DETECTING AND CLASSIFYING MALWARE IN ELECTRICAL POWER GRIDS VIA CYBERDECEPTION

by

TALLAL MOHAMED OMAR

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND INFORMATICS

2024

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Mohamed A. Zohdy, Ph.D., Chair
Julian Rrushi, Ph.D., Co-Chair
William Edwards, Ph.D.
Eralda Caushaj, Ph.D.
Sara Sutton, Ph.D.

© Copyright by Tallal Mohamed Omar 2024
All rights reserved.

This dissertation is dedicated to my dear mother, who has been my support until my research was fully finished, and my beloved wife, who for years has encouraged me attentively with her fullest and trustiest attention to accomplish my work with truthful self-confidence.

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude to Prof. Mohamed Zohdy and Prof. Julian Rrushi for their ultimate support, assistance, and insightful suggestions during the process of preparing this research project.

I would like to express my gratitude to the members of my committee, namely Dr. Eralda Caushaj, Dr. William Edwards, and Dr. Sara Sutton. I take great pride in having you as a member of my committee. I would like to express my gratitude to all of you for dedicating your time and providing valuable advice. I would also like to express my gratitude to my mother, Zeineb, for the unwavering love and support she has provided me with throughout the course of my life's journey. I would like to extend my sincere gratitude and admiration to my spouse, Qamrah Khalleefah. Her assistance has been crucial in enabling the successful completion of my Ph.D. dissertation. Additionally, I would like to extend my sincere appreciation and gratitude to my siblings for the unwavering support they have provided me with throughout the course of my existence. Lastly, I thank everyone who participated in my research and contributed across my research area. I would like to express my gratitude to the instructors and staff members affiliated with the Department of Computer Science and Electrical Engineering, as well as the Business School at the University of Oakland, for their invaluable collaboration and advice throughout my academic journey.

Tallal Omar

ABSTRACT

DETECTING MALWARE IN THE ELECTRICAL POWER GRID VIA DEFENSIVE CYBERDECEPTION

By

TALLAL MOHAMED OMAR

Adviser: Mohamed A. Zohdy, Ph.D., Char
Advisor: Julian Rrushi, PhD, Co-Char

Artificial intelligence (AI) has become an essential instrument for enterprises aiming to protect their digital assets within a progressively aggressive cyber landscape. As the dependence on digital technologies increases among companies and individuals, the risks associated with cyberattacks are also advancing in terms of complexity and magnitude. AI and the proliferation of technology has led to a significant concern over security, mostly due to the escalating prevalence of malware on industrial computers. This has resulted in potential physical harm to computer systems and the individuals involved. Malware is a collection of malicious programming code that aims to inflict harm against computer systems, programs, or online apps. These applications lack the ability to differentiate between legitimate system calls and those that are intended to cause harm. Therefore, it is imperative to ensure that computer systems and online applications are constructed in a manner that enables the identification and differentiation of malicious activities from legitimate application activities. The utilization of AI in the realm of cybersecurity is revolutionizing the domain of digital

protection. There are various techniques that can be used to identify malicious activity, leveraging innovative concepts such as AI, machine learning, and deep learning. The present study presents a proposal for utilizing AI approaches to identify and mitigate malware activity in computer memory, with the aim of safeguarding against unauthorized access to and manipulation of physical data within the system. This research aims to combine the traditional K-means algorithm with other methods and functionalities to perform data aggregation tasks on a physical dataset. The primary objective is to identify anomalies in the dataset using clustering techniques. These anomalies will serve as triggers for creating a replica of the main process as a decoy thread. The decoy thread will be equipped with decoy sensors and actuators. The analysis will be conducted on the decoy thread rather than the main process, allowing for intrusive observation. The same host environment will be provided to memory-resident malware, enabling it to continue operating within the main operating system process. The analysis process involves utilizing a replicated instance of malware that resides within a deceptive thread.

TABLE OF CONTENTS

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WCSS	Within Cluster Sum of Squares
PCA	Principal Component Analysis
Cov	Covariance Matrix
API	Application Programming Interface
GPU	Graphics Processing Unit
CPU	Central Processing Unit
PIN	Personal Identification Number
IC3	Internet Crime Complaint Center
SIEM	Security Information and Event Management
LDAP	Lightweight Directory Access Protocol
CGI	(computer-generated intelligence)
AI	(artificial intelligence)
ML	Machine Learning
DL	Deep Learning
EPG	Electrical Power Grid
IPC	Inter-Process Communication
SSE	Sum of Squared Errors
ICS	Industrial Computer Systems

