Sentiment Clustering:
A Hybrid Approach for Insider Threat Detection

Rawabi AlQahtani
Upstream Digital Center
Saudi Aramco
Dhahran, Saudi Arabia
rawabi.alqahtani@aramco.com

Hafiz M. Farooq
Upstream Digital Center
Saudi Aramco
Dhahran, Saudi Arabia
hafiz.farooq@aramco.com

Ahmad Khoraidly
Upstream Digital Center
Saudi Aramco
Dhahran, Saudi Arabia
ahmad.khoraidly@aramco.com

***Abstract* - Sentiment Moderation has resurfaced as an effective technique for detecting insider threats after the rise of Large Language Models (LLMs). However, applying negative sentiment analysis algorithms in real time on security big data on-prem is an emerging challenge. Therefore, in this paper we utilize unsupervised Numerical Clustering (K-Means) to cluster the network behavior of enterprise users and identify noisy and malicious behaviors through Natural Language Processing (NLP) algorithms. This proposed hybrid ML-NLP approach facilitates security analysts or threat researchers selecting anomalous features (users) and apply negative sentiment analysis in a much more scalable way. We performed empirical and statistical analysis to earmark the performance gains achieved through our new proposed hybrid insider-threat detection approach.**

Keywords - Sentiment Analysis, Machine Learning, NLP, Cybersecurity, Insider Threats, Cyber Threat Detection, Security Operations, Data Leakage

I. INTRODUCTION

In the era of Large Language Models (LLMs), Sentiment Moderation has emerged as a powerful technique for detecting insider threats. Natural Language Processing (NLP) algorithms can therefore discover hidden indicators related to the suspicious activities of disgruntled employees and external attackers during the different phases of cyberattacks. However, the real-time application of negative sentiment analysis algorithms on security and big data presents an ongoing challenge. Larger enterprises usually have terabytes of security logs corpus that requires sentiment moderation, while processing this security big data on premises (mostly due to data privacy and regulatory reasons). Therefore, we believe there is a requirement to retrofit existing sentiment and opinion mining techniques and propose a technique that could handle real-time processing and scalability problems in large enterprise environments.

This paper addresses these problems by presenting a new hybrid approach that leverages (K-Means) clustering and NLP algorithms, where K-Means groups the network behavior of users, then NLP identifies the noisy and malicious behavior. Combining ML and NLP in this hybrid approach offers a scalable solution, enabling security analysts and researchers to identify anomalous features (events) and apply negative sentiment analysis more effectively.

II. LITERATURE REVIEW & LIMITATIONS OF CURRENT TECHNIQUES

Most of the previous work related to sentiment related numerical clustering was mostly related to opinion mining and social-media related negative sentiment detections only. Zul et al. [1] and Villanueva1 et al. [2] have recently used unsupervised clustering algorithms to detect negative sentiments in social media posts on the X (Twitter) platform. We have extrapolated similar clustering methodology to important cybersecurity datasets (normally collected from an employee's enterprise endpoints). We believe sentiment analysis is not only suitable for opinion mining, but also very effective in discovering the zero-day cyber-attacks and security vulnerabilities persistent in the enterprise network.

A. Literature Review

The power of unsupervised numerical algorithms in clustering is to enable the extraction of useful information from datasets by analyzing their structure to discover meaningful relationships and associations (as explained by Syraif et al. [3]), to detect the hidden patterns without the need for labeled or supervised training models. As a result, these clustering algorithms produce and develop a model that is capable of accurately predicting labels for raw data based on their features [4].

Unsupervised clustering empowers (not only cybersecurity) but also a wide range of applications such as customer segmentation, anomaly detection, topic modeling, and recommendation systems. In addition, clustering is used for data reduction purposes. For example, data is partitioned based on a certain criterion, then analyzes the interesting clusters instead of processing all the data and exhausting the system and its resources. Moreover, it is used in hypothesis generation and testing. For example, use a dataset for an online shop and generate the hypothesis and then verify accuracy by applying clustering analysis on the

representative dataset. Furthermore, in business, clustering may help marketers discover significant groups in their customers' database and characterize them based on purchasing patterns. Also, clustering is used in recommender systems to provide automatic personalized suggestions for information, products, and services [5].

