UDC 519.21

USING MACHINE LEARNING TO DETECT THREAT ANOMALIES FOR REDUCING FALSE-POSITIVES ON THE DAILY CYBERSECURITY OPERATION CENTRE ROUTINE

R. Karpiuk[1,2], P. Venherskyi[2]

[1] *CSOC, SoftServe Inc., e-mail: simmppllee@gmail.com*
[2] *Ivan Franko National University of Lviv,*
Universytetska str., 1, Lviv, 79000, e-mail: petro.vengersky@gmail.com

With machine learning (ML), we are able to detect a variety of cybersecurity threats, such as brute force, abnormal growth or decline in network traffic, monitor end-user infections with malware, or detect attacks on critical infrastructure, such as AD, DNS. The main advantage of using ML for such scenarios is the accuracy of detection of certain anomalies. This, in turn, significantly reduces the financial cost of cybersecurity in the organization and the speed of countering attackers.

Key words: Machine Learning (ML), Security Information and Event Management (SIEM), Splunk, density function, correlation search, analysis, cybersecurity, cybersecurity operation center (CSOC)

1. Introduction

Every day, cybersecurity operation centers (CSOC) face the need to find a balance between the number of professionals who can analyze cybersecurity events to the number of those events. We will focus on how to reduce the burden on analysts, namely, how to reduce the number of false positives. What is the main source of input for the analyst? That's right, correlation rules. What does the Threat Detection engineer first face when it wants to improve the response time of a CSOC – by reducing the number of false positives generated by correlation rules. What can be done for this? Give up static thresholds and use statistics instead. The idea is good, but not very effective, because the ecosystem in which cybersecurity operates is extremely dynamic and there is a high probability of "loss from the radar" is something extremely important due to changes in the behavior of one of the controlled objects (end devices, servers, network equipment or other). And when the statistics no longer meet your requirements in that case comes machine learning. Therefore, all improvements and construction of interaction with ML will be carried out based on the SIEM "Splunk".

2. Model problem

Consider the problem of transforming the usual static or statistical methods of detecting anomalies (correlation rules) to a method of applying machine learning to search for harmful patterns.

3. The main research

First of all, when trying to improve the detection of threats, as well as the quality of correlation rules, it is necessary to abandon statistical thresholds, and move to the

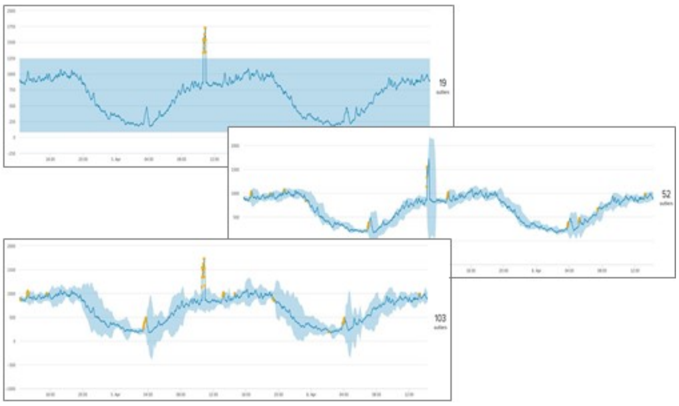
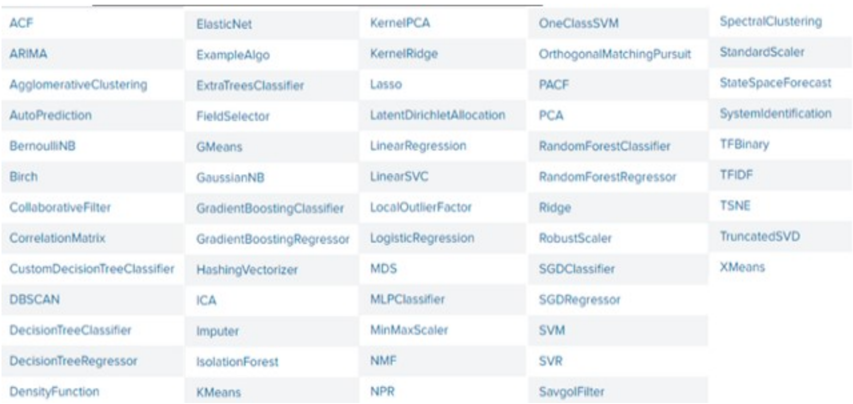


Fig. 1. Histogram of using deviation to detect outliers. (1) – Using Standard Deviation with no sliding window; (2) – Using Standard Deviation with a sliding window; (3) – Using Median Absolute Deviation with a sliding window

analysis of data using statistics, for example, use the deviation and calculate certain deviations from it.

This approach will significantly improve the detection of some malicious activity, but if the data with you are working changes quickly or their consistency is affected by many side parameters – the number of false-positive, after the correlation rules, won't be good enough because you can not specify the required granularity for your data set.

