POLITECNICO DI TORINO

Master's Degree Course in Computer Engineering

Master Degree Thesis

Analysis of security events with Anomaly Detection and Machine Learning techniques

(Consoft Sistemi Spa)

Supervisor:
Prof. Paolo Garza
Dr. Marco Scala (Consoft Sistemi Spa)

Candidate:
Akhil Anand
Student Id: 260278

Academic Year 2020-2021

Acknowledgements

My foremost thanks are offered to Dr. Marco Scala from Consoft Sistemi Spa, whose guidance with this project were absolutely essential to the progress that was made during this thesis. I also extend my gratitude to Prof. Paolo Garza for graciously agreeing to be my second mentor. Splunk Education Platform and Consoft Sistemi Spa provided me with vital support as I explored unfamiliar tools and technologies during this past year. At the Consoft Sistemi, the technical team for giving the support by providing me the system and their internal system working environment configurations which let me learn the application and adaption of the things that I have learned so far into the real world. Politecnico Di Torino, for providing me in depth knowledge over the many topics that I have learned so far from classroom and the laboratories for the practical applications of the knowledge.

I would like to thanks especially all of my friends at Politecnico di torino who provided me with encouragement and support throughout the process. To all my friends and fellow members as the Politecnico Indian Student Association, who have given me the closest things make it possible for me to consider the place as second home through which I came to know many other interesting peoples, I would like to thank you all from bottom of my heart. I foremost like to thank Lucia Pandolfo who make it more possible to consider the place as second home and made me more familiar to all the possible aspects of learning a new way to live and enjoy the life also for the helping me with Italian culture, foods and language and places in the past year her contribution is unforgettable.

Lastly, to my family Mumma, Pappa, brother Akshay, Sister Ankita and brother in law their support and love have not only enabled this thesis, but provided me with the opportunity to study abroad and continuous support during all these years. This thesis is for you and all you have done for me in last twenty-five years.

To anyone else whom I forget to mention, thank you.

Sincerely,
Akhil

Abstract

Due to major shift of personal computing in past decade, a rise in the number of users using different platform in the field of computer applications and network devices have been reported. This leads to higher complexity in examining data, which makes bigger question towards securing it and update the security majors efficiently with time is being the top priority in every field. Keeping the above points in mind this project work was carried in relation with the Consoft Sistemi Spa, in Torino. The project helps to implement security major by detecting anomaly with detection of Outliers in the usages of system to visit distinct destinations by each user. For carrying out the work the web platform Splunk Enterprise is used.

The work started with some learning part of Splunk Infrastructure and its applications. Followed by learning part of SPL (Splunk Processing language) which is the language to communicate with Splunk. The two main applications of Splunk Enterprise that are used over the work are Splunk Machine learning toolkit and Access Anomalies.

The goal of the project was to apply the best possible anomaly detection machine learning algorithm to detect the outlier by using the real time data of Consoft Sistemi that are regularly updated and stored in the data model inside the Splunk platform. The data model contains the information of every user and their use pattern of the system with the details over the time.

The initial step is exploring the data to get the better visualization and understanding the raw data followed by cleaning and transformation which makes data more understandable and helpful for implementing machine learning algorithms. After the clearly understanding the data the different machine learning algorithms is trained over the data of last 30 days. After training the different models, the inspection of the model is carried out with the help of summary command of SPL which helps for the inspection of the build model and it is saved as a report which can be also visualized inside the Splunk dashboard of Access Anomalies. This reports over the dashboard is scheduled to update at an interval of every 24 hours.

The Dashboard reports the different values over the anomalous use pattern of every user. With the help of these values the outlying users are reported in different sections of dashboards using different machine learning algorithms. A comparison has been done in between the different algorithms, by calculating the concordant and discordant results. Also to get the clear information over each user the detailed overview why that particular user is outlying is calculated.

Finally, the dashboard sends an alert to the admin on the scheduled time which can be also set to real time which should be selected on the basis of security level needed (also the scheduled alert can be set hourly/daily/weekly basis), after the alert admin can get the details of each users which are outlying by visiting the dashboard and looking to the detailed view section where the required details of outlying users have been shown, from this section admin can notice which particular user is outlying in visiting how many destinations and at what time he did the failure. Moreover, by selecting the particular user it will redirect the admin to the failure event where it can be found which are the particular destinations, source and host used by the outlying user.

Contents

List of Figures

List of Tables

1 Introduction

In the past decade, the personal computing has undergone a major shift. As the capabilities of personal computer and networking have grown, more complex and compute-intensive applications have been developed to serve companies and their clients using faster and efficient devices. These two areas of growth have progressed at different speed, and, as such applications powered by internet are generating a lot of data which is being more complex to examine and implementation of security being the most important part to carry out work in safe and secured environment inside company.

1.1 State of Art

In response to these concerns, and with demonstrated interest from the "Consoft Sistemi Spa" this project aimed to analyse the security events by detecting the anomalies using machine learning algorithms in the "Consoft Security lab" which is the software platform based on "Splunk Enterprise" of Consoft Sistemi Spa to analyse security events over the network and application inside the association.

The purpose of the project is to analyze security events, access to information and application systems to identify anomalous situations and possible indicators that give evidence of possible ongoing or successful data breach activities. The Anomalous situation in this project work is to detect the outliers in a particular situation where outliers are the users failing to access different destinations, and also if the number of failure is greater than expected by a particular user is considered as outlier. The particular behaviour of a user is calculated on the basis of his past failures to access to multiple destination considering access to destinations in last 30 days.

Outlier detection will deal with making observations with data models. One of the important fact that outlier detection can truncate a variety of application domains anomalies in data which mostly turns to some critical actionable observations. For example, an anomalous user pattern inside the virtual private network of a company could signal that a system has been hacked which is sending out some sensitive data to any unauthorized destination. Some other critical cases of Anomaly detection is sharing credit card transaction over the network which could indicate credit card or identity theft.

The project was be carried out using Splunk platform, which is the market-leading data intelligence platform, which allows to collect events in real time from any heterogeneous source in a non-structured way, to make correlations and analyzes, using pattern matching, anomaly detection and predictive models an gives the access to multiple algorithm of machine learning based on data type and problem to solve.[1]

1.2 Overview

This thesis work includes a first phase of study of the some applications and infrastructure of splunk platform the initial study phase started with "Splunk infrastructure study" where the different resources and their uses inside splunk is covered, followed by "Splunk fundamentals" which covers the the parts of Splunk Enterprise use with different methods how the splunk enterprise can get the data from different sources, "Splunk Machine Learning toolkit" which demonstrate the machine learning uses inside the splunk platform and also the use of "Splunk Processing language(SPL)" which is the core to use the splunk enterprise or in simple communicating language of splunk with the users.

Some hands on practice using the above applications with the data provided as csv file inside Splunk enterprise and also from data coming from Consoft Sistemi internal systems. Subsequently, the analysis methodologies of some security "use cases" was analyzed in order to define trends and behavior models and consequently report anomalous events.

Finally, the methodologies developed is applied to recognize access anomalies to sensitive data that fall within the scope of the GDPR.

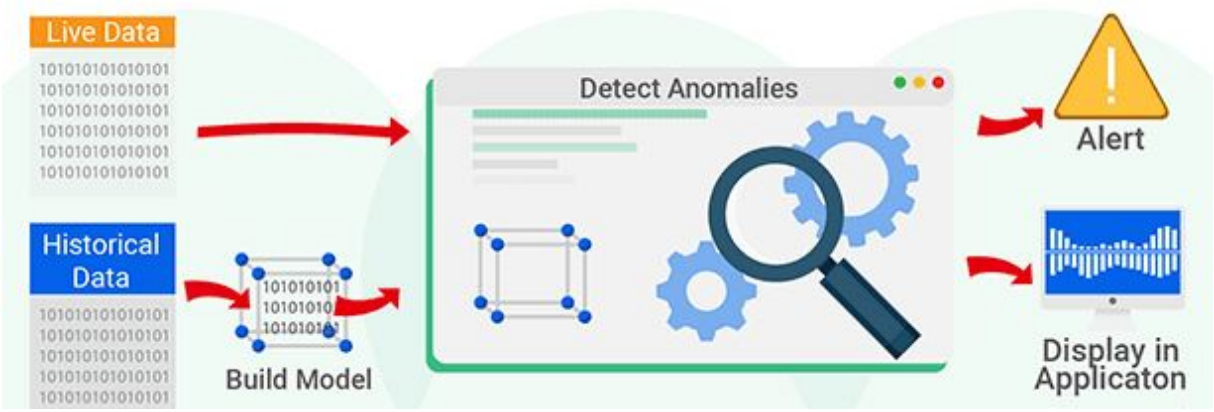


Figure 1: Experiment approach Overview

In the next chapters, I will discuss the different Applications and Data sets/models used to carry out project work in details. Furthermore the solutions proposed for different data sets followed by some small experiments related to understand the use pattern of the platform. Later on the main goal of detecting the anomalies will be explained with the information and details of the application, dataset and machine learning algorithms used to find out anomalies. A comparison between the existing method and the proposed method of detecting anomalies using the Machine learning algorithms will be discussed. In conclusion, the benefits of using the proposed method over the current existing method of detecting anomalies will be covered.

2 Related Work/ description of used Software

In this chapter introduction to the splunk platform and the various applications inside it is covered starting from Splunk Enterprise, Splunk Processing Language (SPL) followed by the basics of Splunk Machine Learning toolkit(Splunk MLTK) application along with some Machine learning algorithms and illustrative examples of MLTK useful applications. Moreover, the effect of Machine Learning toolkit over different data sets and services inside the company are analysed, and the factors regarding the possible services using machine learning are reviewed. Choosing the correct algorithm and metrics for both machine learning toolkit and dataset is also vital, so these matters are covered in this chapter.

2.1 Introduction to Consoft Security Lab

Consoft security Lab is the Consoft Sistemi Spa software interface which implements Splunk Enterprise inside it. Moreover, Splunk enterprise is the backbone of the Splunk Security Lab which combines all the other applications required for analyzing the data, searching particular in the data to better understand and visualizing events from the data entering from different sources of Consoft Sistemi infrastructure. The data for the splunk enterprise can be taken from multiple sources as the Splunk enterprise is installed in distributed manner inside the Consoft Sistemi Spa infrastructure, sources can be web applications, system application, sensors on different devices, network devices and others. Once the data sources is defined it is being indexed by Splunk enterprise and interpret data into events for each specific cases which makes easier to search and view the details.

Splunk Enterprise platform can be accessed using web browsers to create and manage knowledge objects inside the data, create reports, and so on.Splunk enterprise can be administered by the use of command-line interface.

Splunk Enterprise infrastructure depends on the need of organisation. It can satisfy the needs of organisation by giving options to use multiple applications. An application[2] have various combinations of configurations that can be installed on splunk enterprise for specific use through Splunk application base[3]. Each application has its own knowledge object to create some views over data and create the dashboards inside the Splunk infrastructure. Single Splunk infrastructure group multiple applications inside it. We can also develop own application through to access to Splunk developer site[4].

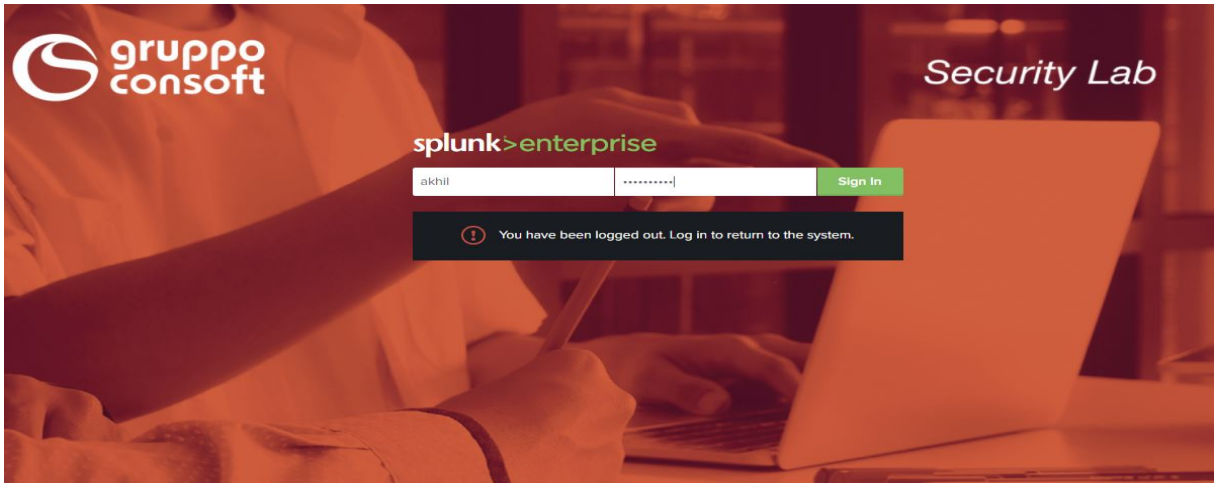


Figure 2: Splunk Security lab

Once we are logged in to the Consoft Security lab [1] we will land into Splunk Enterprise, where we can choose the desired application for the analysis. Below (fig.3) a small snippet of Splunk enterprise showing the overview of app Access Anomalies (where the final analysis of the project is done).

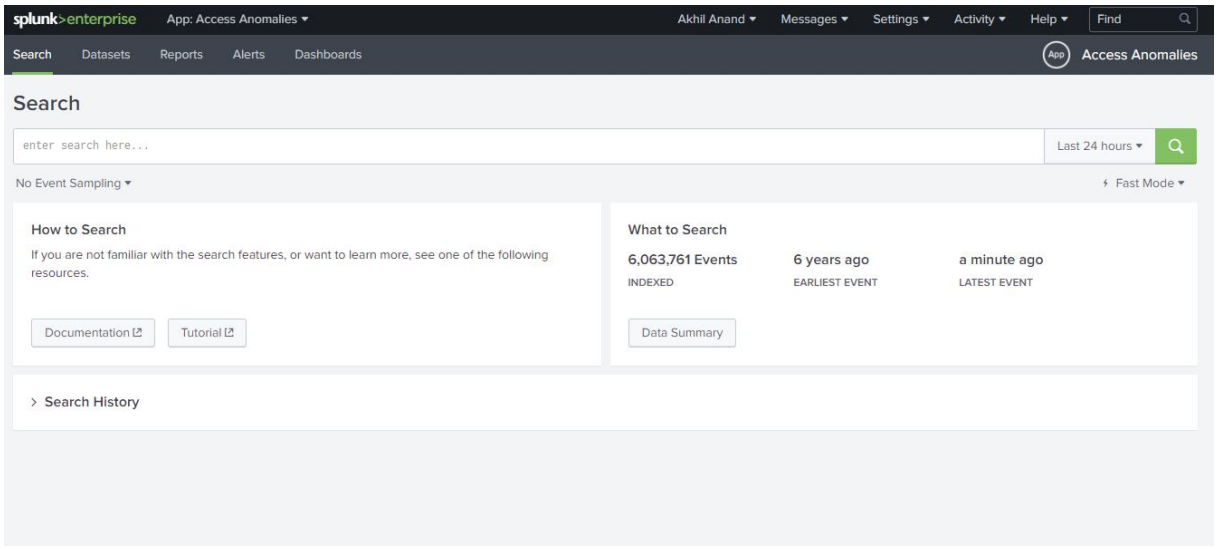


Figure 3: Snippet of Splunk Enterprise

2.2 Structure of Splunk Enterprise

Before showing the structure I would like to discuss the features of Splunk Enterprise which will be helpful for understanding the structure and its use. The main features of Splunk are:

- **Indexing**
  Splunk collects data through operating systems, databases, servers, devices, websites and more. After data collection, index segments compresses and stores the data to accelerate searching by maintaining metadata.