There are two types of clustering that are partitioned and hierarchal clustering. In partitioned clustering, the data can be either part of one cluster only (hard clustering) or into the data can be part of multiple groups (soft clustering). There are various partitioned clustering algorithms to cluster the data such as K-Means, which is the most widely used clustering algorithm. In addition to partitioned clusters, hierarchy clustering produces a set of nested clusters that may be arranged to form a tree structure [6] and recommended for relatively smaller and structured cybersecurity datasets.

B. Limitations of Current Techniques

In this age of big data, enterprise need to process petabytes of data to discover negative insider employees in real time. Sentiment Analysis can be effective against such insiders, however processing large volume of datasets is an emerging challenge. Generally, the main objective of sentiment analysis is to transfer the textual information by extracting sentiment, emotions, and opinions, or even intentions from text data and classify them into positive, negative, neutral, and then map the result into actionable insights. Sentiment Analysis doesn't provide a native pre-processing or filtering technique to massage the incoming data before it invokes the sentiment analysis, to label it as positive, negative, or neutral [7].

There are different approaches for tackling sentiment analysis, starting with a lexicon-based approach, which depends mainly on a predefined sentiment dictionary with its sentiment score. Moreover, a machine learning-based approach uses supervised learning models to identify patterns, trends, and features in the text that are associated with different sentiment classes and are commonly used to capture complex patterns and adapt to different domains. After that, the model can predict the sentiment of new and unseen texts [8]. Furthermore, a deep learning-based approach is capable of automatically learning the text data hierarchically, considering contextual dependencies and semantic relationships. Their strength lies in managing long-range dependencies and are particularly useful for understanding sentiments in large text sequences. However, the limitation of the deep learning models is they require large computational resources and training data [9]. In addition, a hybrid approach, which combines multiple techniques, such as lexicon-based, machine learning, and rule-based methods, is used to leverage their respective capabilities and minimize their weaknesses. For example, a hybrid approach uses a lexicon-based approach to assign sentiment scores for the text and then use a machine learning model to enhance the predictions based on context and other factors. This will benefit the sentiment analysis as it will improve the accuracy and robustness [10].

In addition, a hybrid approach combines multiple techniques, such as lexicon-based, machine learning, and rule-based methods, to leverage their respective capabilities and minimize their weaknesses. For example, a hybrid approach uses a lexicon-based approach to assign sentiment scores for the text. Then, it uses a machine learning model to enhance the predictions based on certain criteria. The combinations will benefit the sentiment analysis as it will improve the accuracy and robustness [11]. Figure-1 below summarizes the sentiment analysis approaches