That is why, the next step, when analyzing an array of cybersecurity data is to use machine learning. Splunk, out of the box, provides the ability to operate various machine learning algorithms, namely:



ACF	ElasticNet	KernelPCA	OneClassSVM	SpectralClustering
ARIMA	ExampleAlgo	KernelRidge	OrthogonalMatchingPursuit	StandardScaler
AgglomerativeClustering	ExtraTreesClassifier	Lasso	PACF	StateSpaceForecast
AutoPrediction	FieldSelector	LatentDirichletAllocation	PCA	SystemIdentification
BernoulliNB	GMeans	LinearRegression	RandomForestClassifier	TFBinary
Birch	GaussianNB	LinearSVC	RandomForestRegressor	TFIDF
CollaborativeFilter	GradientBoostingClassifier	LocalOutlierFactor	Ridge	TSNE
CorrelationMatrix	GradientBoostingRegressor	LogisticRegression	RobustScaler	TruncatedSVD
CustomDecisionTreeClassifier	HashingVectorizer	MDS	SGDClassifier	XMeans
DBSCAN	ICA	MLPClassifier	SGDRegressor	
DecisionTreeClassifier	Imputer	MinMaxScaler	SVM	
DecisionTreeRegressor	IsolationForest	NMF	SVR	
DensityFunction	KMeans	NPR	SavgolFilter	

Fig. 2. Splunk ML algorithms

All these algorithms will be useful and functional under different scenarios for creating correlation rules.

One of the most effective algorithms for detecting anomalies is DensityFunction. The implementation of this algorithm allows you to set different parameters on which training may depend and, consequently, the end result. The algorithm also involves estimating

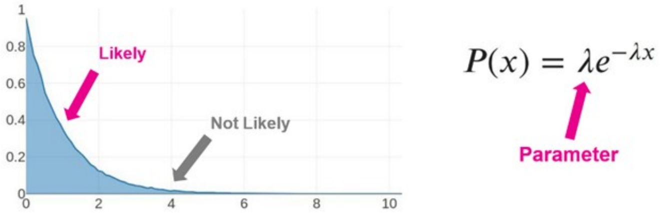


Fig. 3. The parameter that is responsible for the input data on which the training will depend

the different distribution of events within the sample for training.

About λ, this parameter can be anything that helps to classify information or set certain characteristics for sampling. For example, if you want to detect a password matching attack, this setting can be a sign of the day – day and night, or a sign of the week - weekends or weekdays. That is, you look at the data and perceive them differently depending on whether it is now weekend and night (the probability of attack is higher) or now weekdays and day (the probability of attack is normal).

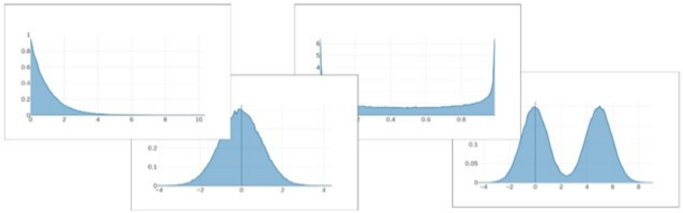


Fig. 4. Types of distribution: Exponential, Normal, Beta, Gaussian Kernel Density Estimation (Gaussian KDE) distribution

To build a correct and clear detection of anomalies, you should not forget to remove the extra noise from your sample for training. What is meant by noise – it is the expected anomalous activity from legitimate objects, the principle of operation of which is like the actions of the attacker. Such objects should be added as an exception, so they do not spoil the statistical sample and do not interfere with the learning of your algorithm.

Also, if you have only a few data points it's likely you're fitting on noise.

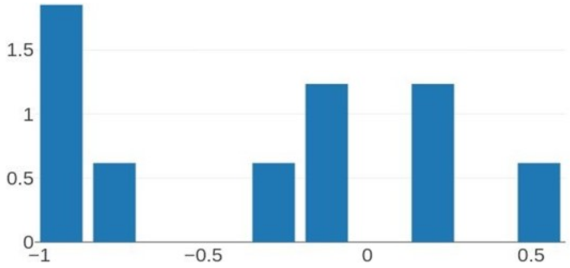
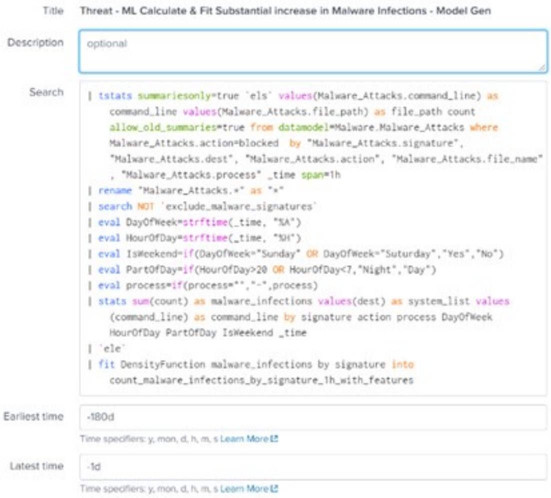