- **Search**
  The data inside is Splunk Enterprise is navigated through the search. Every search can be saved as report and it can be used to provide the details for dashboard panels. A data insight can be provided by the search by:

  - Retrieving events
  - Metrics calculation
  - Search certain conditions in particular time frame
  - Data pattern Identification
  - Predicting trends of data

[1] login credentials provided by admin which should be used under OpenVPN/internal network of Consoft Sistemi

- **Alerts**
  When results generated by search (for real-time and historical searches) matches the configured conditions then it is notified through the alerts. Alerts can also be configured for triggering actions by sending information of the alert to particular email, reporting RSS feed with the alert information, and initializing custom script, like the one that report the alert to system log.

- **Dashboards**
  It displays results of different real-time searches which are running in the background. It have multiple modules within the panels, the modules can be charts, reports and so on. Each panel of dashboard is mostly associated with pivots and some searches which are already saved.

- **Pivot**
  Pivot helps to visualize the data in charts, table or visualization created through Pivot Editor[5]. The Editor helps to map attributes of the data without the use of Search Processing Language (SPL)[6]. It can be added to the dashboard and also can be generated as a report.

- **Reports**
  The Searches can be saved as reports and can be added as the dashboard panel. Reports can be scheduled to run on some defined intervals and it can be used for generating the alerts if it meets some particular defined conditions.

- **Data Model**
  It encodes the domain knowledge of single or multiple indexed data. Data Model gives the users of Pivot Editor access to create the dashboards and reports without writing the search to generate it.

2.2.1 Processing Tier and Data Pipeline

Splunk Enterprise instance performs a specialized task and resides on one of three processing tiers corresponding to the main processing functions:

- Data input tier
- Indexer tier
- Search management tier

These specialized instances are known as “component”. The table below explains the functions that each components performs.

Tier	Component	Description
Data Input	Forwarder	Forwarder gets data from multiple sources and forwards it to indexer. Forwarder works with minimum resources, and it resides mildly on machines that generates data.
Indexing	Indexer	Indexer takes the data through multiple forwarders and indexes it. It transforms data to events and then stores event as an index. Indexer searches the data that have been indexed to respond the search requests generated by search head. Multiple indexers are deployed in indexer clusters, for ensuring the high data and protect from data loss. The indexers reside on dedicated machines
Search Management	Search Head	Search head is the direct point of interaction, it directs the requested search to a group of indexers, and produces the results to us. For assure high availability of the search head and to simplify horizontal scaling, a set of search heads deployed in search head clusters. The search head resides on a dedicated machine.

Table 1: Splunk Enterprise components and processing tiers

We can add components to each tier as necessary to support greater demands on that tier. For example, for a large number of users, we can add extra search heads to serve better the users.

The diagram[4] demonstrate an example that how the different processing components of splunk resides on multiple processing tiers. Starting from bottom, diagram describes all three processing tiers, where data is entering the system through forwarders, after performing some preprocessing it forward the data to indexers for Indexing. Then indexers perform indexing of the data by reorganising it as an event and save events as Index. Finally, a search head execute the functions for search management. Search head takes the search requests of users and then it distribute that request to a group of indexers, here indexers perform search on the data that are stored locally. Search head then combines all the indexer results and produce it to the user.

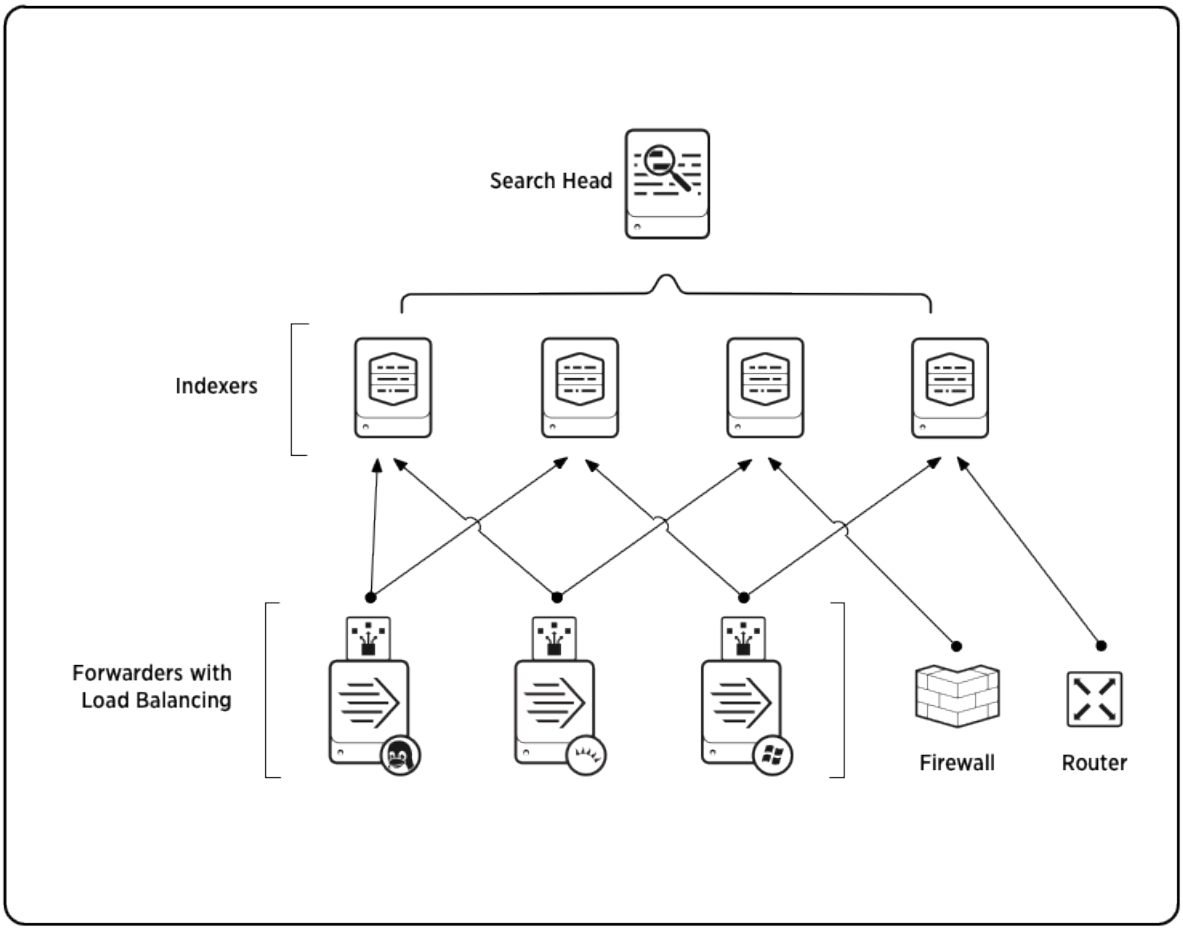


Figure 4: Example of Splunk Deployment

The Splunk processing tiers proportionate to data pipeline, it is basically the path followed by the data inside Splunk. When the data goes through this pipeline the components transform it at each step and makes it available for search.

Different segments of Data pipeline are:

- Input segment
- Parsing segment
- Indexing segment
- Search segment

The co-relation of three processing tiers with these four segments is follow:

- Input segment takes care of data input.
- Parsing segment and Indexing segment handles indexing tier.
- Search segment deals with Search management tier through search head.

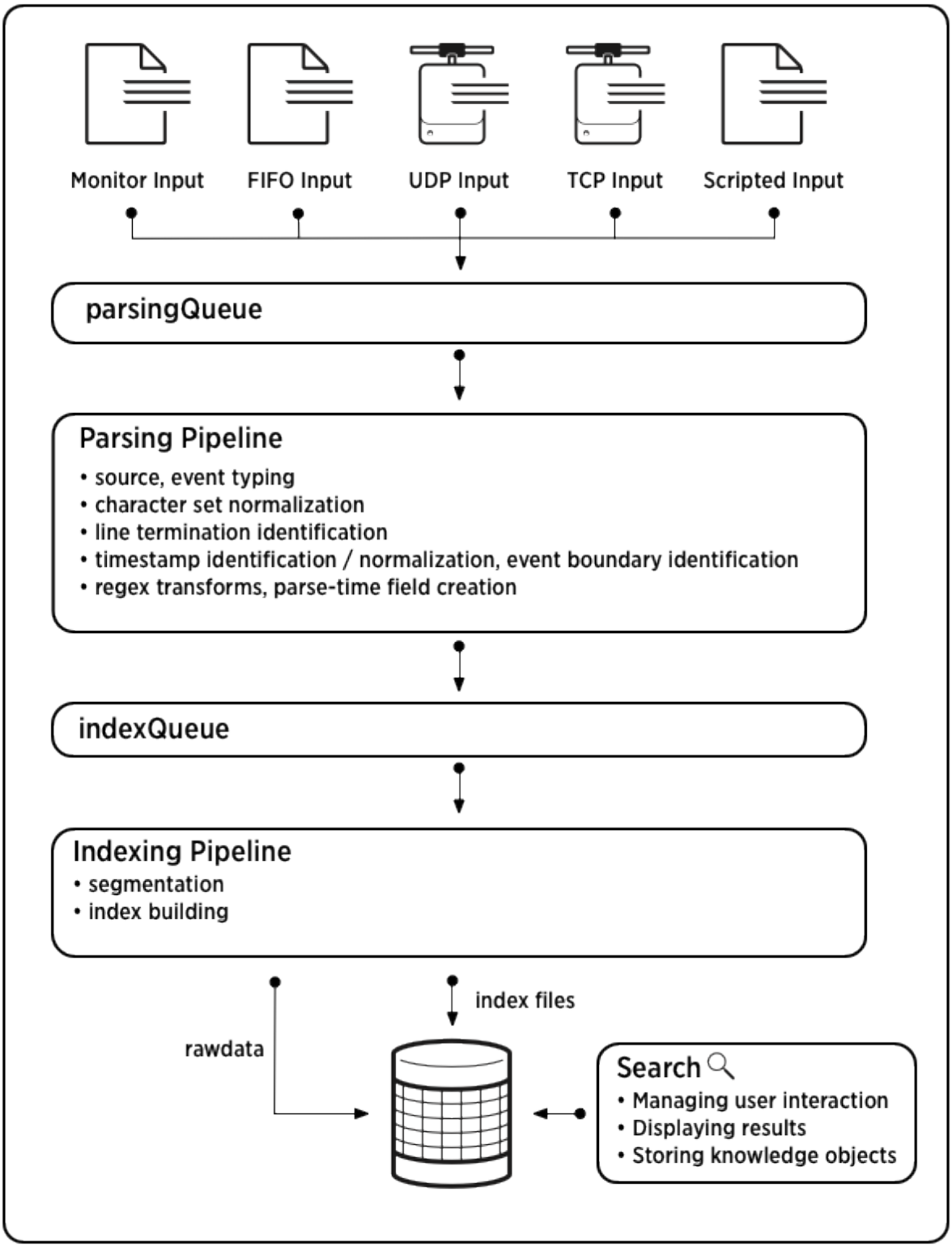


Figure 5: Splunk Data Pipeline

2.2.2 Insight into data pipeline segments functionalities:

- **Input**
  Splunk uses input Segment to get the data in the system. The data are normally raw data and it is distributed in 64k blocks with annotation of each block with metadata keys. These keys have the information of the data source, the generating host and the type of the data. Sometime some of the keys also include extra internal information about character encoding, or index value for storing the events. Splunk does not examine the contents of each data stream, it just apply the key to complete source despite of event type. At this stage splunk does not have impression of each events, it just have information about data stream which have some global properties.

- **Parsing**
  At this stage, Splunk analyzes the data stream, examine the data and transform it to individual events. It can be also called event processing as it process the events through data stream. Parsing segment is divided in sub-phases as follow:

  - Data stream is divided to individual lines
  - Parsing, identifying the data type, and setting timestamps to each events.
  - Interpreting individual events through metadata that are copied from source key.
  - Apply regex rule to transform metadata and data as events.

- **Indexing**
  During indexing segment Splunk writes the parsed events to the disk with index. It writes indexing file as well as compressed raw data to the disk.

  Generally, indexing and parsing called together as indexing process. But to examine the data more precisely , the two segments are considered individually.

- **Search**
  Search manages multiple features like how the data is accessed, which kind of view is applied to the data, and howindexed data should be used. As part of the search function, Splunk software stores knowledge objects created by user,like event types, reports of particular search, creation of dashboards, generation of alerts, and extraction of field. It also manages its process itself.

2.3 Search Processing Language(SPL)

SPL is the short form of Search Processing Language. This language is used to communicate to the Splunk Software, it gives additional functionality with more than 140 search commands. It is based on SQL and UNIX pipeline with optimization done for time-series data. Its scope merges multiple functionality like data search, commands to insert, modify or delete, also some functionalities of filtration, enrichment and manipulation can be achieved. It includes command for machine learning such as Anomaly detection which will be used in this project work.
Search commands tell Splunk software what to do to the retrieved events from the indexes. We can use command to extract information by filtering some unwanted information, some statistical calculation can be also performed and the charts can be created through the

search command itself.
Search commands are also associated with functions and their arguments which can be used to specify the action of commands over the result. We can use some clauses for grouping the results like clauses on the number of days or for a particular department.

2.4 An Introduction to Machine Learning Toolkit (MLTK)

Machine Learning Toolkit (MLTK) is an application inside the Splunk Environment that helps to apply various machine-learning algorithms and techniques like Outlier Detection, Classification, Anomaly detection, Regression to the data.

Machine learning is the understanding of computer algorithms which automatically improves with experience and with the proper use of data that are already present to us. We create a model using Machine learning algorithm based given sample data, which is called as "training Data", which can be used to for multiples tasks like identify the data pattern, future prediction of data trend, detecting some anomaly in real time data and many more.

MLTK opens the gateway to create the models of Machine learning, and put it into the operation. It include new SPL[6] commands to create and work with machine learning models.

2.4.1 Features of MLTK:

- It has a dashboard with showcase of different sample data sets for exploring the concepts of Machine Learning algorithms.
- Assistants for managing the data source, for selecting proper algorithm on the basis of data selected algorithm, and any tuning some extra parameters that are required to configure a particular algorithm. To fit and apply the model every assistant gives access to multiple machine learning algorithms.
- It provides more than 30 machine learning algorithms and it gives the access over 300 open-source algorithms build using Python.
- SPL command extensions for using machine learning techniques on data, like fitting the model and applying it to the data. Some extra commands for summary and deletion of the model is also provided.
- Reusable information graphics to analyze the data and visualise it in some particular format.

2.4.2 Types of Machine Learning Algorithms

- **Regression**
  Regression is used for the prediction of a numeric event from multiple contributing factors. It is an prediction analysis which helps to understand the dependency of target variable with the change of corresponding independent variable keeping other variables fixed. It predicts the real values which are numeric such as age, amount, expense, etc.

  It is supervised learning which helps to find a correlation between some variables of dataset and helps us to predict some continuous output based on single or multiple predictor variables.

- **Classification**
  Classification algorithms is used for predicting a class or a category for the particular data by consider multiple contributing factors. As it gives prediction to unlabelled data it is considered as Predictive algorithm.

  It is an Supervised Learning which is used for identification of category to some new observations or some un-categorised data through training of categorised data. Here a model learns from categorised observations and then classifies new observation to a set of classes. Like pink or yellow, 1 or 0, wine or not wine, tiger or cat, etc.