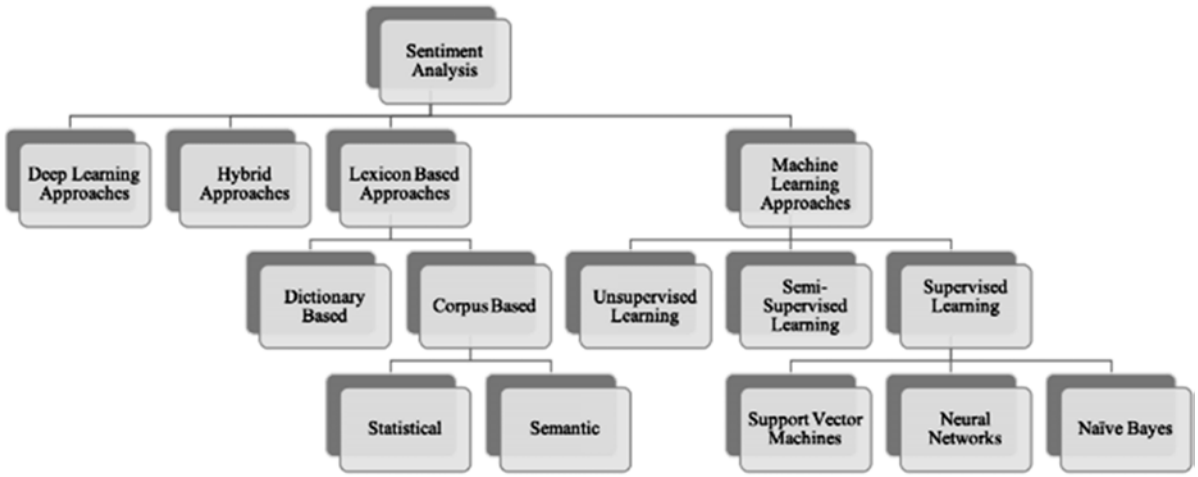

Figure 1. Sentiment Analysis Approaches

III. OUR PROPOSED NEW TECHNIQUE

Our proposed technique is a two-step analysis, explained in figure-2. As a first step, we used unsupervised *K-Means numerical clustering algorithm* for grouping the enterprise insiders based on their behavioral features, and then subsequently applying the *sentiment analysis algorithm* only on the abnormal (noisy) cluster to discover any zero-day negative security sentiment. Suspicious clusters are selected based on their noisy behavior (explained in next section) on the enterprise assets (endpoints, network, and applications).

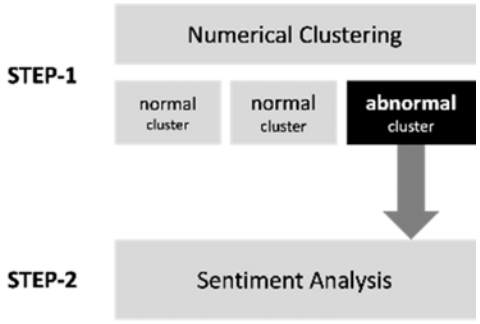

Figure 2. Euclidean Distance Formula

This approach is suitable for security big data environments, where industry NLP solutions are not generally powerful enough to perform the sentiment analysis on tera/petabytes of enterprise data corpus received in real time. Therefore, we believe filtering the security big data first and selecting only abnormal clusters for sentiment analysis will reduce the data size and increase performance of the threat detection pipeline.

A. Step-1: Detect Abnormal Clsuter using K-Means

Processing large amounts of security data corpus in real time is challenging and requires a significant number of resources and time. In essence, the selection of a clustering algorithm hinges on the dataset's specific attributes and the clustering task's demands. K-Means static clustering algorithm is a go-to for simplicity and effectiveness in handling well-separated, spherical clusters. Other dynamic clustering algorithm, like DBSCAN shines in spotting clusters of any shape and dealing with noise. Hierarchical clustering is optimal for discerning nested structures but is less effective with large datasets. GMM, offering adaptable and probabilistic cluster assignments, is suitable for complex cluster shapes [12].

Therefore, the unsupervised K-Means algorithm is used to cluster the data into a predetermined number of clusters that are represented as "*k*". The clustering assignments are refined iteratively until the convergence is achieved and with the objective of minimizing the sum of the squared distances from the cluster centroid. K-Means clustering can be divided in three different steps:

1 - *Centroid Initialization*: Select the number of the cluster (*k*=*n*).

2 - Assignment: Partition the dataset and assign the cluster. The assignment is based on minimizing the sum of squared Euclidean distance (figure 3).

3 - *Centroid Update*: Iterate the assignment and recalculate the centroid by calculating the mean of all data points in each cluster until convergence [13].

$$a_k = \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} z_{ik}}$$

$$z_{ik} = \begin{cases} 1 \; if \; \|x_i - a_k\|^2 = \min\limits_{1 \le k \le c} \|x_i - a_k\|^2 \\ 0, \quad otherwise \end{cases}$$

This technique is used to cluster the enterprise users' activities based on the two features, number of process executions (*nProcesses*) and similarly number of network connections (*connections*).

nProcesses = No of OS processes executed by an OS user / hour
nConnections = No of network connections invoked by OS user / hour

Here these two features (*nProcesses* and *nConnections*) have been selected (per user) empirically, due to their importance for the threat hunting, to detect zero-day malwares and zero-day attackers. Once the users have been clustered (based on *nProcesses* and *nConnections*), the noisy cluster is finally selected based on high cluster-centroid value. Figure-4 (on next-page) displays the result of the K-means clustering, cluster-2 is a noisy cluster due to its high centroid in the XY-axis.

