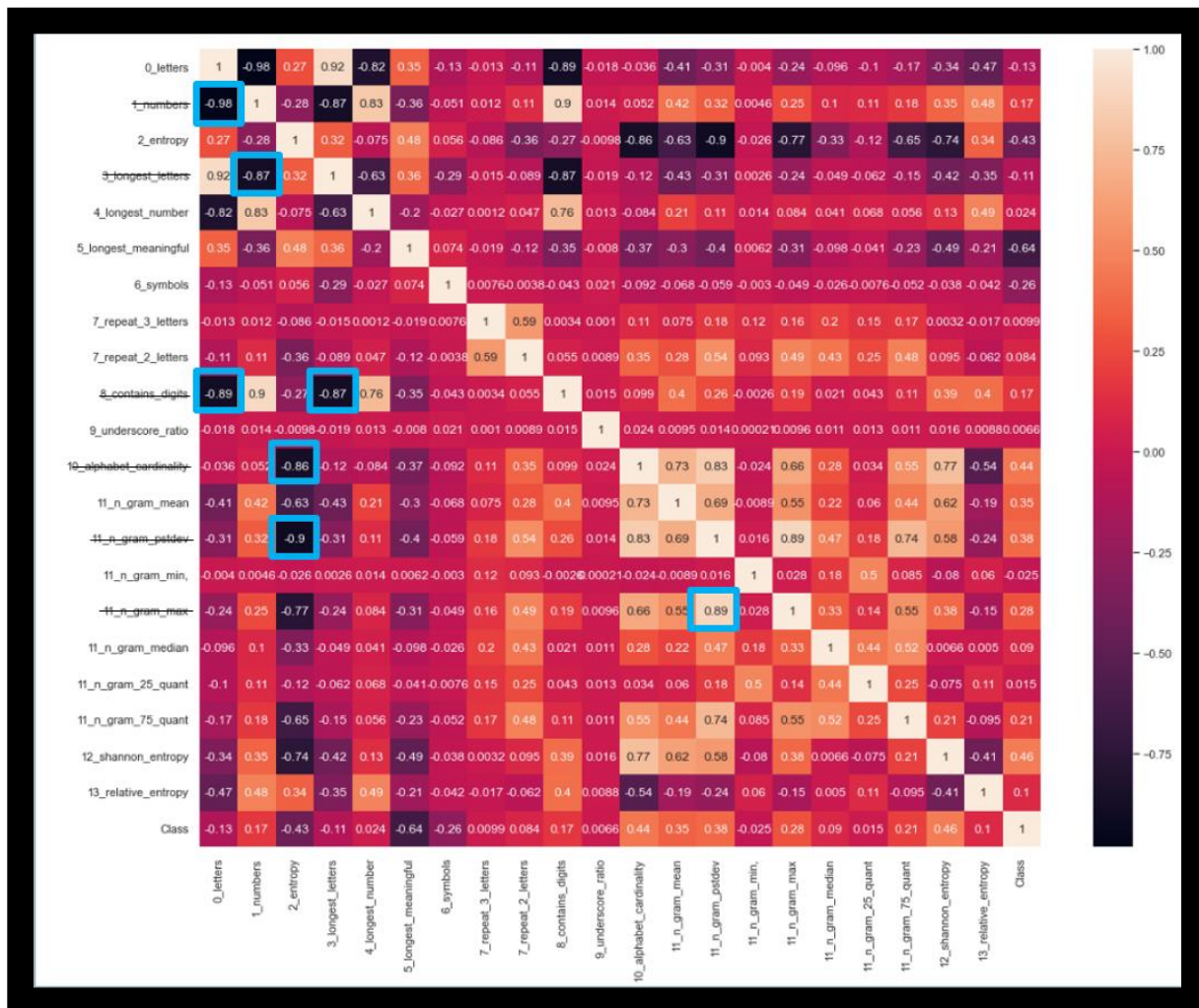
Data Management

Dataset is one of the main components in building an ML model.

- The data consists of different DGA families. The input dataset is split into training data and test data.
- The training data consists of 2.1M data points, and the test data consists of 0.9M points.

Feature Engineering

It is the process of extracting the features from the raw data to capture the data properties and give them as input to the ML model. TCPWave’s ML Engineering team has obtained some of the [features](#), and the features are further filtered by conducting multicollinearity checks; below is the plot of the features. [Click here](#) to know about the feature selection.