- **Forecasting**
  It is also an predictive analytic which predicts the value which are moving in time. Forecasting learns from past values of single variable, such as expense per hour or RAM utilization per min, and gives prediction of their future expected trends.

- **Clustering**
  Clustering algorithm is used for grouping same data points together. It is unsupervised learning and is used as a data analysis to discover interesting patterns inside the data sets, such as grouping of components of light on the basis of their uses.

  This technique is applied when the class is not there for the prediction and the instances have different behaviors.

- **Anomaly Detection**
  This algorithm is used to find outliers(different from all other) in data set by knowing the expected behaviour of the particular data that differs from behaviour of all other of sme kind. The behavior can be learned by using its previous use pattern also, and comparing it to current reality.

  It is also called outlier detection, as anomaly detection is the mode of identifying data as anomalous. It can be also the the mode of detecting and identifying anomalous data in any data-based event or observation that differs majorly from the rest of the data.

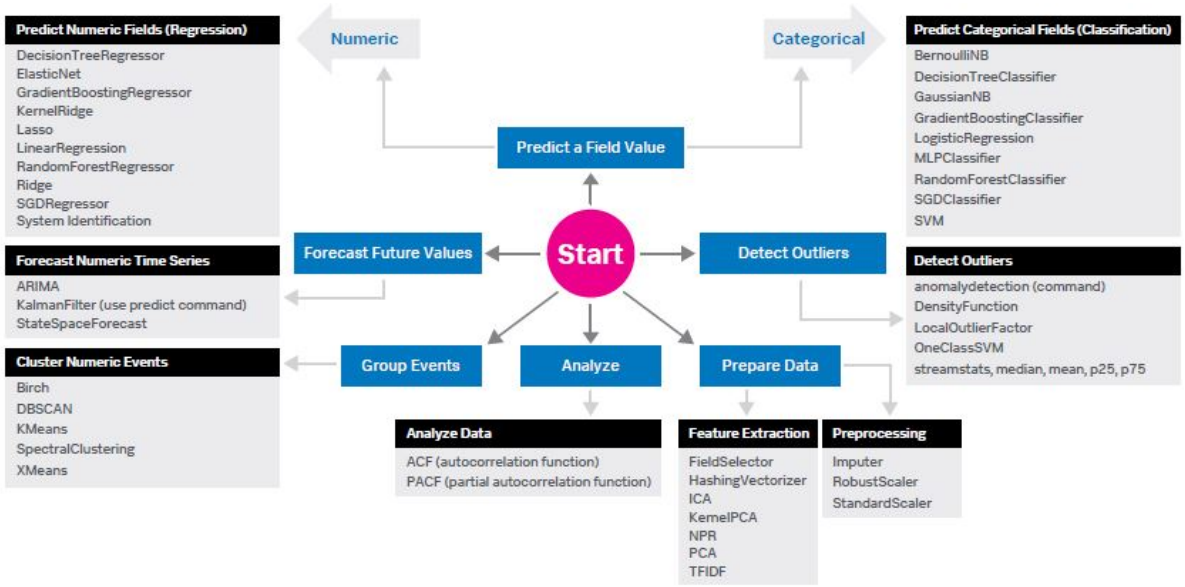


Figure 6: Machine Learning algorithms categorization

2.5 Machine Learning Process

A series of steps is followed by the machine learning process, starting from collection of data from various sources, cleaning the data by removing or transforming the missing data, explore the data with the better visualisation, building the model using any of the appropriate algorithm for training Data, and evaluate model through the Testing data, finishing with deployment of machine learning model.

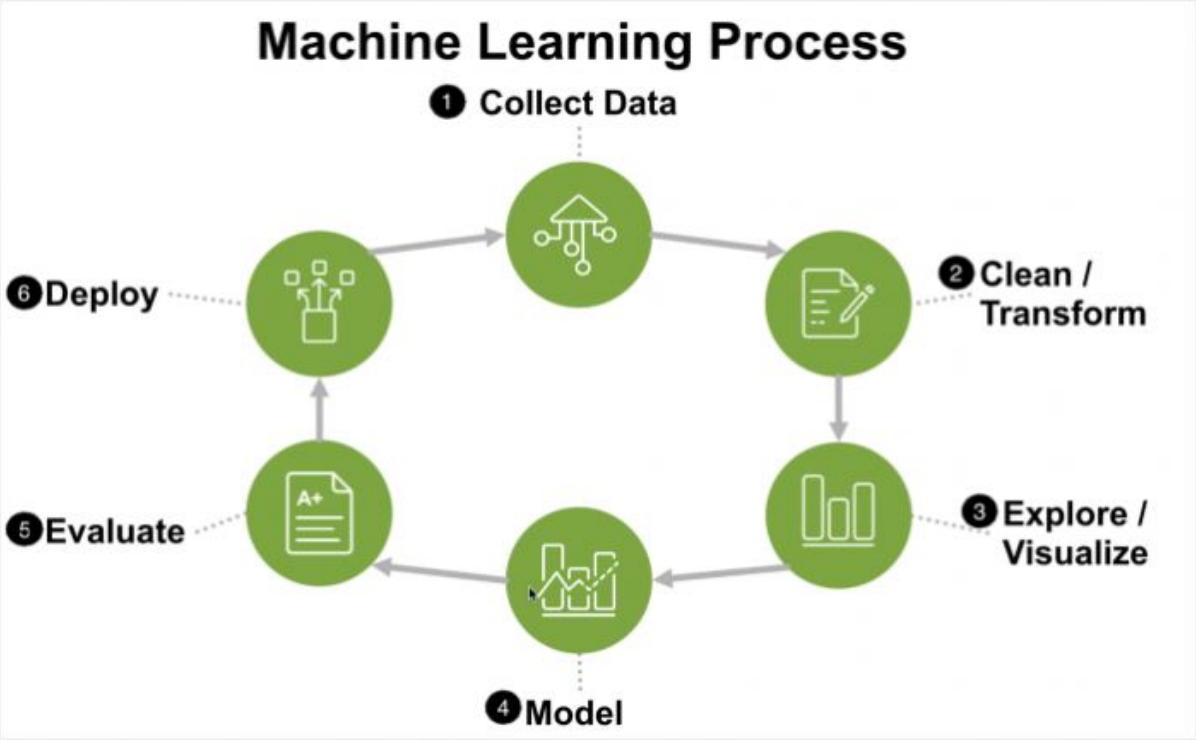


Figure 7: Machine Learning Process Theoretically

1. Data collection from various sources.
2. Cleaning and transforming the data. Machine learning algorithms learns the data which are in matrix form and there should be no missing values. So this step cleans and transforms the data.
3. Explore/visualize data and we have to be sure that it is encoded as expected.
4. Use these training data for building the model.
5. model is evaluated by test data.
6. Deploy model for new observations.

In practice machine learning does not follow the linear path:

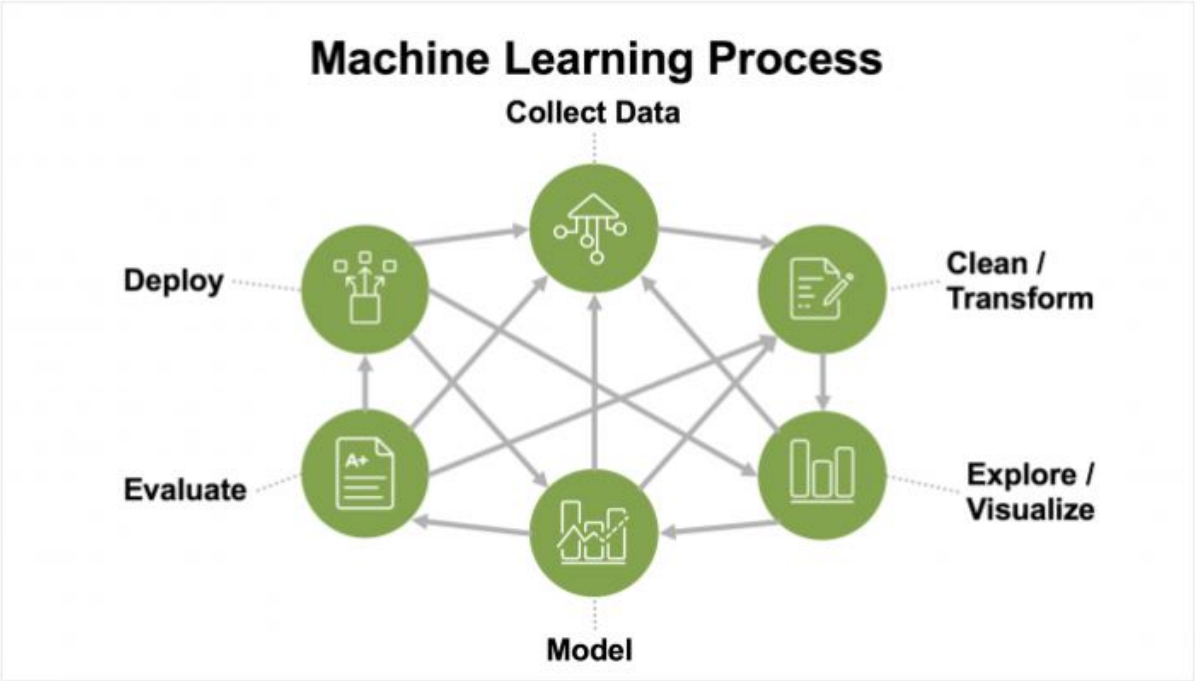


Figure 8: Machine Learning Process Practically

During the evaluation we can discover that the performance of the model is not generating the results as per our expectation, and we need to clean the data furthermore. It is possible as there might be possibility of some missing data,some disagreement with unit of data, it might not be properly weighted, etc. We need to clean the data and train the model until it provide expected results. .

2.6 Machine Learning Process within MLTK

The MLTK application operates as an extension for Splunk environment and gives user access to complete the machine learning process. The MLTK helps to creates some custom machine learning algorithms in particular cases. The MLTK enables the workflow using a suite of guided modelling Assistants. The MLTK can also be used outside of the guided framework with a series of machine learning specific Search Processing Language (SPL) commands and over 300 algorithms.

2.6.1 Explore Data

After ingesting the data it is explored to ensure it is suitable for and ready to be used in a machine learning process. The ingested data into the MLTK is easily visualized in both tables and graphics. The Splunk platform and the MLTK offers several methods through which we can clean and transform our data and address common data issues including the identification and removal of errors, addressing missing values, and potentially converting categorical values into numeric values.

Ingested data typically goes through three stages in order to be ready for machine learning.

- Stage 1

Data is ingested into the Splunk platform during the first stage. The data is typically semi-structured. We can see some commonality between the events, such as URLs and https calls in the example below:

Figure 9: Machine Learning Raw Data

- Stage 2

After ingesting the data we perform field extraction, where the data is partially organized within a table. This table sometimes include problems, such as missing data, non-numerical data, values in a widely-ranging scale, and words representing values. Data in the following example is not yet suitable for machine learning because the scales for the request_size and response_time are very different, which can negatively affect the chosen algorithm:

JSESSIONID	method_modified	request_size	response_size	response_time
95ca35e8-8ecc-11e8-9b3b-784f4371b6d6	GIFTCARD	0.09		2429370
95ca35e8-8ecc-11e8-9b3b-784f4371b6d6	GIFTCARD	0.084	511	2826820
95c5b8c5-8ecc-11e8-9d94-784f4371b6d6	GIFTCARD	0.09		2467710
952ad821-8ecc-11e8-8568-784f4371b6d6	CREDITCARD	0.082		2411460
952ad821-8ecc-11e8-8568-784f4371b6d6	CREDITCARD	0.115	559	3167550
952ad821-8ecc-11e8-8568-784f4371b6d6	CREDITCARD	0.103	500	2634080
952ad821-8ecc-11e8-8568-784f4371b6d6		0.081	522	3252010
9529c8e1-8ecc-11e8-a493-784f4371b6d6	GIFTCARD	0.098	501	3172080
9529c8e1-8ecc-11e8-a493-784f4371b6d6		0.114	520	3012260
95286c40-8ecc-11e8-a8bf-784f4371b6d6	GIFTCARD	0.118	514	3219490
95286c40-8ecc-11e8-a8bf-784f4371b6d6	GIFTCARD	0.094	581	2578870
95286c40-8ecc-11e8-a8bf-784f4371b6d6	GIFTCARD	0.088	494	3437350

Figure 10: Machine Learning messy table Data

- Stage 3

Following the further field analysis and data cleaning, the data is in a clean matrix and is amenable to machine learning algorithms. This data has no missing values, is strictly numeric, and the values are scaled correctly. This data is ready for machine learning:

method_numeric	request_bytes	response_bytes	response_time	SS_request_bytes	SS_response_bytes	SS_response_time
1	117	484	28.633	1.52829611445	-0.273100937052	-5.6116822758
0	118	560	24.977	1.61486301221	1.04517158866	-5.68840180219
0	107	581	25.55	0.662627136916	1.40943110235	-5.67637765322
0	88	599	25.136	-0.982143920403	1.72165354265	-5.68506525824
1	94	521	26.524	-0.462742533881	0.368689634678	-5.65593869844
0	89	599	26.11	-0.89557702265	1.72165354265	-5.66462630345
0	84	582	29.068	-1.32841151142	1.42677679348	-5.60255399517
1	90	599	242.937	-0.809010124896	1.72165354265	-1.11460859337
0	84	511	282.682	-1.32841151142	0.195232723399	-0.280577527647
1	90	559	246.771	-0.809010124896	1.02782589754	-1.03415381652

Figure 11: Machine Learning clean table Data

2.6.2 Experiment

Data experimentation is the process of analyzing training data and creating a machine learning model by using the desired algorithm. The MLTK have several machine learning commands and some built-in algorithms through which we can perform data experimentation. The MLTK also have the guided machine learning workflows through a series of *Smart Assistants* and *Experiment Assistants*.

-Smart Assistants Overview

Smart Assistants enable advanced query building and machine learning outcomes. It is built on the backbone of the Experiment Management Framework (EMF), Smart Assistants offer a segmented, guided workflow with an updated user interface. Smart Assistants let us quickly move from fitting a model on historic data to applying a model on real-time data and taking action.

There are four Smart Assistants available on Splunk MLTK:

- Forecasting Assistant
- Outlier Detection Assistant
- Clustering Assistant
- Prediction Assistant

The workflow inside Smart Assistants moves through different stages staring from defining, learning, reviewing and operationalize to load data, build model, and putting the model to the production. After each stage we have an access to data preview and visulization panel. The Smart Assistant provides an updated look and feel as well as the option to bring in data from different sources to build the model.