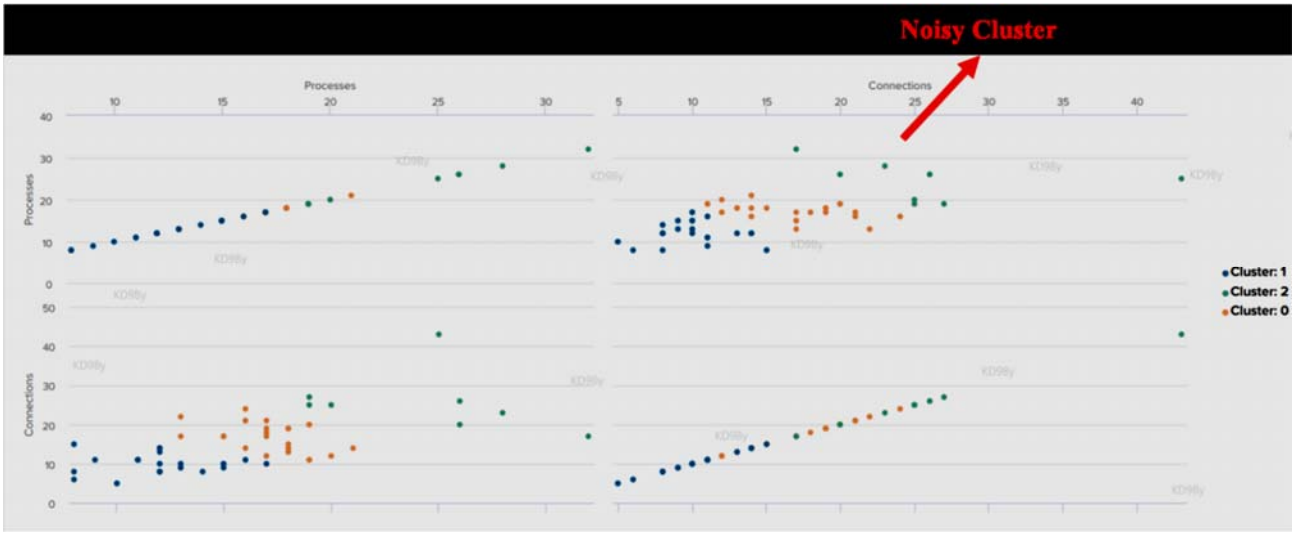

Figure 4. K-Means Algorithm for Noisy Cluster Classification.

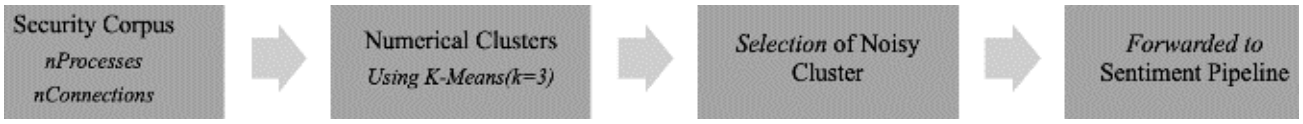

Figure 5. A Framework for Noisy Cluster Classification

The noisy cluster (after its selection process) is forwarded to sentiment analysis pipelines (i.e., VADER algorithm), the workflow is explained in figure-5 above.

The query below (generated using Splunk SPL) in figure-6 explains the technique used for identifying the noisy cluster among all users based on their processes and network connections behavioral datasets. The first step is to statically discrete count the inputs, which are processes and connections to eliminate counting duplicated processes. Secondly, the fit K-Means is used to cluster events based on their similarity using the K-Means clustering algorithm. Thirdly, the event stats are used to display all information related to this event. Fourthly, the max function is used to calculate the highest mean and a where condition is used to limit the result on the noisy cluster. Finally, the result will be saved in csv file, which will be used in the sentiment analysis pipelines (i.e., VADER algorithm).

```
index = network AND host= <hostname>
AND NOT User IN ("UserA", "UserB" , "UserC")
| table _time host User Connections Processes
| stats dc (Processes), dc (Connections)
| fit KMeans k=3 Connections Processes into Processes_Model
| appendpipe
  [ | eventstats mean (Processes) as Mean_Processes,
     mean (Connections) as Mean_Connections by cluster
    | eval Centroid = cluster]
| search Centroid=*
| eventstats max (Mean_Connections) AS Highest_Mean
| where Mean_Connections = Highest_Mean
| outputlookup noisy_cluster.csv
```