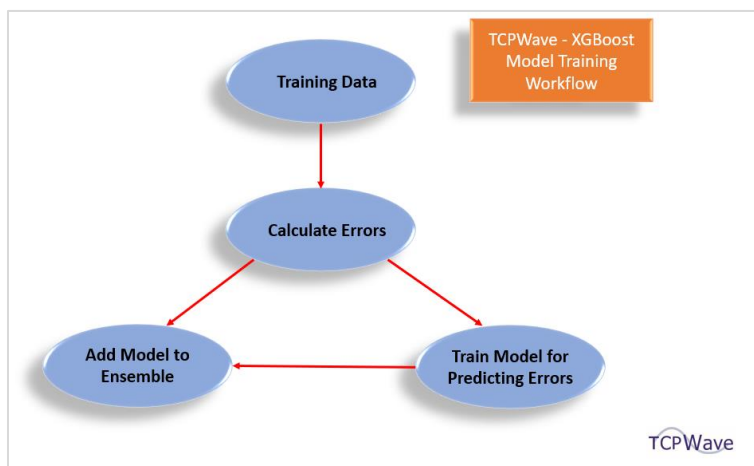


Model Training

TCPWave’s ML Engineering team used the above data management and feature engineering steps to train and evaluate an XGBoost model.

About XGBoost

- [XGBoost](#) stands for eXtreme Gradient Boosting.
- It is one of the well-known boosting techniques(ensemble) in ML. The following is the algorithm workflow.
- TCPWave ML Engineering team has used ASAI 2019 dataset to train the ML model - XGBoost.



To improve the ML model’s accuracy, the TCPWave’s ML Engineering team has used the RandomizedSearchCV - one of the best hyperparameter tuning methods. In this strategy, each iteration tries a random combination of hyperparameters. It records the performance and returns the combination of the following hyperparameters that gives the best performance:

Hyperparameter	Description
learning_rate	It is the regularization parameter step size to avoid overfitting. It shrinks feature weights in each boosting step.
n_estimators	It is the number of trees that is boosted.
max_depth	It is the maximum depth of the decision tree.
colsample_bytree	It is the fraction of features selected randomly from all the features.
subsample	It is the ratio of training instances. (Example: 0.5 means 50% data used before growing trees)

Model Evaluation

Based on the best hyperparameter combination, the XGBoost's innovative features and algorithmic optimizations rendered an accuracy close to the Atlantis model.

Note: [Atlantis](#) is a hybrid model whose deep learning architecture is designed using Convolution Neural Networks (CNN) layer and a Long- and Short-Term Memory (LSTM) layer in parallel.

Metrics	Description	Percentage (%)
Accuracy	It represents the number of correctly classified data points over the total number of data points.	93.66
Precision	It is the ratio of true positives to the sum of true positives and false positives.	94.54
Recall	It is the ratio of true positives to the sum of true positives and false negatives.	95.26
AUC	It is the score representing the area under the curve of ROC (receiving operating characteristics) and is the measure of the ability of a classifier to distinguish classes.	93.15
F1 score	It is the harmonic mean of precision and recall.	93

Configuring XGBoost Model in TCPWave IPAM


XGBoost Model is categorized as an ML model for anomaly detection in TCPWave IPAM. To configure it:

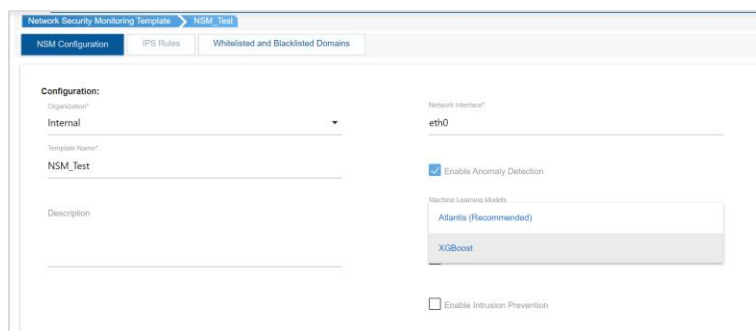
1. Create a Network Security Monitoring (NSM) Template, and enable the Anomaly Detection option.
2. Associate the NSM Template with one or more DNS appliances. An NSM Templates can be associated with the following types of appliances:
 - ISC BIND Authoritative appliances with recursion enabled
 - ISC BIND Cache appliances
 - Unbound Cache appliances

Also, an NSM Template can only be associated with an appliance in the same Organization as the Template.

