Contents

Introduction .................................................................................................................. 2
Defining the Problem .................................................................................................. 2
The Use of Machine Learning for Intelligence Analysis ........................................... 3
TISA™ Text Analysis and Feature Extraction .............................................................. 4
TISA™ NLP Tokens .................................................................................................... 5
Handling Structured vs. Unstructured Data ................................................................. 6
TISA™ Structured Text Data Analysis ....................................................................... 6
Combining Ontology, Entities & Knowledge Base using TISA™ ................................. 7
TISA™ Machine Learning Algorithm ......................................................................... 9
Building the TISA™ Model – Step by Step ............................................................... 10
Summary .................................................................................................................... 11
Introduction

This paper presents the TISA™ methodology, a model of Threat Intelligence Scoring and Analysis, which redefines the ability to provide targeted contextual intelligence. The model has been built upon the unique combination of knowledge of CyberInt’s cybersecurity experts, intelligence analysts, PhDs in artificial intelligence, and data scientists. This methodology enables automated and effective, real-time identification and prioritization of targeted threat intelligence artifacts (“Indicators”). By employing the TISA™ methodology, the Argos™ threat intelligence platform is able to process tens of millions of indicators per day, gathered from tens of thousands of sources across the Internet, Deep Web and Dark Web, to provide timely, accurate, and actionable threat intelligence.

An indicator is a data structure, which includes text content (i.e. a tweet on Twitter or a post on a hackers’ forum), metadata (date, source name, author, title, etc.) and enriching fields meant to give additional information about every indicator.

Using TISA™ methodology, significant and cyber-related indicators are identified through machine learning algorithms and are further analyzed and enhanced in order to extract additional intelligence and insights of the data, while ignoring large amounts of “noise” and unrelated information. Ultimately, this methodology provides the ability to integrate real-time actionable threat intelligence into cybersecurity centers, enhancing the efficacy of detection and response processes.

Defining the Problem

Cybersecurity operation teams are struggling to effectively incorporate actionable cyber threat intelligence into their defense systems and processes. One of the major obstacles in achieving the above is the wealth of distracting and mostly irrelevant threat intelligence, usually acquired through general “feeds” and “crawlers”. For a human to make sense out of this data is virtually impossible.

Therefore, a key ability in any intelligence platform is to quickly and effectively sift through large quantities of intelligence and classify it for relevance. Hence, given an indicator, we need an algorithm to automatically decide whether it’s interesting (cyber-related) or not. Moreover, there is a need to attribute a score to indicate the indicator’s severity (or interest) level.
The Use of Machine Learning for Intelligence Analysis

The problem defined above is a classification problem, i.e. we need to classify the input indicator as cyber-relevant (Positive) or not (Negative). A suitable solution is using Machine Learning (ML) techniques to build a classifier model to handle this task.

**Machine Learning Definition:**
- A field of study that gives computers the ability to learn without being explicitly programmed
- Belongs to the fields of artificial intelligence and pattern recognition
- Includes elements from computer science, statistics and mathematics
- Applied on (almost) any field such as: finance, medicine, cyber security, social, physics, sports

**TISA™ employs two types of ML algorithms:**
- Supervised Learning – Building a model based on data samples and their corresponding labels (i.e. tagged data)
- Semi Supervised Learning – Building a model using a small amount of labeled data with a large amount of unlabeled data

The data samples are the indicators and are being labeled by two categories: cyber-relevant and not cyber-relevant. The two categories mean binary classification.

Employing TISA™ machine learning capabilities, the system constantly improves its classification accuracy based on the tagging, which is made by the system’s end-users, intelligence analysts, and input.
TISA™ Text Analysis and Feature Extraction

The most important part of the indicator is its content, which may come from forums posts, Facebook comments, paste sites, dumps and others. This textual content should be automatically analyzed and should uncover some initial insights.

Analyzing those texts from diverse sources creates numerous challenges, some of which are:

- Many different languages to analyze (50+)
- Contents with “noise” data (insignificant characters, ASCII art, etc.)
- Database dump structures
- Slang, shortcuts and hash tags

TISA™ overcomes these challenges by using NLP (Natural Language Processing) as described below.
**TISA™ NLP Tokens**

In the algorithmic text analysis and NLP (Natural Language Processing) terminology, a “token” is a sequence of characters extracted by breaking a stream of text. To differentiate from words, a token may also be a phrase, URL, email address, hash, IP address, and so on.

The process of breaking a stream of text into tokens is called Tokenization. Tokenization can be applied also to extract pairs or trios of tokens called 2-grams or 3-grams respectively. 2-grams can allow us to treat “New York” as a single token instead of two, and 3-grams can allow us to treat “Denial of Service” as one token instead of three. The basic tokenization is called unigram.

I'm a hacker planning to attack cyberint.com next week

I'm a hacker planning to attack cyberint.com next week

I'm a A hacker Hacker planning planning to To attack

Attack cyberint.com Cyberint.com next Next week
Handling Structured vs. Unstructured Data

Analytics processes are usually applied on structured data. Structured data set is a collection of data samples that can be represented in a matrix or table form. Each row in the table is a set of values representing a single data sample. The table columns represent the values of the table fields (referred to here as “features”). The dataset is represented as a matrix: #examples × #features. The number of features is effectively the data dimension.