Figure 6. Splunk Query for Noisy Cluster Classification

B. Cyber Security Sentiment Dictionary

Sentiment dictionary is used in natural language processing and sentiment analysis. It is a structured list consisting of words and sentiment scores to determine the polarity of the words varying from -4, which indicates that it is extremely negative, to +4, which indicates that it is extremely positive [14]. There are various dictionaries such as WordNet, SentiWordNet, SenticNet, and others [15]. However, we have developed a Cybersecurity Sentiment Dictionary "SAREM" (an Arabic word that means it will strictly catch any suspicious activity). SAREM included cybersecurity lexicons (like backdoor, rootkit, waterhole, etc.) since the default VADER dictionary was more specific to human emotions, instead of negative cybersecurity behaviors. SAREM dictionary was developed by us based on the aggregate negative scores given by senior SOC Analysts during a negative scoring survey. It consists of around 600 words and mainly contains security words, attack names, and other suspicious words that could impact operations negatively and each word and its score was chosen based on the survey result.

The key benefits of the cybersecurity dictionary are to understand and classify the sentiment in the raw data. For example, sentiment dictionaries enable the automation of sentiment analysis, which will speed up the processes efficiently by eliminating the need for manual classification and reducing the investigation time to target only suspicious words.

However, the sentiment dictionaries may not understand the tone of the human sentiment expression, including sarcasm, irony, and context-dependent sentiment. Therefore, the dictionaries require periodic updates to remain accurate and to be aligned with the operational work. Sentiment analysis using a dictionary-based approach is described in Figure-7 below [15].

Figure 7. A Framework for Sentiment Classification

C. Sentiment Analysis using VADER Algorithm

The automation of sentiment analysis is essential as it will speed up the processes efficiently by eliminating the need for manual classification and reducing the investigation time to automatically target suspicious activity only. Different algorithms and methods can be used to exploit these lexicons for sentiment analysis. Each technique varies in complexity and approach, each with unique strengths and weaknesses, depending on the requirements of the task, such as accuracy level, text complexity, and context. At its most basic, the Simple Word Count method determines sentiment by tallying positive and negative words from a lexicon.

The sentiment is inferred from the net difference in these counts [16]. In contrast, the Weighted Word Count approach builds on this by assigning varying weights to words based on their sentiment strength, computing the final score as a weighted sum [17]. Rule-Based Systems introduce a set of predefined rules to this process. These rules, used in conjunction with the lexicon, account for linguistic nuances like negations, intensifiers, and diminishers, thereby refining the sentiment score [18]. Similarly, the Valence Shifters method considers elements like negations and amplifiers that alter a word's sentiment value, as seen in phrases like "not good." [19].

The Semantic Orientation Pointwise Mutual Information (SO-PMI) method utilizes pointwise mutual information to gauge a term's sentiment by its association with positive and negative reference words. SentiWordNet, another technique, employs a lexical resource where each synset in WordNet is tied to sentiment scores, aggregating these to deduce the overall sentiment of text content [20]. Integrating lexicon-based features with machine learning models represents a more sophisticated approach. Here, sentiment scores of words are coupled with other textual features for nuanced classification.

Lastly, Aspect-Based Sentiment Analysis zeroes in on sentiments regarding specific aspects of a product or service. This method combines lexicon-based scores with aspect identification, often achieved through advanced NLP techniques.

IV. EXPERIMENTAL METHODOLOGY

In our proposed technique, the VADER algorithm is used. It's a lexicon-based sentiment analysis tool that quantifies sentiment based on valence, intensity, and polarity. It will be used to extract sentiment, emotions, and opinions, or even intention from text data and classify them into positive, negative, neutral, and then map the result into actionable insights based on the cybersecurity dictionary "SAREM" that was developed internally. The VADER algorithm was chosen among all other techniques due to various reasons such as their simplicity, accuracy, scalability, and customizability to target the organization's needs. Figure-8 shows a sample of SAREM dictionary values.

Word	*Score*
Ransomware	-3.7
Dark Web	-3.6
Hijack	-3.6
Trojan	-3.6

Figure 8. Sample VADER Scoring in SAREM Dictionary

The steps of classification for a word using a VADER algorithm are as follow:

-*1. Tokenization*: This process involves segmenting the text into individual elements, a critical step in identifying key parts for subsequent analysis.