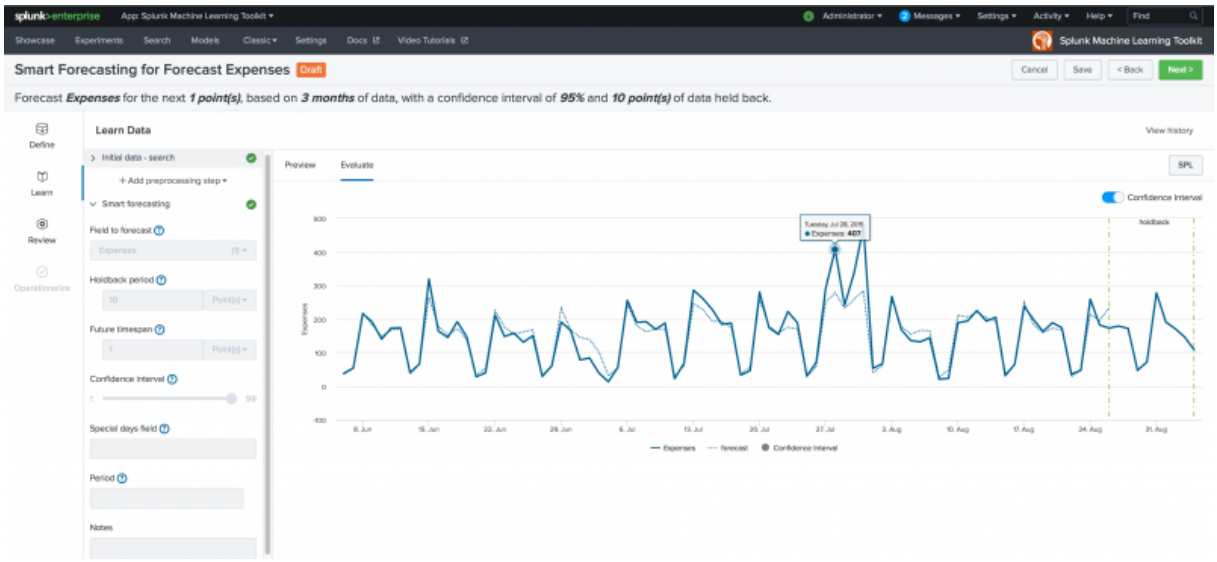


Figure 12: Smart Assistant Overview

- Smart Forecasting Assistant

  The Smart Forecasting Assistant uses the StateSpaceForecast[7] algorithm to forecast future numeric time-series data. StateSpaceForecast formed on Kalman filters, and automatically imputes any missing values in the data. To help improve the accuracy of forecast, this algorithm includes the ability to account for the effects of specific days that need to be treated differently. Version 4.4.0 and above of the Smart Forecasting Assistant offers both univariate and multivariate forecasting options.

- Smart Outlier Detection Assistant

  The Smart Outlier Detection Assistant uses the DensityFunction[8] algorithm to leverage a density algorithm and segment data in advance of anomaly search. DensityFunction gives a efficient workflow for creation and storing the density function which can be utilized for detection of anomaly. DensityFunction algorithm groups the data with the use of by clause, and a unique density function is stored for each of these group. It supports the multiple probability density functions such as Gaussian KDE, Normal, Beta distribution and exponential function. Anomaly detection accuracy of DensityFunction is based on size and quality of training dataset.

- Smart Clustering Assistant

  The Smart Clustering Assistant uses the K-means[9] algorithm to partition events into groups. K-means is computationally faster than most other clustering algorithms. K-means Clustering algorithm divides the data points in K different groups. It uses the implementation of K-means from scikit-learn. K-means algorithm is used as it works better with the unlabeled data and when we have knowledge of the groups in which data will be divided.

- Smart Prediction Assistant

  Smart Prediction Assistant uses the AutoPrediction[10] algorithm to determine numeric or categorical data type. AutoPrediction uses the RandomForestClassifier[11]

algorithm for the prediction of numeric and category of the data.The RandomForestClassifier uses RandomForestClassifier estimator from the scikit-learn for fitting the model and predicting the categorical fields value.

AutoPrediction executes the train-test split during the fitting of the model. It uses the train-test split function of sklearn to do the data split.

- Experiment Assistants Overview

Experiment Management Framework (EMF) provides aspects for a machine learning pipeline which can be monitored through one user interface with by the use of built-in automated model setting.

Experiments manage the algorithm, data source, and the parameters used to configure algorithms inside one framework. An Experiment is an object in Splunk enterprise which saves all the track of its history and settings, also details about the scheduled training and alerts.

There are six Experiment Assistants available:

- Predict Numeric Fields
- Predict Categorical Fields
- Detect Numeric Outliers
- Detect Categorical Outliers
- Forecast Time Series
- Cluster Numeric Events

Experiment workflow begins with the creation of a new machine learning pipeline, based on the selected MLTK guided modeling interface or Assistant. Once we select and apply Experiment parameters to our data and generate results, the workflow continues through the available visualizations and statistical analysis. Through the guided Experiment Assistant we make selections including specifying data sources, selection of an algorithm and algorithm parameters, selection of the fields for the algorithms to analyze, setting training/test data splits.

Every step of an Experiment provides an option to open a clone of the SPL in a new search window for further customization. Once we save an Experiment, a new exclusive knowledge object is created in Splunk that keeps record of affiliated alerts and scheduled training as well as setting of the pipeline.

We can choose an Experiment Assistant based on the type of machine learning we wish to perform on our data.

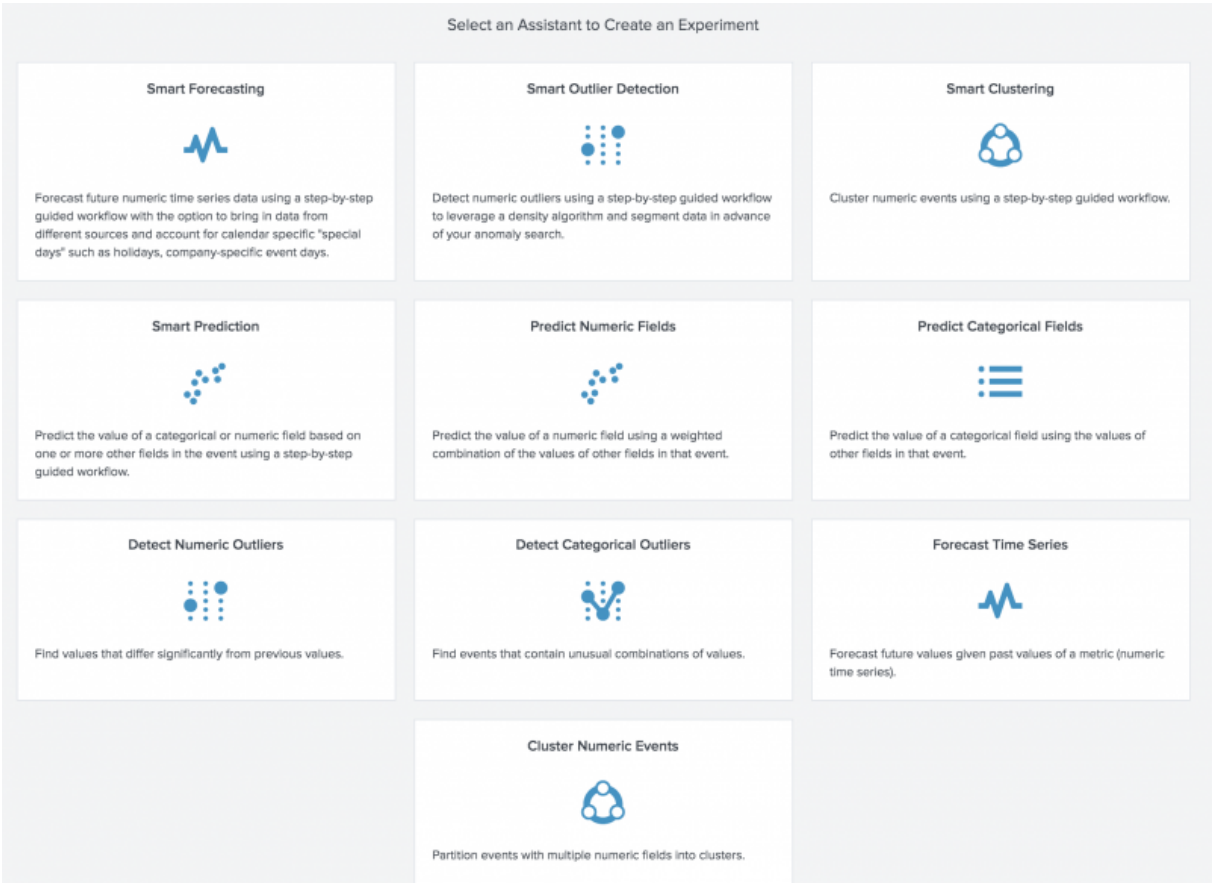


Figure 13: Assistant Overview

- Predict Numeric Fields

  The Predict Numeric Fields Experiment Assistant uses regression algorithms to predict or estimate numeric values. Such models are used for determining to what extent certain peripheral factors contribute to a particular metric result. After the regression model is computed, we use these peripheral values to make a prediction on the metric result. We can use any of the following available algorithms inside Predict Numeric Fields Experiment Assistant:

  - Linear Regression
  - RandomForestRegressor
  - Lasso
  - KernelRidge
  - ElasticNet
  - Ridge
  - DecisionTreeRegressor

  The following visualization illustrates a scatter plot of the actual versus predicted results. This visualization is taken from the experiment that will be discussed in the next chapter about Server Power Consumption.

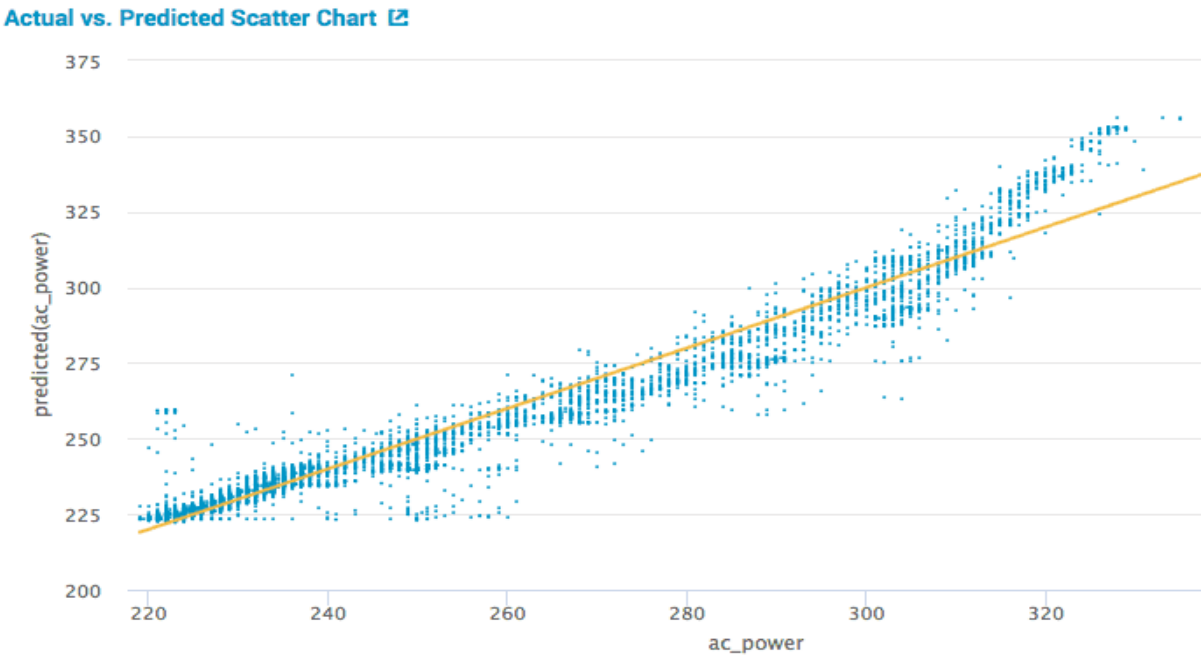


Figure 14: Predict numeric field illustration

- Predict Categorical Fields

  The Predict Categorical Fields Experiment Assistant uses the classification Algorithm to categorise the data. A classification algorithm learns the tendency for data to belong to one category or another based on related data. We have choice of following algorithms inside Predict Categorical Fields assistant:

  - LogisticRegression
  - (Support Vector Machine)SVM
  - RandomForestClassifier
  - GaussianNBGaussianNB
  - BernoulliNB
  - DecisionTreeClassifier

  The following classification table shows the actual state of the field versus predicted state of the field:

Prediction Results

DiskFailure	predicted(DiskFailure)
Yes	Yes
No	No
No	No
No	No
Yes	Yes
No	Yes
No	No
No	No
Yes	Yes
No	No

Figure 15: Categorical Prediction result

- Detect Numeric Outliers

  The Detect Numeric Outliers Experiment Assistant determines values that appear to be extraordinarily higher or lower than the rest of the data. The identified outliers are indicative of interesting, unusual, and possibly dangerous events. This Assistant is restricted to one numeric data field. The Detect Numeric Outliers Assistant is compatible with the following distribution statistics:

  – Standard deviation
  – Median absolute deviation
  – Interquartile range

  In the following visualization, the yellow dots indicate outliers:

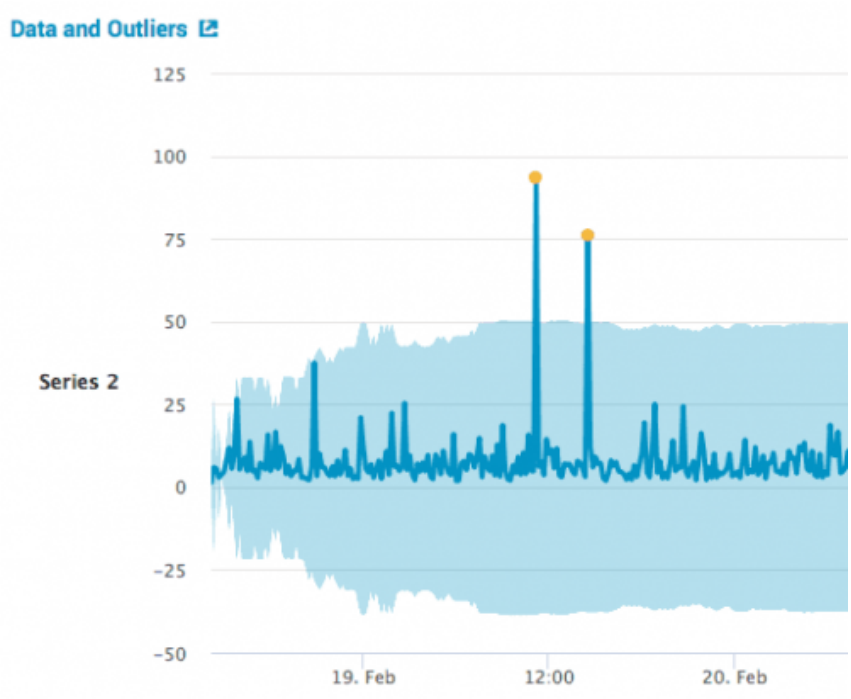


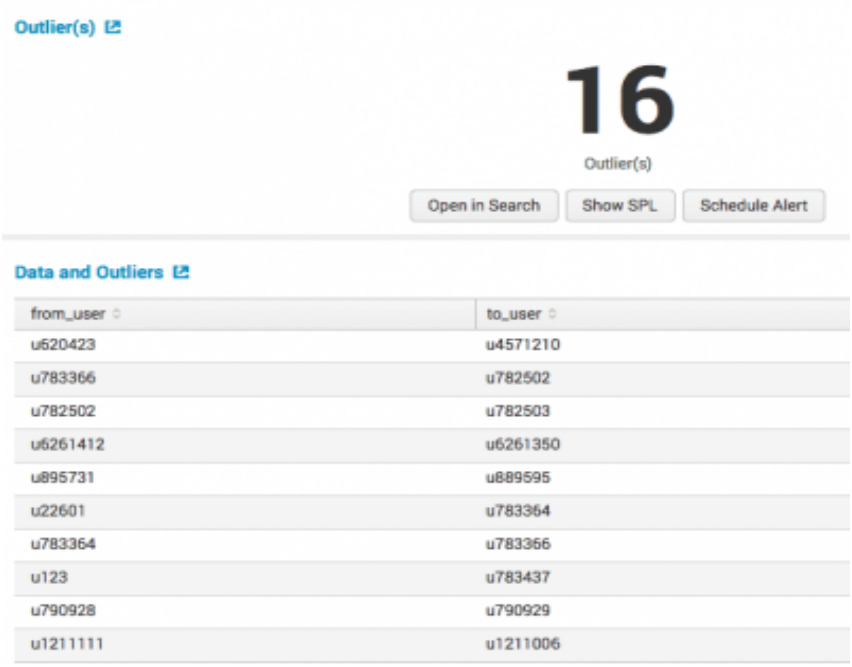
Figure 16: Numeric Outlier Visualization

- Detect Categorical Outliers

  The Detect Categorical Outliers Experiment Assistant helps to identify the data that indicate interesting or unusual events. This Assistant allows non-numeric and multi-dimensional data, such as string identifiers and IP addresses. To detect categorical outliers, we need to input data and select the fields from which to look for unusual combinations or a coincidence of rare values. When multiple fields have rare values, the result is an outlier.