Fig. 5. Caveats with DensityFunction. Poor date set

So, let's write a correlation rule that will detect an abnormal number of end-user infections with one type of malware. To implement we need input data, in our case, it will be data from the EDR system. To train DensityFunction, we will create a sample 180 days ago, not counting today. We will use the trained algorithm every hour to analyze events for the previous hour and thus we will detect anomalies.



Title: Threat - ML Calculate & Fit Substantial increase in Malware Infections - Model Gen

Description: optional

Search:

```
| tstats summariesonly=true `els` values(Malware_Attacks.command_line) as
    command_line values(Malware_Attacks.file_path) as file_path count
    allow_old_summaries=true from datamodel=Malware.Malware_Attacks where
    Malware_Attacks.action=blocked  by "Malware_Attacks.signature",
    "Malware_Attacks.dest", "Malware_Attacks.action", "Malware_Attacks.file_name"
    , "Malware_Attacks.process" _time span=1h
| rename "Malware_Attacks.*" as "*"
| search NOT `exclude_malware_signatures`
| eval DayOfWeek=strftime(_time, "%A")
| eval HourOfDay=strftime(_time, "%H")
| eval IsWeekend=if(DayOfWeek="Sunday" OR DayOfWeek="Suturday","Yes","No")
| eval PartOfDay=if(HourOfDay>20 OR HourOfDay<7,"Night","Day")
| eval process=if(process="","-",process)
| stats sum(count) as malware_infections values(dest) as system_list values
    (command_line) as command_line by signature action process DayOfWeek
    HourOfDay PartOfDay IsWeekend _time
| `ele`
| fit DensityFunction malware_infections by signature into
    count_malware_infections_by_signature_1h_with_features
```

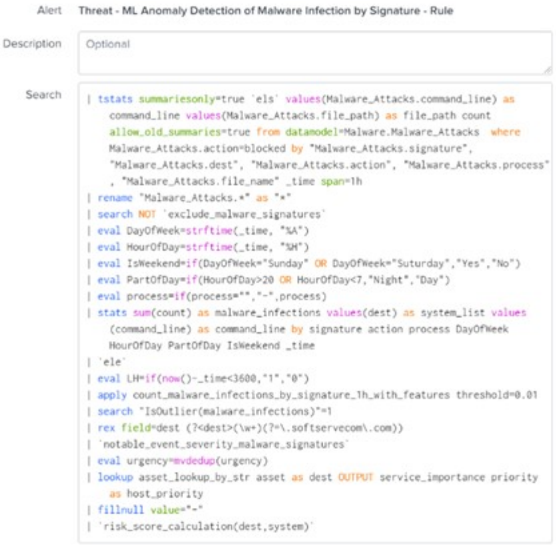
Earliest time: -180d

Time specifiers: y, mon, d, h, m, s Learn More

Latest time: -1d

Time specifiers: y, mon, d, h, m, s Learn More

Fig. 6. DensityFunction training and recording of results in the algorithm “count _malware _infections by _signature _1h _with _features”