-*2. Normalization and Preprocessing*: This stage transforms all characters to lowercase to ensure uniformity and facilitate normalization. Additionally, it involves removing superfluous or stop words such as "*is*," "*are*," and "*and*," which are less relevant for analysis.

-*3. Lexicon Look-Up*: This is the comparison process of each token (word) against the VADER Lexicon. Each word in the lexicon has a corresponding sentiment intensity score and is labeled according to their semantic orientation as either positive, negative, or neutral.

-4. Composite Score Calculation and Categorization:

This final step to calculate a composite score for the entire text after the contextual adjustment. This score is a normalized, weighted composite score that considers valences of each word and applies the adjustments per VADER's specific rules. Accordingly, text is categorized as either positive, negative, or neutral sentiment [21].

The query below (generated using Splunk SPL) explains the technique used for applying a clustering algorithm (VADER) for the detection of negative security sentiments in different security datasets. The initial step involves filtering processes that interact with specific file formats, such as pdf, doc, xls, ppt, and html. This selective filtering is crucial to target potentially relevant data. Secondly, identify the processes that were listed earlier using machine learning K-Means algorithm. Thirdly, following the identification of relevant processes, the "rex" command in Splunk is utilized for field extraction. This step is pivotal for isolating specific data elements required for sentiment analysis as shown in Figure 9 and 10. Lastly, the VADER algorithm is applied to the extracted fields. VADER, adept at sentiment analysis, categorizes the text into positive, negative, or neutral categories. This final step is instrumental in discerning the overall sentiment of the security-related text data.

```
index=winlogs host=<hostname> 4688 (.pdf OR .doc OR .docx OR
.xls OR .ppt OR .pptx OR .htm OR .html OR .txt)
| mvexpand User
| search NOT User IN("UserA", "UserB" , "UserC")
| table User Process_Name  Process_Command_Line
| join type=inner UserName
    [ | inputlookup noisy_cluster.csv
      | table User Process Connections]
| table User Process_Command_Line
| rex field=Process_Command_Line max_match=0
"(?<extracted_words>([a-zA-Z0-9\s_\\.\-
\(\):])+)\.(pdf|PDF|doc|docx|xls|xlsx|ppt|pptx|html|htm|txt)"
| vader textfield=extracted_words full_output=t
```

Figure 9. Splunk Query for Sentiment Analysis

Figure 10. REX Field Extraction

V. RESULTS AND DISCUSSIONS

The combination of K-Means clustering with VADER sentiment analysis in analyzing large textual datasets like security threats offers notable advantages, but also comes with inherent limitations.

This approach excels in efficiently organizing large textual datasets, where K-Means clusters texts into meaningful groups based on similarity, simplifying analysis. Enhanced sentiment analysis is another benefit, as applying VADER to each cluster reveals insights into prevalent sentiments, particularly useful for datasets with diverse topics. This method ensures contextual sentiment understanding since sentiment analysis through VADER becomes more accurate within homogenized contexts. Additionally, the scalability of this combination is notable, handling large-scale data analysis effectively. It facilitates the identification of trends and patterns, and for businesses, it aids in decision-making and strategy development by analyzing sentiments related to specific topics. Moreover, it allows for targeted analysis and actionable insights by understanding sentiments within specific clusters.

However, challenges exist. K-Means requires a predefined number of clusters, which can be difficult to determine without prior data knowledge. It's also sensitive to the initial placement of centroids and outliers, potentially leading to inconsistent clustering results. K-means operates under the assumption of spherical clusters of similar sizes, which may not reflect real-world data complexities [22]. VADER, while effective for social media texts, may struggle with complex or nuanced expressions and specific jargon. It also provides a broad categorization of sentiments but may lack depth in understanding more complex emotional contexts.