  The Detect Categorical Outliers Assistant uses the Probabilistic measures to find out the Outlier.

The following image illustrates results of one of the exercise done using detect Categorical outliers:



from_user	to_user
u620423	u4571210
u783366	u782502
u782502	u782503
u6261412	u6261350
u895731	u889595
u22601	u783364
u783364	u783366
u123	u783437
u790928	u790929
u1211111	u1211006

Figure 17: Categorical Outliers detection

- Forecast Time Series

  The Forecast Time Series Experiment Assistant forecasts the next values in a sequence for a single time series. Forecasting makes use of past time series data trends to make a prediction about likely future values. The result includes both the forecasted value and a measure of the uncertainty of that forecast.

  The Forecast Time Series Experiment Assistant gives choice of the following algorithms:

  – State-space method using the Kalman filter
  – Autoregressive Integrated Moving Average (ARIMA)

  The following visualization shows a forecast of sales numbers:

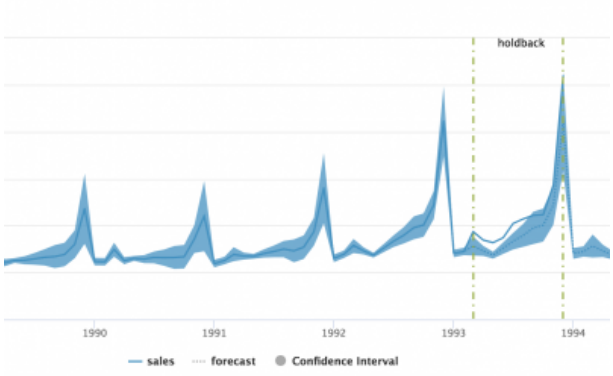


Figure 18: Forecast time series using Kalman filter

- Cluster Numeric Events

  The Cluster Numeric Events Experiment Assistant partitions events into groups of events based on the values of those fields. As we don't have information over groupings, so this is considered as unsupervised learning.

  The Cluster Numeric Events Experiment Assistant gives the choice of following algorithms:

  - K-means
  - DBSCAN
  - Birch
  - Spectral Clustering

  The following visualization illustrates a clustering of humidity data results:

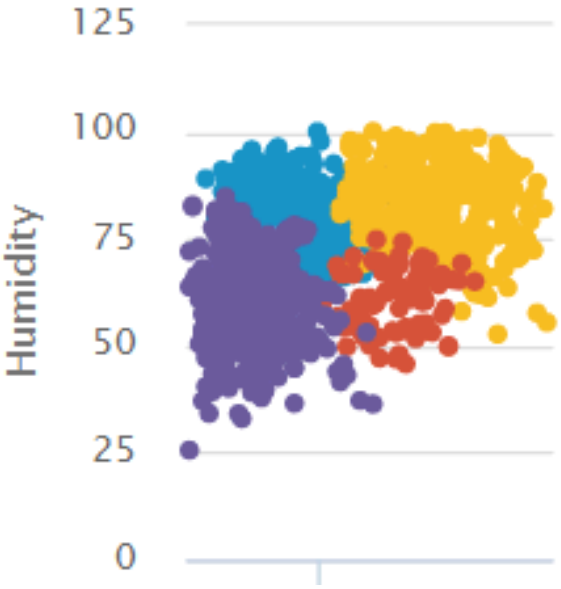


Figure 19: Mapping of clustering result

2.6.3 Evaluate Results

After doing the experiments with our data the most vital role is done in the process of evaluation, where we can find out how useful our machine learning model is with the data. Such as is the problem is adequate for the data, do we have enough data to build such model, or if the data is sufficiently cleaned before learning. Also we need to evaluate if our experiment outcomes give you the results as expected or not. The MLTK guided modeling Assistants all include data visualizations through which we can quickly assess experiment results. We can also choose from a range of scoring metrics to measure your machine learning results.

Some of the custom visualization of evaluation results can be following:

- 3D scatter plot
- Boxplot Chart

- Distribution Plot
- Downsampled line chart
- Forecast Chart
- Heatmap Plot
- Histogram Chart
- Outliers Chart
- Scatter line chart
- Scatterplot Matrix

2.6.4 Tune and Iterate

As part of evaluating experiment results, we can tune and iterate the machine learning model. Adjusting model settings ensures us to get the desired machine learning results prior to applying the model to unseen data and putting the model into operation. The MLTK guided Assistants make it simple to adjust model settings and gauge model performance improvement.

We need to repeat the steps of experimenting, evaluating, and tuning until we are ready to put our trained model into production.

2.6.5 Deploy Model

At this stage the trained machine learning model is ready for deployment and application on new, never-before-seen data. As a best practice,we regularly check our model outcomes, as well as the sources of the new data and make adjustment to our machine learning model settings as needed.

In the MLTK guided modelling Assistants, we can schedule model retraining, get alerted about model, and publish models for making it visible to other users and also for application of model.

In the next chapters, we will apply and use these knowledge over the different data sets and data model. Then we will evaluate how much is the solution is feasible as compared to other possible solutions.

3 Problem Reported and solution proposed

3.1 Overview

In this chapter we are going to discuss experiments carried out using all the knowledge discussed so far over Splunk environment and moreover the implementation of machine learning model's in Access Anomalies application of Splunk using all the three available algorithms for Anomaly Detection inside Splunk environment. The main goal of the experiment is to find out the outlying users, where the users who fails multiple times to connect to different destinations are considered as outliers.

Moreover, for the clear visualization a dashboard for home page of Access Anomalies Application "Outliers in number of Destination" is created where the users that fails to connect to different destinations (Outliers) are evaluated, these outliers are calculated using different machine learning Model generated by using three different machine learning algorithm Density Function, Local Outlier Factor and One Class SVM, also the outlier have been calculated using the simple mathematics to find out the benefits of using machine learning algorithm over the simple mathematics which was previously used by the Consoft Sistemi. A comparison of results between the three machine learning algorithms have been shown clearly in the dashboard. Also a final comparison between concordant and discordant result using simple mathematics and using the best suited Model generated by the DensityFunction algorithm have been shown.

3.2 Data Exploration

3.2.1 Statical overview and correlation of data

In this section we are going to analyse in depth about the data used for carrying out this experiment, the data model that is used is called as "Authentication data model". It is the real time data model inside Consoft Security lab which can be globally accessed from all the application inside the Splunk Enterprise infrastructure. Authentication data model contains multiple fields that contains the information about all the users and their details including the information of source and destination accessed by each user with the time and duration on the basis of unit, category, domain and priority. In total there are 31 fields inside the authentication data model including numeric as well as string types.

The source of the data for the authentication Data Model is Windows Security Event Logs that are generated by multiple hosts. Every event is categorised inside one of the Event ID which is called as Windows Security Log Event ID which have some standard predefined values for every specific action Eg. Event ID 4624, means an account was successfully logged on.The detailed list of Windows security Event ID have been reported by Monterey Technology group[12].

Moreover the principal components that are considered mainly are "user" which contains the username of each user, "action" which reports the user action as success or failure, "dest" which concerns the destinations that user/s are connecting and the time at which the particular destination is visited by individual user.

The other components in the data model reports the details about the source domain, source category, source business unit, source user priority, destination domain, destination category, destination business unit, destination user priority, duration and response time.

It have been found that on an average on a active work day the Authentication data model reports more than 4000 events every day. The data generation on hourly basis of

one active work day have been reported in the figure 20.

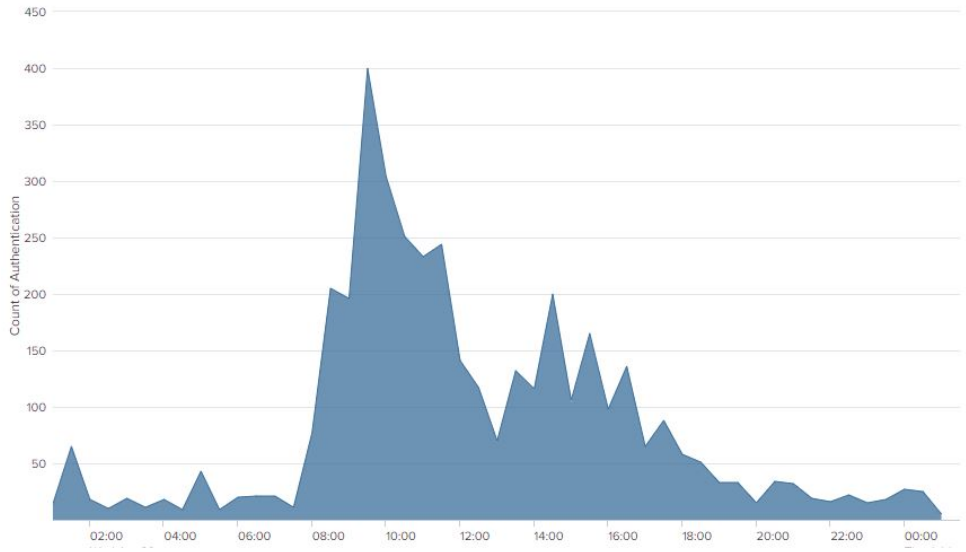


Figure 20: Hourly data generation

It can be clearly see events inside the authentication happens in the morning time wn in figure 20 that the maximum number ofhich is normally the job start time for the users and the following peaks have been seen after the lunch hours which clearly makes sense that the users trying to reconnect to the system after the lunch.

For better understanding, a column chart have been created in descending order of the number of times the different destinations have been accessed by users.

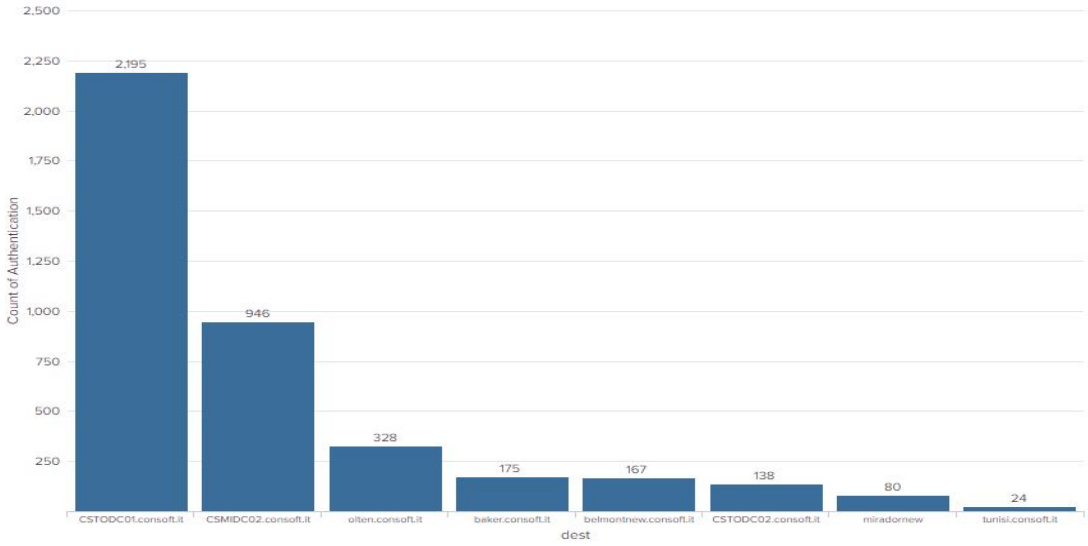


Figure 21: Number of access for each destination

From the figure 21 it can be seen that the destinations are mainly the system that are inside the domain consoft.it, also these are the main destinations that should be protected in any security events so we will only consider the destinations that are having the domain as consoft.it.

One of the field that should be considered is the action field inside the Authentication data model. As discussed earlier in this chapter the action mainly have 2 classes “Success”

and “failure”. In this field every event have been categorised compulsory as one of the action that can be either success or failure. The proportion to success to failure have been found as nearly about 1:4 for a particular active work day. This can be better understood by the figure 22.

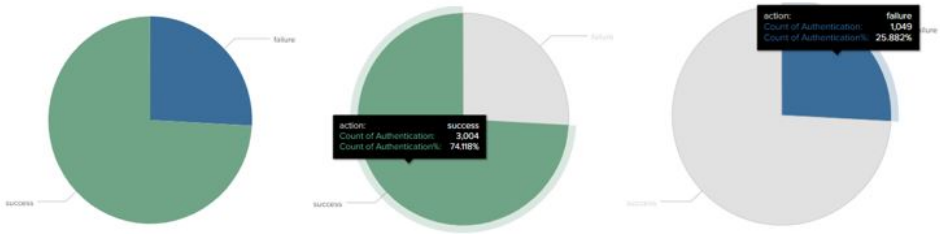


Figure 22: Percentage of action distribution

From the pie chart it can be understandable that on an active work day the 25percent of events are the cases of failure as we look to the number it can be seen that out of 4053 of total number of events of a particular day, 3004 are the success that is approx 75percent and 1049 events are the failure which is nearly 25percent. As, the ratio is quite high for failures if we look from security point of view and also it can be increased of decreased on the basis of user accesses. So, we will consider the cases that are categorised as failure.

3.2.2 Data filtering and cleaning

In this section we will report all the filtering and cleaning processes carried out so far over the Authentication data model that have been discussed in the previous section of data exploration. The main language to apply filtering to the data is by the use of SPL (Splunk Processing Language).