Alert: Threat - ML Anomaly Detection of Malware Infection by Signature - Rule

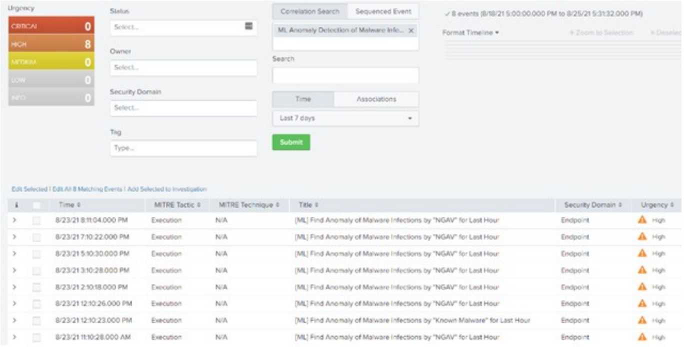
Description: Optional

Search:

```
| tstats summariesonly=true `els` values(Malware_Attacks.command_line) as
    command_line values(Malware_Attacks.file_path) as file_path count
    allow_old_summaries=true from datamodel=Malware.Malware_Attacks  where
    Malware_Attacks.action=blocked by "Malware_Attacks.signature",
    "Malware_Attacks.dest", "Malware_Attacks.action", "Malware_Attacks.process"
    , "Malware_Attacks.file_name" _time span=1h
| rename "Malware_Attacks.*" as "*"
| search NOT `exclude_malware_signatures`
| eval DayOfWeek=strftime(_time, "%A")
| eval HourOfDay=strftime(_time, "%H")
| eval IsWeekend=if(DayOfWeek="Sunday" OR DayOfWeek="Suturday","Yes","No")
| eval PartOfDay=if(HourOfDay>20 OR HourOfDay<7,"Night","Day")
| eval process=if(process="","-",process)
| stats sum(count) as malware_infections values(dest) as system_list values
    (command_line) as command_line by signature action process DayOfWeek
    HourOfDay PartOfDay IsWeekend _time
| `ele`
| eval LH=if(now()-_time<3600,"1","0")
| apply count_malware_infections_by_signature_1h_with_features threshold=0.01
| search "IsOutlier(malware_infections)"=1
| rex field=dest (?<dest>(\w+)(?=\.softservecom\.com))
| `notable_event_severity_malware_signatures`
| eval urgency=mvdedup(urgency)
| lookup asset_lookup_by_str asset as dest OUTPUT service_importance priority
    as host_priority
| fillnull value="-"
| `risk_score_calculation(dest,system)`
```

Fig. 7. Use the “count _malware _infections _by _signature _1h _with _features” algorithm to detect anomalies

The rule is implemented and works in real conditions. The result of work are 8 alerts in the last 7 days, which is an excellent result in terms of load, as the company has more than 20,000 different types of end-users (servers, laptops, PCs) where EDR is installed.



Urgency	
CRITICAL	0
HIGH	8
MEDIUM	0
LOW	0
INFO	0

Status: Select...
Owner: Select...
Security Domain: Select...
Tag: Type...
ML Anomaly Detection of Malware Info... ×
Time: Last 7 days
✓ 8 events (8/18/21 5:00:00.000 PM to 8/25/21 5:31:32.000 PM)

i	□	Time	MITRE Tactic	MITRE Technique	Title	Security Domain	Urgency
›	□	8/23/21 8:11:04.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 7:10:22.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 5:10:30.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 3:10:28.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 2:10:18.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 12:10:26.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 12:10:23.000 PM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "Known Malware" for Last Hour	Endpoint	⚠ High
›	□	8/23/21 11:10:28.000 AM	Execution	N/A	[ML] Find Anomaly of Malware Infections by "NGAV" for Last Hour	Endpoint	⚠ High

Fig. 8. Results

4. Conclusion

The article shows how to use DensityFunction algorithm to detect these anomaly types. Of course, many other anomaly types can be added to detection process, some of them will be discussed in our next articles. Also, we can use other algorithms to detect suspicious activity, in each case you need to do a lot of training with datasets to find the optimal one that works effectively in that moment.

References

1. Haider S., Ozdemir S. Hands-On Machine Learning for Cybersecurity / Soma Haider, Sinan Ozdemir, // Packt Publishing Ltd. – 2018. – 601 p.
2. Electronic Sources: The Splunk Platform. A data platform built for expansive data access, powerful analytics and automation. – Available from: https://www.splunk.com
3. Electronic Sources: Machine learning for cybersecurity. – Available from: https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b
4. Electronic Sources: Machine learning: practical application for cybersecurity. – Available from: https://www.recordedfuture.com/machine-learning-cybersecurity-applications/

Article: received 08.09.2021
revised 20.10.2021
printing adoption 17.11.2021

ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ АНОМАЛІЙ З КІБЕРБЕЗПЕКИ ДЛЯ ЗМЕНШЕННЯ ХИБНОПОЗИТИВНИХ СПРАЦЮВАНЬ У ЩОДЕННІЙ РОБОТІ ЦЕНТРУ З ПРОТИДІЙ КІБЕРЗАГРОЗАМ

Р. Карпюк[1,2], П. Венгерський[2]

[1] *Софтсерв Ко., e-mail:* simmppllee@gmail.com
[2] *Львівський національний університет імені Івана Франка,*
вул. Університеська, 1, Львів, 79000, e-mail: petro.vengersky@gmail.com

За допомогою машинного навчання ми здатні виявляти різні типи загроз з кібербезпеки, наприклад, брут-форс, аномальні зростання або спадання в мережевому трафіку, стежити за зараження кінцевих користувачів шкідливим програмним забезпеченням або виявляти атаки на критичну інфраструктуру, наприклад АД, ДНС. Основна перевага використання МЛ для таких сценаріїв – точність виявлення тих чи інших аномалій. Це суттєво зменшує фінансові затрати на кібербезпеку в організації і швидкість протидії атакуючим.

Ключові слова: машинне навчання (ML), управління подіями з інформаційної безпеки (SIEM), Splunk, функція густини, кореляційний пошук, аналіз, кібербезпека, центр з протидії кіберзагрозам (CSOC).