Another issue is the potential overlooking of sentiment variations within clusters by VADER. The effectiveness of both K-means and VADER heavily relies on the quality of text preprocessing. Additionally, VADER may not accurately detect sarcasm or ambiguity. Despite their efficiency, processing very large datasets can still be resource intensive. Finally, sentiment analysis is typically conducted post-clustering without considering each cluster's context, which might lead to misinterpretation of sentiments. In summary, while the integration of K-means and VADER offers a comprehensive approach for text data analysis, balancing its strengths against its limitations is crucial for effective application.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed an innovative hybrid ML-NLP method which utilized the unsupervised Numerical Clustering (K-Means) to cluster the network behavior of enterprise users, followed by identification of noisy and malicious behaviors through NLP algorithms. This empowers security analysts and threat researchers to efficiently identify and analyze anomalous features or users. The paper discussed the unsupervised numerical cluster, sentiment analysis approaches, the development of SAREM dictionary, the technique and its improvement and limitation.

REFERENCES

[1] M. I. Zul, F. Yulia and D. Nurmalasari, "Social Media Sentiment Analysis Using K-Means and Naïve Bayes Algorithm," *2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI)*, Batam, Indonesia, 2018, pp. 24-29, doi: 10.1109/ICon-EEI.2018.8784326.

[2] Iparraguirre-Villanueva, O., Guevara-Ponce, V., Sierra-Liñan, F., Beltozar-Clemente, S., & Cabanillas-Carbonel, M. (2022). Sentiment analysis of tweets using unsupervised learning techniques and the K-Means algorithm.

[3] Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). Unsupervised clustering approach for network anomaly detection. In Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. Proceedings, Part I 4 (pp. 135-145). Springer Berlin Heidelberg.

[4] Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE access*, *8*, 80716-80727.

[5] Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001, July). Clustering algorithms and validity measures. In *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001* (pp. 3-22). IEEE.

[6] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100-108.

[7] Hamid, H. H., Bangare, S., Nirmal, A., & Bangal, S. (2023, August). *Sentiment analysis: A comprehensive review of techniques and applications*. Journal of Emerging Technologies and Innovative Research (JETIR). https://www.jetir.org/papers/JETIR2308292.pdf

[8] Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine learning-based sentiment analysis for twitter accounts. *Mathematical and computational applications*, *23*(1), 11.

[9] Rani, S., & Kumar, P. (2019). Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering*, *44*, 3305-3314.

[10] Vaitheeswaran, G., & Arockiam, L. (2016). Hybrid based approach to enhance the accuracy of sentiment analysis on tweets. *International Journal of Science, Engineering and Computer Technology*, *6*(6), 185.

[11] Khoo, C. S., & Johnkhan, S. B. (2018). Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science*, *44*(4), 491-511.

[12] Zhu, Y., Xu, X., & Yan, Z. (2021). Hybrid clustering□based bad data detection of PMU measurements. *Energy Conversion and Economics*, *2*(4), 235-247.

[13] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100-108.

[14] Hamdan, H., Béchet, F., & Bellot, P. (2013, June). Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 455-459).

[15] Hardeniya, T., & Borikar, D. A. (2016). Dictionary based approach to sentiment analysis-a review. *International Journal of Advanced Engineering, Management and Science*, *2*(5), 239438.

[16] Günther, E., & Quandt, T. (2018). Word counts and topic models: Automated text analysis methods for digital journalism research. In *Rethinking research methods in an age of digital journalism* (pp. 75-88). Routledge.

[17] O'Keefe, T., & Koprinska, I. (2009, December). Feature selection and weighting methods in sentiment analysis. In *Proceedings of the 14th Australasian document computing symposium, Sydney* (pp. 67-74).

[18] Chikersal, P., Poria, S., & Cambria, E. (2015, June). SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 647-651)

[19] Morsy, S. A., & Rafea, A. (2012, June). Improving document-level sentiment classification using contextual valence shifters. In *International conference on application of natural language to information systems* (pp. 253-258). Berlin, Heidelberg: Springer Berlin Heidelberg.

[20] Khan, F. H., Qamar, U., & Bashir, S. (2016). SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. *Applied Soft Computing*, *39*, 140-153.

[21] Bhonde, R., Bhagwat, B., Ingulkar, S., & Pande, A. (2015). Sentiment analysis based on dictionary approach. *International Journal of Emerging Engineering Research and Technology*, *3*(1), 51-55.

[22] Yuan, C., & Yang, H. (2019). Research on K-value selection method of K-means clustering algorithm. *J*, *2*(2), 226-235.