The first step is to filter the users and select only the users that are real users basically elimination of all the events that are saved by individual machines not by users. Which can be done by adding following lines to the SPL during data preparation process:

```
Authentication.user!=”-” AND Authentication.user!=”unknown” AND Authentication.user!=*$
```

Where all the events that are not generated by the users was eliminated and moreover they are not crucial for detecting the outliers as there will be no security events that can be anomalous from the events generated by machine in this particular case. After that the failure events are considered by filtering the Authentication data on the basis on action component where every event is categorised either as success or as a failure.

```
Authentication.action=fail*
```

Adding the above condition to SPL will filter the Authentication data model on the basis of its action, so all the events where the action is failure are selected. Lastly, the destination visited by each user counted distinctly and grouped by with a time span of 1day. Moreover, only the destination where the destination address is ending with consoft.it are considered as in we wanted to report the access anomaly inside the organisation which came with the exploration of the data in previous section with the figure 21.

```
tstats dc(Authentication.dest) as Dest_count count as Failures from datamodel=Authentication

where Authentication.action=fail* AND

Authentication.user!="-" AND Authentication.user!="unknown" AND Authentication.user!=*$ AND

Authentication.dest=*.consoft.it

groupby _time span=1d, Authentication.user

rename "Authentication.user" as user

eval "atf_hour_of_day"=strftime(_time, "%H"),
"atf_day_of_week"=strftime(_time, "%w-%A"),
"atf_day_of_month"=strftime(_time, "%e"),
"atf_month" = strftime(_time, "%m-%B")

eventstats dc("atf_hour_of_day"),
dc("atf_day_of_week"),
dc("atf_day_of_month"),
dc("atf_month")

eval "atf_hour_of_day"=if('dc(atf_hour_of_day)'<2, null(),'atf_hour_of_day'),
"atf_day_of_week"=if('dc(atf_day_of_week)'<2, null(),'atf_day_of_week'),
"atf_day_of_month"=if('dc(atf_day_of_month)'<2, null(),'atf_day_of_month')

fields  "dc(atf_hour_of_day)",
"dc(atf_day_of_week)",
"dc(atf_day_of_month)",
"dc(atf_month)"

eval "_atf_hour_of_day_copy"=atf_hour_of_day,
"_atf_day_of_week_copy"=atf_day_of_week,
"_atf_day_of_month_copy"=atf_day_of_month,
"_atf_month_copy"=atf_month

fields  "atf_hour_of_day","atf_day_of_week","atf_day_of_month","atf_month"

rename "_atf_hour_of_day_copy" as "atf_hour_of_day",
"_atf_day_of_week_copy" as "atf_day_of_week",
"_atf_day_of_month_copy" as "atf_day_of_month",
"_atf_month_copy" as "atf_month"

lookup mask_users_new.csv user OUTPUT newuser as user
```

The above SPL are the combination of all the filtering conditions over the data, that

have been discussed so far. Applying these filtering condition and counting all the failures that has been done by each user over the total number of distinct destinations, counting distinct destination that have been accessed by each user had been calculated using SPL keyword "dc" means distinct count. Also 4 extra field have been added where atf define the "actual time of failure", the actual time to failure have been calculated as hour of the day which tells at what time the event have been failed during the particular day, similarly its been calculated for the day of the week and the day of the month and also the month of failure for a broader and clear view of long term security. Moreover, a lookup file have been created for the data privacy purpose of the Consoft Sistemi to mask the actual username for the view of this thesis report. This lookup table is used for masking the user name of each user. These conditions have been applied over the data of last 90 days. The snapshot below makes it more clearly understandable.

_time	user	Dest_count	Failures	atf_day_of_month	atf_day_of_week	atf_month
2021-04-04	user85	1	176	4	0-Sunday	04-April
2021-04-05	user85	1	174	5	1-Monday	04-April
2021-04-06	user6	3	150	6	2-Tuesday	04-April
2021-04-06	user7	4	205	6	2-Tuesday	04-April
2021-04-06	user33	3	27	6	2-Tuesday	04-April
2021-04-06	user72	3	108	6	2-Tuesday	04-April
2021-04-06	user73	3	150	6	2-Tuesday	04-April
2021-04-06	user74	3	150	6	2-Tuesday	04-April
2021-04-06	user82	1	1	6	2-Tuesday	04-April
2021-04-06	user85	1	177	6	2-Tuesday	04-April
2021-04-06	user86	3	60	6	2-Tuesday	04-April

Figure 23: Filtered data table

In total we get around 93k events out of which 1091 are the events of failure in the last 90 days. And it have also been found that many users are failing to access to more than one destinations inside the enterprise and the number of failures are quite high for some users even with a single destination. Moreover, the data have been grouped for each user on the basis of each day so the field hour of the day was ignored by the splunk.

At this stage the data set is more clear to be fitted with the machine learning algorithms to learn the use pattern and failure pattern of each user on the basis of their accesses.

3.3 Experimental analysis

This part will deal with achieving the main goal of finding Anomalous situation for each user who have been outlying where the outlying users are those whose failing behaviors differs from the one of his previous failure attempts to the system to connect to one or more destinations. To achieve this goal we will use the following three machine learning algorithms given inside the Splunk enterprise to be used for Anomaly Detection:

- Density Function Algorithm
- Local Outlier Factor Algorithm
- One Class SVM Algorithm

3.3.1 DensityFunction Algorithm

The DensityFunction algorithm gives smooth and consistent workflow for creating and storing the density functions for each group of data and use it for detecting the anomaly.The grouping is done with the use of *by* clause, and a unique density function is stored for each of these group.

It supports the multiple probability density functions:

1. Gaussian KDE (Kernel Density Estimation)
2. Normal Distribution
3. Beta Distribution
4. Exponential Distribution

Anomaly detection accuracy of DensityFunction is based on size and quality of training dataset, moreover accuracy of the fitting distribution modeling the basic data generation process and the value selected for threshold parameter should be taken into account.

The model will perform more accurately by satisfying the following conditions:

- There should be at least 50 data points. If the more training data is not possible create less groups by using *by* clause, so that we will have more data points for each group.
- the **threshold** value should always be chosen rather then using the default value. As for each experiment it varies on the basis of domain knowledge. Threshold parameter should be tuned multiple time to get the best suited results.
- inspecton of model can be done by **summary** command.
- the model should be trained more frequently if the data distribution changes over time.

Parameters to be tuned to build model:

- **dist**: it is used to select the density function and based on that the distribution will be carried out, distribution values which are valid in parameter include *gaussian_kde* (Gaussian KDE distribution),*expon* (exponential distribution), *norm* (normal distribution), *auto* (automatic selection) and *beta* (beta distribution).

  When we set this parameter to *auto* it runs all the 4 density function and selects the best one out of them. So, this is set to auto in our experiement.

- **metric**: it is used to calculate the distance from density function and training data to sampled data. Its values include *kolmogorov_smirov* and *wasserstein*. We leave this value to default as it will be wasserstein which is default.

- **threshold**: it works as center for the outlier detection process. It shows the percentage of area that is under each of the density function. The value can lie between 0.000000001 (refers to 0%) and 1 (refers to 100%). This parameter guides algorithm through the fitted distribution to mark the outlier. The different values used to train the model are 0.01(also default), 0.05, 0.1,0.2,0.3.

- **show_density**: by default is False. If we set it to True, each of the data points density will be in output with the field *ProbabilityDensity*. The parameter *show_density* set to true in fitting the model.

- **fit**: command is having the main use which is used to build the model using the density function algorithm. It saves the parameters and details about the build model in the model file. It outputs the outlier by creating a new field *IsOutlier*

- **IsOutlier**: it shows a list of labels, where number 1 tell its an outliers, and 0 tell its an inliers.

- **BoundaryRanges** represents the boundary values of outliers which is calculated on the basis of density function which can be set in accordance with threshold parameter. It follows multi_value field convention where every new line is composed of one boundary region.
  first boundary region
  second boundary region
  nth Boundary region If there are only one boundary region the percentage of boundary region is the threshold value. This field can be empty in two cases:

  – sharp peak in density function because of low standard deviation

  – few data points

  So, the data points which are exactly at boundary closing or opening point are considered as inliers. Closing and opening point is based on the density function which was used.

  – Gaussian KDE: More than one boundary regions are possible, it depends on the number of dips and peaks inside the density function. Outliers will be the data points which will be lying inside these boundary regions.

  – Normal distribution: Two boundary regions one is left other is right. Data points which are on the left side of the left region closing point and the data point on the right of the right closing point are considered as outliers.

  – Beta Distribution: It has only one boundary region. The data points which are placed on the left of left closing boundary region are outliers.

  – Exponential Distribution: It also have one boundary region. The data points which lies down on the right of the closing point of right boundary region are outliers.

  The table below in figure 24 reports all the different threshold used and their boundary regions on the basis of the Density function used in particular case, as we have used the distribution type as auto so it was different density function distribution for each case.

BoundaryRanges_th=0.01	BoundaryRanges_th=0.05	BoundaryRanges_th=0.1	BoundaryRanges_th=0.2	BoundaryRanges_th=0.3
337243896.5135:361177242.5946:0.01	8647275.7568:9784312.3243:0.0501	221721.3784:257170.2703:0.1001	5681.5225:6757.9459:0.2006	142.039:172.2162:0.3005
-Infinity:150.0:0.005 150.0:Infinity:0.005	-Infinity:150.0:0.025 150.0:Infinity:0.025	-Infinity:150.0:0.05 150.0:Infinity:0.05	-Infinity:150.0:0.1 150.0:Infinity:0.1	-Infinity:150.0:0.15 150.0:Infinity:0.15
-Infinity:1740422941:0 265924081796.7838:272289953094.5134:0.0097 355046279965:Infinity:0	-Infinity:8531485:0 1237671667.4865:1393697924.7838:0.0495 1740422941:Infinity:0	-Infinity:41821:0 5633611.7027:7163280.8919:0.0998 8531485:Infinity:0	-Infinity:205:0 23116.7117:37863.5225:0.2 41821:Infinity:0	-Infinity:1:0 89.8288:194.994:0.3 205:Infinity:0
-Infinity:27.0:0.005 27.0:Infinity:0.005	-Infinity:27.0:0.025 27.0:Infinity:0.025	-Infinity:27.0:0.05 27.0:Infinity:0.05	-Infinity:27.0:0.1 27.0:Infinity:0.1	-Infinity:27.0:0.15 27.0:Infinity:0.15
-Infinity:108.0:0.005 108.0:Infinity:0.005	-Infinity:108.0:0.025 108.0:Infinity:0.025	-Infinity:108.0:0.05 108.0:Infinity:0.05	-Infinity:108.0:0.1 108.0:Infinity:0.1	-Infinity:108.0:0.15 108.0:Infinity:0.15

Figure 24: Boundary range distribution

The above figure shows BoundaryRanges on Authentication Data model. The values are represented as x:y:z where x is the left boundary range, y is the right boundary range and z is the percentage of total area of Density function. As, in figure 24 the row 2 with threshold 0.3 represent the boundary range as "-Infinity:150.0:0.15 150.0:Infinity:0.15" which means the left boundary start with -infinity and goes up to 150 and it covers 0.15 i.e 15 percent of total area under density function. Also the area of covered by right boundary range is the same i.e. 15 percent and it goes from 150 to infinity. The area covered by the boundary range is based on the value of threshold so if threshold is 0.3 the sum of area covered by boundary ranges will be always 30 per cent for all the cases.

Syntax:

```
fit DensityFunction field by "field1,field2,....field5" into model name
dist=str
show_density=true or false
sample=true or false
full_sample=true or false
threshold=float
metric=str
random_state=int
```

The above SPL command syntax will be used to fit the model and saved as the model name inside the splunk environment.

```
apply model name
threshold=float
show_density=true or false
sample=true or false
full_sample=true or false
```

After, fitteing the above apply command of SPL is used to apply the saved model to the new data, and it gives option to update some of the parameters value during the apply phase.
The model learned can be inspected using the summary command.

```
summay model name
```

The summary command help to get in detail view of the fitted machine learning model with some extra parameters that gives an in-depth understanding.

Model creation using Density function algorithm:
With the understanding of Density function algorithm, the model density_mode_outlier have been fitted inside the Splunk machine learning toolkit(MLTK version 5.2).

```
| tstats dc(Authentication.dest) as Dest_count count as Failures from datamodel=Authentication where Authentication.action=fail* AND Authentication.user!="-" AND
   Authentication.user!="unknown" AND Authentication.user!=*$ AND Authentication.dest=*.consoft.it groupby _time span=1d, Authentication.user
| rename "Authentication.user" as user
| eval "atf_hour_of_day"=strftime(_time, "%H"), "atf_day_of_week"=strftime(_time, "%w-%A"), "atf_day_of_month"=strftime(_time, "%e"), "atf_month" = strftime
   (_time, "%m-%B")
| eventstats dc("atf_hour_of_day"),dc("atf_day_of_week"),dc("atf_day_of_month"),dc("atf_month")
| eval "atf_hour_of_day"=if('dc(atf_hour_of_day)'<2, null(), 'atf_hour_of_day'),"atf_day_of_week"=if('dc(atf_day_of_week)'<2, null(), 'atf_day_of_week'
   ),"atf_day_of_month"=if('dc(atf_day_of_month)'<2, null(), 'atf_day_of_month'),"atf_month"=if('dc(atf_month)'<2, null(), 'atf_month')
| fields - "dc(atf_hour_of_day)","dc(atf_day_of_week)","dc(atf_day_of_month)","dc(atf_month)"
| eval "_atf_hour_of_day_copy"=atf_hour_of_day,"_atf_day_of_week_copy"=atf_day_of_week,"_atf_day_of_month_copy"=atf_day_of_month,"_atf_month_copy"=atf_month
| fields - "atf_hour_of_day","atf_day_of_week","atf_day_of_month","atf_month"
| rename "_atf_hour_of_day_copy" as "atf_hour_of_day","_atf_day_of_week_copy" as "atf_day_of_week","_atf_day_of_month_copy" as "atf_day_of_month"
   ,"_atf_month_copy" as "atf_month"
| fit DensityFunction "Failures" by "user,atf_month" dist=auto threshold="0.3" show_density=true  into "density_Model_outlier"
```

Figure 25: Density Model fitting inside Splunk MLTK

After fitting the above model inside Splunk MLTK over the last 90 days of data, the model have been given the permission to be available globally to all the other applications to use the trained model "density_Model_outlier

Moreover, the density_model_outlier is applied inside the Access anomaly Application of Splunk to find out the outlying users with details over the data of last 30days. There were around 30k events to be evaluated out of which 46 events have been found as outliers using the density function algorithm model. A view of output can be seen in figure 26.

✓ **29,335 events** (05/06/2021 00:00:00.000 to 05/07/2021 15:00:33.000) No Event Sampling ▾ Job ▾ Fast Mode ▾

Events Patterns **Statistics (46)** Visualization

50 Per Page ▾ Format Preview ▾

_time	user	Failures	Dest_count	atf_day_of_week	atf_day_of_month	atf_month
2021-06-07	user85	175	1	1-Monday	7	06-June
2021-06-07	user126	1	1	1-Monday	7	06-June
2021-06-07	user165	4	1	1-Monday	7	06-June
2021-06-07	user236	1	1	1-Monday	7	06-June
2021-06-09	user98	435	2	3-Wednesday	9	06-June
2021-06-09	user354	5	2	3-Wednesday	9	06-June
2021-06-10	user7	1	1	4-Thursday	10	06-June
2021-06-10	user172	32	1	4-Thursday	10	06-June

Figure 26: Density Model application over Access Anomalies

3.3.2 LocalOutlierFactor Algorithm

Local Outlier Factor uses scikit-learn Local Outlier Factor (LOF) to measure the local density deviation of a particular sample from adjacent neighbors. It runs the training data once and returns outliers by fitting on the training data. LocalOutlierFactor is the unsupervised form of outlier detection. The score of anomaly depends on how separated an object is from its adjacent neighbors.

Moreover,it uses k-nearest neighbors to give the locality, local density is calculated over the distance of locality. A comparison between local densities of all the neighbouring samples and a single particular sample data gives the outlier, outliers are the sample data that are having lower density than its neighbors.

Parameters:

- **anomaly_score**: it can be either True or False. By default its True, it can be disabled by adding the keyword False to the SPL command.
- n_neighbors: it define the number of neighbors sample to be considered. By default the value is 20.
- algorithm: this defines the algorithm that will be used to compute nearest neighbors. The different algorithms are
  - BallTree
  - brute-force search
  - KDtree

  the the algorithm is set as auto parameter, it will decide the most appropriate algorithm on the basis of the values passed to fit command.
- leaf_size: it is passed to KDTree or BallTree. It affects the speed of query, moreover it affects also the memory required to store tree.
- contamination: it should be in the range 0.0 to 0.5. The default value is 0.1.
- p: it is called as Minkowski metric. If p=1, manhattan_distance(l1), and euclidean_distance(l2). if p=2, minkowski_distance(l_p).

