1. **Time-Window Data Retrieval**

The first step of our pipeline consists in collecting Twitter messages from the Tweets Database which were posted within a given time range. Considering that our objective is to provide a continuous threat identification and alerting system, the time range will be a sliding time window considering the end time of the previous time range as the start time for the next time range. All the resulting Twitter messages will follow to the Unknown Word Selection, described in the next subsection.

1. **Word Selection**

The objective of this component is to identify unknown words or terms appearing in collected Twitter messages as they, accordingly to further analysis, may represent the name of the identified emerging threat. The idea of identifying those unknown terms came from the analysis of cyber threats names, which usually receive very strange names - either given by their creators of by the cyber security experts who first spotted them. Wannacry, NotPetya, Cookthief, Emotet, lokibot, and 16shop are some examples of threat names. For the proposed architecture, a term is considered unknown if it passes through the Unknown Word Selection pipeline, which comprises the following procedures: Normalization, URL/E-mail/Author filtering, NLTK Word Tokenize, Correct Word Filtering, Stop-words and punctuation filtering, NER (Named Entity Recognition) filtering and, finally, Dictionary words filtering, as described below.

1. **Normalization**

Considering that we are using Twitter messages posted by a variety of people and that Twitter itself imposes a length limit for the post message (nowadays 280 characters), it is very common to have terms for the same meaning written and shortened in different forms. For example, ’C2 server’, ’C&C server’ are written in different forms but, in the context of cyber security, mean the same thing: ’command and control server’. Command and control servers are computers controlled by an attacker or cybercriminal which is used to send commands to systems compromised by malware and receive stolen data from a target network [39].

1. **NLTK Word Tokenize**

This step consists in splitting each collected tweet into words. The process of splitting sentences into words or just word tokenize is very commonly used by natural language processing solutions. To employ word tokenization into the proposed solution, we use Natural Language ToolKit (NLTK) Python module 11. The output of this step is, for each tweet, an array of its words or tokens. See in the example below how the content of a tweet is split into tokens: Tweet: "The RobbinHood ransomware is using a vulnerable legacy Gigabyte driver in order to get around antivirus protections".

1. **NER Filtering**

After applying the above filters in the pipeline, we noticed that, among unknown words, there were many organizations’ names like Microsoft, Google, and so on. Although they were really unknown words for the filters used until this point of the pipeline, we should eliminate them because, knowingly, they did not represent threat names. There is a field called Named Entity Recognition (NER) which is considered a fundamental task in a natural language processing (NLP) system. NER is a subproblem of information extraction and involves processing structured and unstructured data to identify expressions that refer to people, places, organizations, and companies [40]. Thus, applying NER to our pipeline would help reduce the number of companies being considered ’unknown words’.

1. **TF-IDF**

In this subsection, we describe the use of TF-IDF (Term Frequency-Inverse Document Frequency) to transform the text documents coming from both MITRE ATT&CK corpus and Twitter messages into a vectorized representation needed by both One-Class and Multi-Class machine learning (ML) algorithms - the next steps in the pipeline. Machine learning algorithms, more specifically the ones used in this work, operate on a numeric feature space, expecting input as a two-dimensional array where rows are instances and columns feature. To perform ML on the text we need to transform our documents into vector representations such that we can apply numeric machine learning in a process called feature extraction or vectorization [43]. To perform the feature extraction, we employed a method called TF-IDF (Term Frequency-Inverse Document Frequency) [44]. TF-IDF is a numerical representation of the importance (weight) of a term t in a specific document d within a corpus of documents.

1. **Classification process**

* **Logistic Regression Algorithm**

It is a SML model that is very commonly or widely used for the classification. Performance of LR model for linearly separable classes is very well and even easy to implement. Specially, in industry it is most commonly used. In general LR is used for binary classification as it is a linear model but using technique OvR it may be used for classification of multi class [9]. LR is applied on dataset by considering three different train test ratio (80:20, 60:40, and 70:30) to predict whether the bank currency is forge or genuine. For train test ratio 80:20 ROC curve and learning curves are drawn. Accuracy of LR is observed around 98% .

* **Decision Tree Algorithm:**

It is a classification model having a structure like a tree. DT is incrementally developed by breaking down the data set into smaller subsets. DT results are having two types of nodes Decision nodes and leaf nodes. For an example consider a decision node i.e., Outlook and it have branches as Rainy, Overcast and Sunny representing values of the tested feature. Hours Played i.e., a leaf node it gives the decision on numerical targeted value. DT can handle both numerical as well as categorical data [8]. DT is applied on dataset by considering three different train test ratio (80:20, 60:40, and 70:30) to predict whether the bank currency is forge or genuine. For train test ratio 80:20 ROC curve and learning curves are drawn. Accuracy of DT has been observed around 99%.

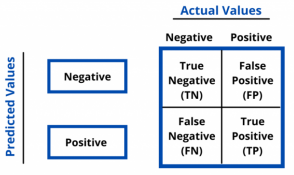
* **Random Forest Algorithm**

Random Forest is that the prevalent supervised technique. it's useful for mainly doing classification challenges and also regression challenges. RF is one amongst the classifiers which holds multiple decision trees in each subset of an assumed data set and computes the everyday value that enhances prediction accurateness for the dataset. The random forest doesn't depend upon decision trees. Instead, it gets a prediction from every tree so forecasts the last result which is made upon polls of prevalence estimations. The more trees within the forest, the upper the accuracy and avoid over fitting problems. it's supported the ensemble technique concept, which mixes multiple classifiers to unravel a thorny problem and improves model performance.

1. **Performance Evaluation**

Once the model has been built the accuracy of the model has to be evaluated by the performance metrics in deep learning and machine learning methods. We have used F1-Score, precision, recall, confusion matrix and accuracy score.

***Confusion Matrix:*** Confusion matrix is a very intuitive cross tab of actual class values and predicted class values. It contains the count of observations that fall in each category. Build a model → make class predictions on test data using the model → create a confusion matrix for each model.



**Figure: Confusion matrix**

***Accuracy:*** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:



***Precision:*** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:



***Recall:*** It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.



***F1-measure:*** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