Detailed information on these two configuration steps is provided in the subsections below.

Create an NSM Template

1. Go to **Network Management > DNS Management > DNS Security > DNS Threat Management**.
2. Select the **NSM Templates** tab, and then **Add** .
3. On the **NSM Configuration** tab, enter values as needed, including the following:
 - a. Select the organization from the drop-down.
 - b. Enter the Template Name.
 - c. Enter the **Network Interface** name, such as eth0.
 - d. **Enable Anomaly Detection**. The system displays the following Machine Learning models drop-down:
 - i. Atlantis
 - ii. XGBoost



- e. Select XGBoost model.
- f. (Optional) Select **Enable Intrusion Prevention**.
- g. (Optional) Enter **Rule Variables** for source and destination IP addresses and ports used in rules.
- h. Select **OK**.

Associate an NSM Template with a DNS Appliance

1. Go to **Network Management > DNS Management**.
2. Select an **Appliance Name** to edit that Appliance's configuration.
3. In the **Template Attributes** section, select an **NSM Template**.
4. Select **OK**.

Anomaly Detection Service

1. The system now initializes the anomaly-detection service as shown:

```
[root@TCPWaveRemote-180 ~]# monit summary
Monit 5.26.0 uptime: 15d 19h 12m
```

Service Name	Status	Type
localhost	OK	System
ntpd	OK	Process
named	OK	Process
timsdns	OK	Process
sshd	OK	Process
snmpd	OK	Process
crond	OK	Process
anomaly-detection	Initializing	Process

2. The updated python script from the path `/opt/tcpwave/timsdns/anomalydetection/` sniffs the traffic and logs the suspicious traffic in the path: `/opt/tcpwave/timsdns/logs/tcpwave-anomaly-detect.log` and generates the alerts in the Current Alarms section of IPAM as shown:



3. To block the anomalous traffic, the network administrators can enable the following global options:
 - a. Automatically block anomalous traffic on DNS caches: By default, this global option is set to No; you must set it to Yes to block the anomalous traffic.
 - b. Anomalous traffic blocking methodologies using either Suricata IPS or DNS Blackhole mechanism.
4. Once the above global options are set to Yes, you can remove the blocked sources after the specified time interval (hours) using **DNS Network Security Monitoring Autoblock Purge Interval**. By default, the time interval is 24hrs.

Conclusion

The never-ending battle persists with the complexity of the threats changing as quickly as innovation grows. Many organizations are working hard to overcome DNS anomalies by introducing new configurations or developing in-house ML models. Hence, the organizations look forward to having a framework that ensures all the facets of the security ecosystem are safe and secure. TCPWave's XGBoost model is one of the best solutions to detect and mitigate DNS anomalies. For a demo, contact the [TCPWave Sales Team](#).

Appendix

FANCI: TCPWave's ML Engineering team obtained some of the features using this system.

Features

Name	Description
Letters Ratio	The ratio of the number of letters to the total number of characters.
Digits Ratio	The ratio of the number of digits to the total number of characters.
Metric Entropy	Shannon's entropy is divided by string length.
Longest letters string ratio	The ratio of the number of consecutive words to the total number of characters.
Longest number string ratio	The number of consecutive numbers to the total number of characters.
Longest meaningful word ratio	The ratio of the meaningful word (obtained enchant python library) to the number of string characters.
Symbols ratio	The ratio of the number of symbols in the string to the length of the string.
Repeat 3 Letters	Count of the words that are repeated thrice.
Repeat 2 Letters	Count of the words that are repeated twice.
Contains Digits	One, if contains digits else 0.
underscore_ratio	The ratio of the number of underscores to the number of letters.
alphabet_cardinality	Returns the total number of characters in the query.
n_gram_frequency_dist	returns a list of following: [mean, standard deviation, min, max, median, 0.25 quantile, 0.75 quantile]
shannon_entropy	Calculates Shannon entropy for data.
relative_entropy	Calculates relative entropy for data.