The output value inside is_outlier as 1 for the outliers, and for inliers it will be -1.
Syntax:

```
fit LocalOutlierFactor fields
n_neighbors=int
leaf_size=int
p=int
contamination=float
algorithm=str
anomaly_score=true or false
```

The problem with this algorithm that it does not support saving the model and so the model cannot be applied to new data. Also the prediction is not possible using this algorithm as it predicts on the basis of n_neighbours.

As the model cannot be saved so we will build model inside the Access anomaly application of Splunk:

```
| tstats dc(Authentication.dest) as Dest_count count as Failures from datamodel=Authentication where Authentication.action=fail* AND Authentication.user!="-" AND
    Authentication.user!="unknown" AND Authentication.user!=*$ AND Authentication.dest=*.consoft.it groupby _time span=1d, Authentication.user
| rename "Authentication.user" as user
| eval "atf_hour_of_day"=strftime(_time, "%H"), "atf_day_of_week"=strftime(_time, "%w-%A"), "atf_day_of_month"=strftime(_time, "%e"), "atf_month" = strftime(_time
    , "%m-%B")
| eventstats dc("atf_hour_of_day"),dc("atf_day_of_week"),dc("atf_day_of_month"),dc("atf_month")
| eval "atf_hour_of_day"=if('dc(atf_hour_of_day)'<2, null(), 'atf_hour_of_day'),"atf_day_of_week"=if('dc(atf_day_of_week)'<2, null(), 'atf_day_of_week'
    ),"atf_day_of_month"=if('dc(atf_day_of_month)'<2, null(), 'atf_day_of_month')
| fields - "dc(atf_hour_of_day)","dc(atf_day_of_week)","dc(atf_day_of_month)","dc(atf_month)"
| eval "_atf_hour_of_day_copy"=atf_hour_of_day,"_atf_day_of_week_copy"=atf_day_of_week,"_atf_day_of_month_copy"=atf_day_of_month,"_atf_month_copy"=atf_month
| fields - "atf_hour_of_day","atf_day_of_week","atf_day_of_month","atf_month"
| rename "_atf_hour_of_day_copy" as "atf_hour_of_day","_atf_day_of_week_copy" as "atf_day_of_week","_atf_day_of_month_copy" as "atf_day_of_month"
    ,"_atf_month_copy" as "atf_month"
| fit LocalOutlierFactor Failures n_neighbors=20 algorithm=auto metric=minkowski p=1 contamination=0.1 leaf_size=30
| where isOutlier=1
| table _time, user, Failures, Dest_count, atf_day_of_week, atf_day_of_month, atf_month
| lookup mask_users_new.csv user OUTPUT newuser as user
```

Figure 27: SPL to create the LocalOutlierFactor algorithm model

After building the model it have been applied to the filtered data and it have been seen the output(figure:27) that it is giving the 35 users as the outliers out of all the failures.

✓ 29,335 events (05/06/2021 00:00:00.000 to 05/07/2021 15:08:39.000) No Event Sampling ▾ Job ▾ Fast Mode ▾

Events Patterns Statistics (35) Visualization

50 Per Page ▾ Format Preview ▾

_time	user	Failures	Dest_count	atf_day_of_week	atf_day_of_month	atf_month
2021-06-07	user98	2170	1	1-Monday	7	06-June
2021-06-08	user7	161	5	2-Tuesday	8	06-June
2021-06-08	user72	108	3	2-Tuesday	8	06-June
2021-06-08	user98	673	1	2-Tuesday	8	06-June
2021-06-09	user98	435	2	3-Wednesday	9	06-June
2021-06-10	user98	342	1	4-Thursday	10	06-June
2021-06-14	user142	2	2	1-Monday	14	06-June
2021-06-15	user72	108	3	2-Tuesday	15	06-June

Figure 28: LocalOutlierFactor algorithm model output

As we don't have the possibility to save the model it will be detecting the outliers where the field isOutlier=1, after detecting the outliers the the values will be reported back to the dashboard for the visualization and comparison purpose between the multiple machine learning algorithms.

3.3.3 OneClassSVM Algorithm

OneClassSVM is based on scikit-learn OneClassSVM, model is fitted over the set of features or fields. It detects outliers and anomalies, whose feature are having numerical values. It is an unsupervised outlier detection method..

Parameters:

- **kernel**: it help to decide the kernal type to be used in the algorithm. different kernal types include:
  - linear
  - rbf (Radial Basis Function)
  - poly (Polynomial)
  - sigmoid
- **nu**: it is used to define the upper bound on the training error fraction and lower bound on the support vectors fraction.
- **degree**: It have to be defined if using the polynomial kernel.
- **gamma**: it helps to specify the single data instance influence.
- **coef0**: its an independent term used when using sigmoid or polynomial function.
- **tol**: helps to specify tolerence for defining the stopping criteria.
- **shrinking**: it is used to define the if we need to use the shrinking criteria or not it can be either True/ False.

Syntax:

fit OneClassSVM fields into model name
kernel=str
nu=float
coef0=float
gamma=float
tol=float
degree=int
shrinking=true or false

Now the model OneClassSVM have been fitted inside the Splunk MLTK over the data of last 90 days and saved as OCS_outlier model, which will be further applied inside the Access Anomalies Application to detect outlier.

```
| tstats dc(Authentication.dest) as Dest_count count as Failures from datamodel=Authentication where Authentication.action=fail* AND Authentication.user!="-" AND
   Authentication.user!="unknown" AND Authentication.user!=*$ AND Authentication.dest=*.consoft.it groupby _time span=1d, Authentication.user
| rename "Authentication.user" as user
| eval "atf_hour_of_day"=strftime(_time, "%H"), "atf_day_of_week"=strftime(_time, "%w-%A"), "atf_day_of_month"=strftime(_time, "%e"), "atf_month" = strftime(_time
   , "%m-%B")
| eventstats dc("atf_hour_of_day"),dc("atf_day_of_week"),dc("atf_day_of_month"),dc("atf_month")
| eval "atf_hour_of_day"=if('dc(atf_hour_of_day)'<2, null(), 'atf_hour_of_day'),"atf_day_of_week"=if('dc(atf_day_of_week)'<2, null(), 'atf_day_of_week'
   ),"atf_day_of_month"=if('dc(atf_day_of_month)'<2, null(), 'atf_day_of_month'),"atf_month"=if('dc(atf_month)'<2, null(), 'atf_month')
| fields - "dc(atf_hour_of_day)","dc(atf_day_of_week)","dc(atf_day_of_month)","dc(atf_month)"
| eval "_atf_hour_of_day_copy"=atf_hour_of_day,"_atf_day_of_week_copy"=atf_day_of_week,"_atf_day_of_month_copy"=atf_day_of_month,"_atf_month_copy"=atf_month
| fields - "atf_hour_of_day","atf_day_of_week","atf_day_of_month","atf_month"
| rename "_atf_hour_of_day_copy" as "atf_hour_of_day","_atf_day_of_week_copy" as "atf_day_of_week","_atf_day_of_month_copy" as "atf_day_of_month"
   ,"_atf_month_copy" as "atf_month"
| fit OneClassSVM Failures kernel="poly" nu=0.5 coef0=0.5 gamma=0.5 tol=1 degree=3 shrinking=f into OCS_outlier
```

Figure 29: SPL for creating OneClassSVM algorithm model

After, applying the model over the last 30days of data it gives the output with a new column named as isNormal, each case where the value of isNormal=-1 is considered as outliers.

Failures	_time	user	Dest_count	atf_day_of_week	atf_day_of_month	atf_month	isNormal
176	2021-06-05	user85	1	6-Saturday	5	06-June	-1
178	2021-06-06	user85	1	0-Sunday	6	06-June	-1
15	2021-06-07	user7	2	1-Monday	7	06-June	-1
175	2021-06-07	user85	1	1-Monday	7	06-June	-1
2170	2021-06-07	user98	1	1-Monday	7	06-June	-1
1	2021-06-07	user126	1	1-Monday	7	06-June	-1
4	2021-06-07	user165	1	1-Monday	7	06-June	-1
1	2021-06-07	user201	1	1-Monday	7	06-June	-1

Figure 30: OneClassSVM algorithm model output

After training all the models based on Machine learning algorithms for Anomaly detection, a dashboard have been created inside the Access anomaly application of the Splunk enterprise, where all the application and their output have been reported inside the different sections of the dashboard. The detail description of the dashboard have been reported in next chapter, where we will see the evaluation of the densityfunction model and the comparison between output of the different machine learning models.

4 Experimental Results

4.1 Overview

In this chapter we will evaluate all the results found out by using different machine learning algorithm models. The detail description of the dashboard "Outlier in number of destination" which is saved as the home page for the Splunk enterprise Access anomaly Application have been also discussed. Moreover, the effect of each algorithm over the authentication data will be discussed. Also, the comparison between the results found by using the machine learning algorithm and the pre-existing mathematical formula is discussed. The mathematical formula have been used by consoft sistemi earlier to detect the outliers.The concording and discording results have been reported in the dashboard between machine learning algorithm and pre-existing math solution of consoft sistemi.

4.2 Evaluation over the Outliers

The Outliers calculated by using different machine learning algorithms have been seen in previous chapter. As, each of the algorithms have its own kind of effect on the Authentication data after the filtering.

Moreover, as seen all the models couldn't be saved inside the splunk, in particular LocalOutlierFactor algorithm doesn't give the possibility to save the model so everytime the its model will be fitting for detecting outliers, so with the more number of fitting with every load increases the load to the splunk and also we don't have possibility of getting the insight inside the model. Moreover, LocalOutlierFactor algorithm doesn't give predict method for the new data set for outlier detection. LocalOutlier factor only compares abnormality score of one sample with its neighbours to detect the event as outlier.

OneClassSVM is sensitive for the outlier detection, and its performance is not very well. It can be better suiting for other novelty detection where the data to be trained is not contaminated with the outliers. It means,outlier detection with high-dimension, or with no assumptions over the distribution of inlying data can be very complex. Also, in OneClassSVM we cannot inspect model learned by OneClassSVM with summary command.

DensityFunction Algorithm give the possibility of saving the model as well as inspecting the model. So the model can be fitted once and can be applied later just using the "apply" command of SPL. Moreover, it gives the inspection of model by using "summary" command of SPL. The *Summary* command of SPL which will help us to do the inspection of the model give the access to following insights:

- the number of data points which are used for fitting the density function is represented by cardinality.
- generated distance value tells about metric type used for calculating the distance and about the distance between the sampled data from density function and training dataset.
- mean represents the mean generated by density function.
- std is the standard deviation of density function.
- other field tells about any parameters which are applied other than mean and std.

- type field shows the chosen density function and if the dist parameter was set to auto for particular function.

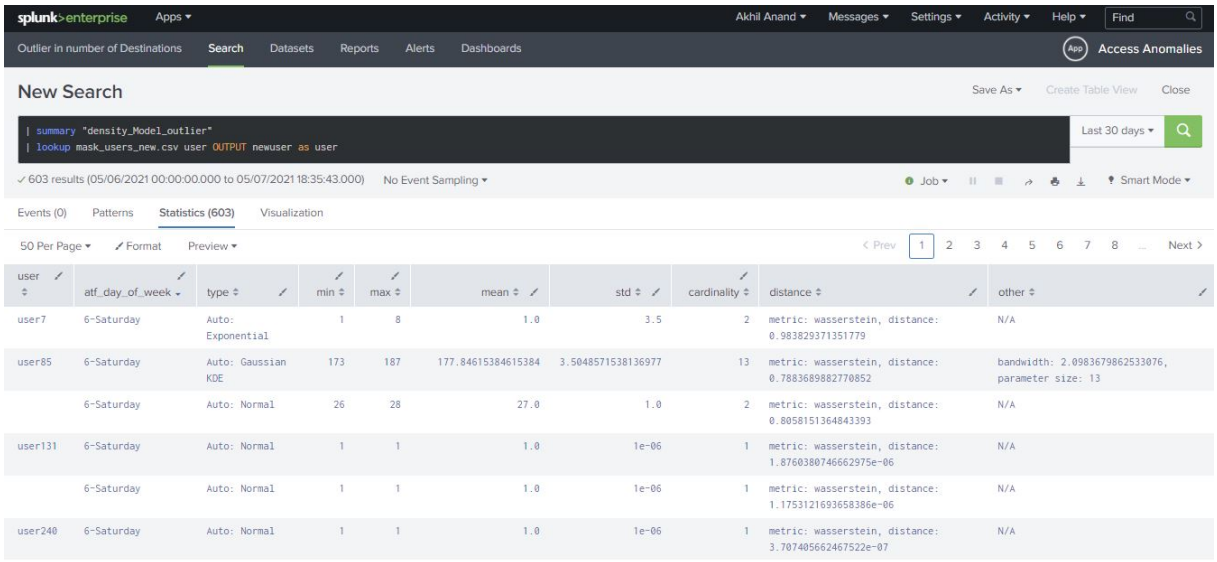


Figure 31: Density Function model summary

In the figure 31 it can be found all the details why a particular user is considered as outlier. Also the type of density function used for a particular event as each row is the failure event that have been occurred by the user.

4.3 Comparison over the Evaluation

After finding out the outliers with the machine learning models using the Density-Function, OneClassSVM and localOutlierFactor Algorithms a visualization in between them have been added inside the home dashboard of Access anomalies Application with the total number of count of outliers that have been found by each of these algorithm have been added in 3 different sections..

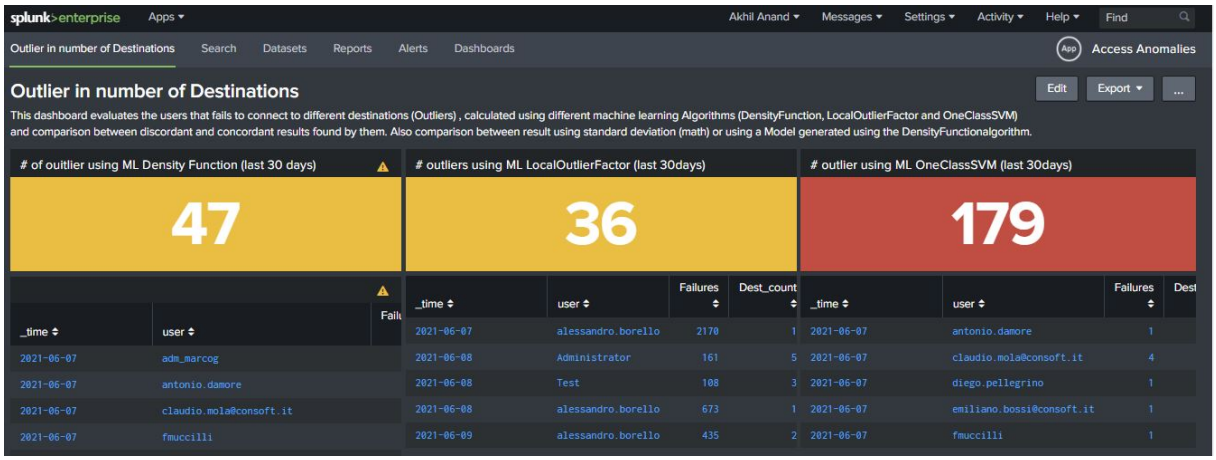


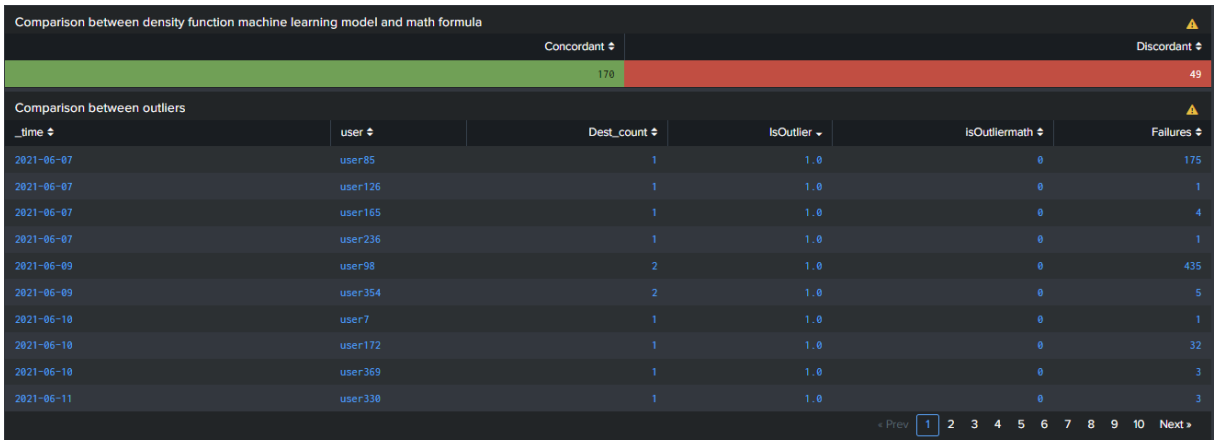
Figure 32: Dashboard outlier in Number of destination

In above figure 32, leftmost section represents outliers that have been found using Density Function Algorithm, middle one reports outliers found using LocalOutlierFactor

algorithm and last and rightmost one is showing outliers using the OneClassSVM Algorithm. Below count section, a section for detailed table view of the users that are outlying have been also added with the functionality of onclick search, which will give access to more broader view about the event of that particular failure of each user. The onclick function also gives the access to the SPL search command that are used for creating each particular section of the dashboard.