<table>
<thead>
<tr>
<th>indicators/features</th>
<th>feature_1</th>
<th>feature_2</th>
<th>...</th>
<th>feature_p</th>
</tr>
</thead>
<tbody>
<tr>
<td>indicator_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indicator_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indicator_n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TISA™ Structured Text Data Analysis

TISA™ employs the “Bag of Words” model to turn words into meaningful numbers, which can then be ingested into an algorithm to prioritize the severity of the indicators. According to this model, each token represents a feature, and Stop words (words with very little significance) are removed (e.g. are, to, a, etc).
Further, word frequencies in a document are computed, and normalized using the “tf-idf” weighting method:

- $tf-idf$ (Term Frequency - Inverse Document Frequency)
- $Tf$-$idf$: Avoid text length bias and weight by how frequent a token is used in all documents
- $Tf(t) = \frac{frequency \ of \ a \ term \ “t” \ in \ document}{total \ number \ of \ terms \ in \ document}$
- $Idf(t): = \log(\frac{total \ number \ of \ documents}{number \ of \ documents \ with \ term \ “t” \ in \ them})$
- $Tfidf(t) = tf(t) \times idf(t)$

<table>
<thead>
<tr>
<th></th>
<th>Unsecure</th>
<th>Network</th>
<th>SQL Injection</th>
<th>Exposure</th>
<th>Mail Files</th>
<th>Msdb</th>
<th>System</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Weight</td>
<td>0.71</td>
<td>0.12</td>
<td>0.78</td>
<td>0.615</td>
<td>0.49</td>
<td>0.72</td>
<td>0.89</td>
<td>0.47</td>
</tr>
</tbody>
</table>

The result is a numeric indicator, which represents the indicator’s properties to be analyzed by the ML algorithm. To get a complete understanding of an indicator, the Title and Author fields are also analyzed, as they tend to include useful (and even crucial) information.
Combining Ontology, Entities & Knowledge Base using TISA™

While the Bag of Words structure can represent the data, the TISA™ model employs additional custom features to get more accurate and extensible analysis capabilities.

- **Key tokens**: Using highly complex regular expressions, TISA™ enables automatic discovery, count and normalization of special tokens (i.e. email addresses, URLs, and IP addresses).

- **Entity types**: An “Entity” is a pre-defined token, indicating this token has specific cyber security context (e.g. “DDoS”, “hack”, “password”, etc.). Each entity may have multiple aliases (in different languages) or be defined as a regular expression. An entity is represented by a data structure with multiple fields to describe it (e.g. threat actor’s name, url, email, etc.) The entities are divided into different types: threat actors, attack tools, campaigns, etc. When discovering a token, which is an entity, it gets an additional weight in the classification task.

- **Knowledge base**: The knowledge base is a dataset containing all entities (reaching thousands of them).

- **TISA™ has a built-in Ontology.** The ontology describes a world of entities by dividing a space into different categories and defining the connections between them in a hierarchical structure (mostly represented by graphs). TISA™ maps the knowledge base entities, divides them into significant categories and subcategories, and defines the relations between them.

TISA™ defines categories such as threat actor, attack techniques, and victims. TISA™ allows for a flexible and customized definition of the categories according to the organization’s needs. Hence, TISA™ ontology can be mapped easily to other models (such as the Kill-chain model or the Diamond Intelligence Model). It can even be extended to physical attributes such as geolocation.

**Example of simple Ontology structure:**

![Example of simple Ontology structure](image-url)
Employing ontology, TISA™ enables analysts to gain further insights, for example, by identifying nontrivial connections between the found entities, and by identifying connections between each entity and other entities in the knowledge base.

TISA™ Machine Learning Algorithm

TISA™ employs the Linear SVM (Support Vector Machine) algorithm. This algorithm finds the optimal maximum margin plane equation, separating between two labels (in our case it's 1 for cyber related indicators and -1 if not). SVM is a powerful algorithm, which can handle a very high number of features, provides good prediction, and is fast. To further extract the most out of the algorithm, an optimization process is done by applying grid search on the parameters the algorithm is depending on.

In the following example for SVM results, the red and blue points represent the cyber relevant and “not relevant” indicators. The plane in green is the boundary separating the different points.
Building the TISA™ Model – Step by Step

TISA™ model is constructed of a full cycle that comprises the following steps:

1. **Data Import**
   - Import tagged data
   - Filter data to get relevant fields
   - Additional data pre-processing is applied

2. **Train and Test Sets**
   - Train set: Used to build the model itself
   - Test set: Used for testing the built model
   - The splitting proportion is usually 75/25 for train/test sets

3. **Feature Extraction**
   - Create a Bag of Words structure
   - Reweight the Bag of Words values according to tf-idf
   - Extract additional features
   - Find all entities and mark their types
   - Find additional entities connections using Ontology

4. **Train ML Model**
   - Use the train set extracted features as an input for the ML algorithm
   - Save the model for predictions
   - Model Testing
   - Use the created model to predict the test set indicators

**Re-Learning**

The above flow is an ongoing process, which is applied each time a significant amount of high quality indicators are extracted and tagged. This process allows for two main advantages: Ongoing model improvement, enriching the “Bag of Words” vocabulary constantly, and keeping the model updated with every new cyber-security entity (e.g. new threat actor emerged, new attack techniques, new victims, etc.), as it emerges in the ever-changing cyber world.

**Summary**

The TISA™ methodology provides an effective model for scoring and analyzing masses of threat intelligence indicators in real-time, based on advanced machine learning and natural language processing. While TISA™ is flexible enough to support other theoretical intelligence models, it provides a practical solution for the real-world challenges facing cyber-intelligence analysts confronting a large volume of indicators in different languages with various levels of relevance and for different customers. Ultimately, these advantages translate to better, and more efficient, Cyber Operation Centers performance, effectively enabling intelligence-led cybersecurity.