Moreover, after looking to the possibility and efficiency the density function algorithm have been seen to be functioning better as compared to other two with more possibility of in depth inspection over the trained model and dataset using the summary command. The main reasons while comparing with Local outlier factor algorithm is that, we can save the model using the into command and use the model later on so we don't need to fit every time the model which reduce the load over the splunk. Moreover, OneClassSVM gives the possibility to save the model as in DensityFunction algorithm but it does not give the option of inspecting the results using the summary command.

So, we will compare densityFunction algorithm based model outliers to the simple maths based outliers which was previously used by Consoft sistemi Spa. A comparison of concordant and discordant results have been found as following:


Comparison between density function machine learning model and math formula

Concordant	Discordant
170	49

Comparison between outliers

_time	user	Dest_count	IsOutlier	isOutliermath	Failures
2021-06-07	user85	1	1.0	0	175
2021-06-07	user126	1	1.0	0	1
2021-06-07	user165	1	1.0	0	4
2021-06-07	user236	1	1.0	0	1
2021-06-09	user98	2	1.0	0	435
2021-06-09	user354	2	1.0	0	5
2021-06-10	user7	1	1.0	0	1
2021-06-10	user172	1	1.0	0	32
2021-06-10	user369	1	1.0	0	3
2021-06-11	user330	1	1.0	0	3

« Prev 1 2 3 4 5 6 7 8 9 10 Next »

Figure 33: Concordant and Discordant outlier

The comparison in the figure 33 shows that the there are many concording results and also 49 of the events that are discordant between them. The field IsOutlier in the table represent the outliers found by using the DensityFunction Algorithm whereas the field IsOutliermath is the outlier which have been found by using the math equation where simple mean and standarad deviation of the mathematical formula have been used for calculation of outlier by the consoft sistemi spa earlier.

The concording and discording results are also comparing the inliers as well as outliers in both cases. So in many cases it have been found that concording results are the inliers in both the cases where as discording results are mostly the cases where the user is considered as outlier with machine learning method. One of the main reason of discording results is because of maths system is considering very few number of users as outliers because it was considering the users with specific calculation of maths based on mean and standard deviation.

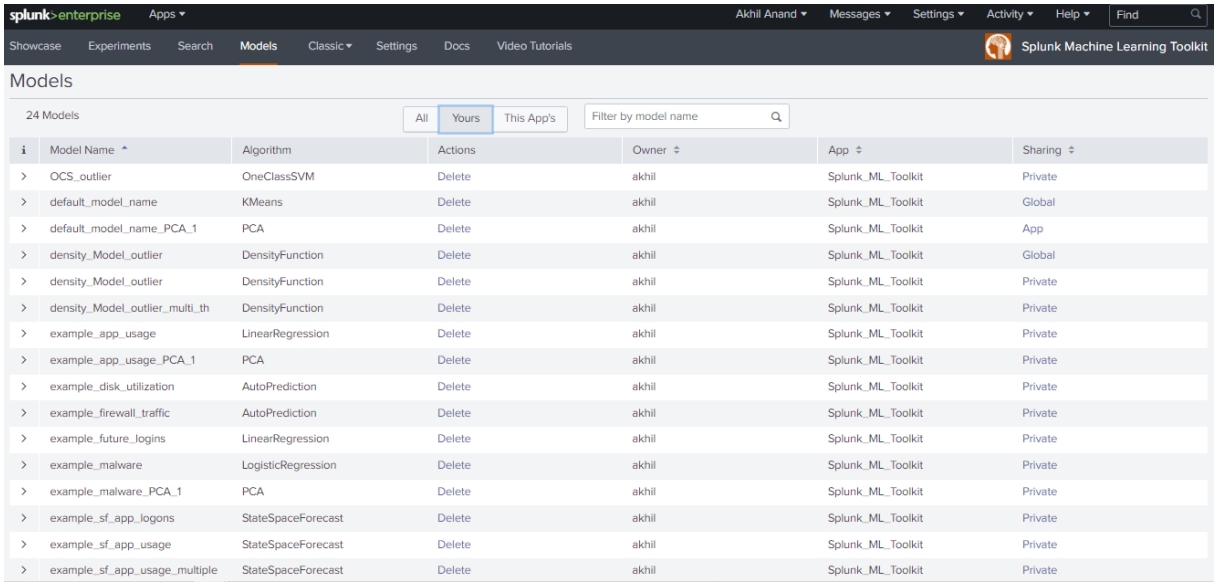
Maths system is not considering many users as outlier even if the number of outliers are quite high as compared to their previous behavior to the system, moreover in some cases it didn't learn the user behaviour and considers the user as outlier even if the user

is having many failures for each access earlier. This problem is solved when using the machine learning model.

Machine Learning model learns the behavior of each user on the basis of their past access to the system. Each time if the user's access behavior towards system changes it is considered as outlier. So it have been found more number of outlying user through this, and is more secured for any safety critical system.

4.3.1 Models fitted inside the Splunk MLTK

This section reports the different models that have been saved inside the Splunk Machine learning toolkit application of the splunk enterprise. These models are used to apply it for some applications over the different datasets. Here we can see that apart from other algorithms used for different experiments the main two algorithms that have been saved are present namely "OCS_outlier" using the OneClassSVM and "Density_model_outlier" using the density function algorithm that was used for the main purpose of this experiment.(figure:34)



i	Model Name	Algorithm	Actions	Owner	App	Sharing
>	OCS_outlier	OneClassSVM	Delete	akhil	Splunk_ML_Toolkit	Private
>	default_model_name	KMeans	Delete	akhil	Splunk_ML_Toolkit	Global
>	default_model_name_PCA_1	PCA	Delete	akhil	Splunk_ML_Toolkit	App
>	density_Model_outlier	DensityFunction	Delete	akhil	Splunk_ML_Toolkit	Global
>	density_Model_outlier	DensityFunction	Delete	akhil	Splunk_ML_Toolkit	Private
>	density_Model_outlier_multi_th	DensityFunction	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_app_usage	LinearRegression	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_app_usage_PCA_1	PCA	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_disk_utilization	AutoPrediction	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_firewall_traffic	AutoPrediction	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_future_logins	LinearRegression	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_malware	LogisticRegression	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_malware_PCA_1	PCA	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_sf_app_logons	StateSpaceForecast	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_sf_app_usage	StateSpaceForecast	Delete	akhil	Splunk_ML_Toolkit	Private
>	example_sf_app_usage_multiple	StateSpaceForecast	Delete	akhil	Splunk_ML_Toolkit	Private

Figure 34: Different models that have been saved inside the Splunk MLTK

The rightmost column of this table shows the sharing option, which can be set to private, app or Global. This enables the usability of the different models throughout the splunk environment. If the haring option is just private it can be only used by the user who have access to it, whereas if it is set to the App it can used inside that particular application whenever required, and when it is set to global it can be shared to all the applications available inside the splunk Environment. So if we give global access to the models we can create the model inside the Splunk Machine learning toolkit and use the same model inside other applications as we have implemented in our case, where the model is trained inside Splunk MLTK and have been used inside the Access Anomalies application. Moreover, if we create the model inside Splunk MLTK we have a option to see what are the already trained model, whereas in other application we don't have the direct possibility to view trained models.

4.3.2 Alert created inside the Splunk Access Anomalies

After, creating the dashboard and reporting all the required details for the understanding the final step is to inform the admin. There are many possibility that are available to sent the alert, it can sent using the email to the admin with the details and also it can seen in the system whenever the conditi bnon that have been created for the alert is met. Here we can define in which cases we need to sent the alert to the admin.

The alert that have been created for the dashboard Outlier in number of destinations is visible in the very first row of the figure 35.

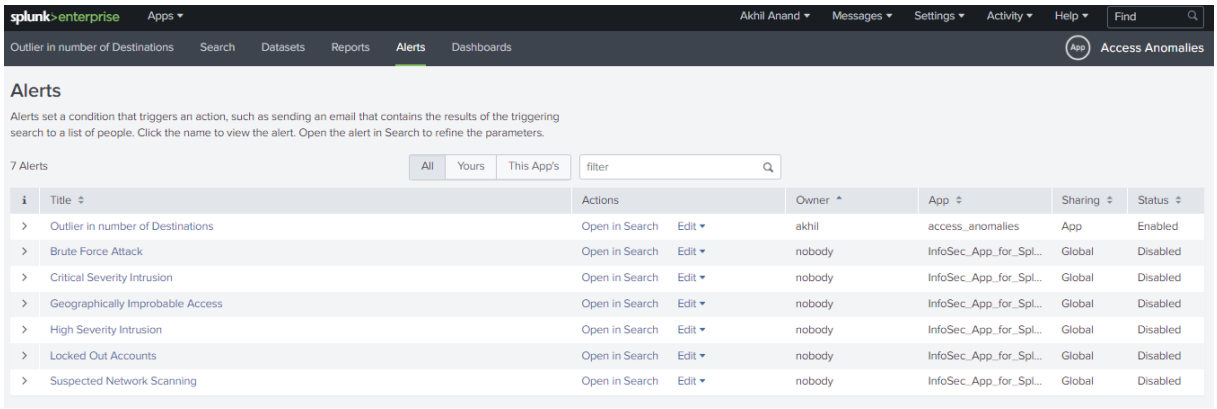


Figure 35: Alert from the Dashboard Outlier in number of destination

Moreover, the alerts can be disabled or enabled when required by setting the status to disable or enable. Also the alerts have the same option of sharing as we have seen for the models inside the Splunk MLTK.

5 Conclusions

In a nutshell, the purpose of this thesis was to cover the topic of machine learning algorithms applications available for detecting the anomalies inside the splunk enterprise infrastructure. Also the perspective that was pursued here was in a way reversed from ordinary, as the aim was to identify the user that are outlying, where each outlying user details have been found through the windows log event which are saved inside the one of the datamodel inside the Splunk enterprise environment of Consoft sistemi which is named as Consoft security lab. Novel finding from the Splunk infrastructure is the outlier detection efficiency related to the failure log events and its visualization to the user with the use of machine learning techniques and Splunk processing Language. The anomalous trends are presented and analysed, and Splunk MLTK application and its implementation were thoroughly covered. The experimental part of this work is related to the case study of Splunk Enterprise simple web service with the uses of Machine learning from above mentioned point of view.

The experimental setup used in the project was designed according to the need of the Consoft sistemi Spa, but naturally limited by the software that was readily available. The development of the experiment was built from scratch for the purpose of thesis around this subject. This was certainly not the fastest way to detect the anomaly from the algorithms used, as there are many other service providers which offers the virtual machine like Splunk Enterprise at low costs. However, getting to know the possibility of the Splunk Applications in different field like Security, IT, Finance, Business analytic, Health , IOT and many others provided me the inner view of applications of virtual machine infrastructure in the real world. Building an maintaining own experiments also enables trying practically anything, when the outsourced service providers doesn't provide customers with all the information relevant for the purpose of project.

The Experiments in this project work provided me some valuable information about how machine learning algorithms behave in certain web service applications: the amount of splunk server memory capacity ceased to be the bottleneck at relatively low number of events which have to be trained by machine learning algorithm. So, in this case the Splunk enterprise server becomes limited by the other applications, such as failing the model to run on even on the small snippet of the data at early stage, training the model indefinitely without any output as it loses the memory from the end of the other application.

The final conclusion come to end with suggestion of best machine learning algorithm that can be applied in place of normal mathematics that have been used by consoft sistemi spa for detecting the outlying users for every failures. The model that have been found to find the best outcome for the failure is based on the Density function Algorithm out of the other all the three available algorithm inside the splunk enterprise.

References

[1] Splunk, "Splunk Overview", [Online].
Available on: https://www.splunk.com/. Accessed in 2020.

[2] App, "Splunk App documentation", [Online]
Available on: https://docs.splunk.com/Splexicon:App. Accessed in 2020.

[3] Splunkbase, "Available apps on Splunk", [online]
Available on: https://splunkbase.splunk.com/apps/#/type/app. Accessed in 2020.

[4] Splunk Dev, "Splunk Developer Platform", [online]
Available on: https://dev.splunk.com/. Accessed in 2020.

[5] Pivot, "Pivot Editor", [online]
Available on: https://docs.splunk.com/Splexicon:Pivoteditor. Accessed in 2020.

[6] SPL, "Search Processing Language", [online]
Available on: https://docs.splunk.com/Splexicon:SPL. Accessed in 2020.

[7] "StateSpaceForecast Algorithm",[Online]
Available on: https://docs.splunk.com/Documentation/MLApp/5.2.1/User/Algorithms#StateSpaceForecast. Accessed in 2021.

[8] "DensityFunction Algorithm", [Online]
Available on: https://docs.splunk.com/Documentation/MLApp/5.2.1/User/Algorithms#DensityFunction. Accessed in 2021.

[9] "KMeans Algorithm", [Online]
Available on: https://docs.splunk.com/Documentation/MLApp/5.2.1/User/Algorithms#K-means. For descriptions of default value of K, see the scikit-learn documentation at https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Accessed in 2021.

[10] "AutoPrediction Algorithm", [Online]
Available on: https://docs.splunk.com/Documentation/MLApp/5.2.1/User/Algorithms#AutoPrediction. Accessed in 2021.

[11] "RandomForestClassifier Algorithm", [Online]
Available on: https://docs.splunk.com/Documentation/MLApp/latest/User/Algorithms#RandomForestClassifier. See the scikit-learn documentation at http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.Accessed in 2021.

[12] "Ultimate Windows Security", [Online]
Available on: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx. Accessed in 2021. Its a division of Monterey Technology Group, Inc.