CHAPTER ONE

INTRODUCTION

This research thesis aims to present an effective classification module for the detection of malware activity on industrial computers. It is important to note that industrial computers differ significantly from commercial desktop computers due to their specialized deployment use cases. While the core components of an industrial computer could be similar to those found in typical desktop computers, such as the CPU, memory, and storage, they include particular tough design elements designed for ensuring durability and precision in industrial automation units. The elbow technique and the K-means++ algorithm were used in this study to find the right number of clusters and check the purity and accuracy of the data classification, even when small changes were made. This project used Principal Component Analysis (PCA), a widely used technique in data analysis that enables the transformation of a dataset comprising potentially correlated variables into a new collection of variables referred to as principal components. Subsequently, the newly introduced collection of variables can be subjected to analysis. The main objective of principal component analysis (PCA) is to establish linear independence among the principal components and effectively capture a substantial portion of the variation inherent in the original dataset. The process of dimensionality reduction is utilized to decrease the number of variables or features in a dataset. The study incorporated the utilization of machine learning in three primary stages. The initial phase of this investigation involves the process of scaling and standardization, which has been used to ensure uniformity in the dataset. Furthermore, the dataset dimension was

reduced using main component analysis. In addition, the elbow approach was used to ascertain the optimal number of clusters, denoted as k, while the calculation and documentation of the sum square error (SSE) were utilized to determine the appropriate number of clusters. In addition, the k-means algorithm was utilized to determine the suitable value of clusters, denoted as k, to be used as an input for the algorithm on the physics dataset.

This research addresses the significance of implementing the elbow approach prior to utilizing the k-means++ algorithm by considering two cases. This approach demonstrates efficacy in enhancing the performance of the k-means++ clustering algorithm [1]. The second observation indicates a deviation in the clustering behavior of the physical data following the initiation of malware activity. This malware manipulates the physical dataset in order to cause physical harm to the power transformer, primarily by impairing the operators' ability to perceive the transformer's condition. This is achieved through the suppression of protection algorithms, which allows the malware to exploit and compromise controllable aspects of the power transformer. In the context of artificial intelligence (AI) and machine learning (ML) applications, the identification of an anomaly inside a clustering process serves as a prompt to quickly initiate the creation of a decoy thread [2]. Once the decoy thread is operational, it will light a flag, thereby initiating the monitoring of any malware activity within the thread. The primary objective of this monitoring is to mitigate any potential physical harm that may be inflicted upon the power transformer.

1.1 Supervised and Unsupervised and Reinforcement ML

The concept of machine learning (ML) refers to a broad variety of methods and processes for extracting information from data in order to create predictions, detect patterns, or enhance decision-making. This information can be used for a variety of purposes. Supervised learning, unsupervised learning, and reinforcement learning are the three separate subfields that can be extracted from the umbrella field of ML [3]. Regression and classification are the two fundamental actions that are carried out during the process of supervised learning. The difficulties that are involved with clustering and association rule mining can be solved through the process of unsupervised learning. On the other hand, reinforcement learning places more emphasis on either exploiting or exploring the environment.

1.1.1 Supervised Learning

Supervised learning is an instance of machine learning wherein an algorithm is trained to make predictions or judgments based on a dataset with labels. Within the framework of supervised learning, the term "dataset" pertains to a compilation of input-output pairs. In this context, the "inputs" correspond to individual data points, while the "outputs" encompass either labels or goal values [4]. The main goal of supervised learning is to develop a mapping or model that can effectively predict the output of novel inputs that have not been previously observed. Supervised learning works in the following manner:

1.1.1.1 Data Collection

It is imperative to assemble a dataset comprising instances of input data alongside their corresponding output or labels that are deemed suitable for these instances. In the context of spam email classification, it is common to possess a dataset of emails accompanied by corresponding labels denoting their classification as either spam or non-spam.

1.1.1.2 Training

The dataset is partitioned into two separate components: the training set and the testing set. For the purpose of training the model, researchers utilize the training set, whereas the test set is utilized to evaluate how well the model is doing.

1.1.1.3 Model Training

The training dataset serves as the input for the machine learning algorithm, commonly referred to as a model. This enables the selection and instruction of the algorithm. Furthermore, the machine learning system is trained to provide predictions by leveraging the input data in conjunction with their corresponding labels.

1.1.1.4 Forecasting

After the completion of the training process, the model becomes capable of making predictions or classifications on datasets that were not previously encountered. For example, it can be utilized to ascertain the presence of spam in incoming emails.

1.1.1.5 Categorization

After the completion of the training process, the model can be used to categorize datasets that were previously unobserved. The model's performance can be evaluated by testing it on a separate dataset, known as the test dataset, in order to assess its

generalizability to unseen data. The selection of appropriate assessment metrics depends on the specific characteristics of the issue under evaluation. In many cases, accuracy and precision are utilized as measures.

Supervised learning is commonly used in numerous applications, such as natural language processing, picture classification, and medical diagnosis. Additional uses encompass recommendation systems. Linear regression, decision trees, support vector machines, and deep neural networks are often employed technologies in the field of supervised learning.

The efficacy of supervised learning hinges on the presence of a labeled dataset that possesses both high-quality and representative characteristics, the careful selection of suitable models and features, and the optimization of hyperparameters. The usefulness of this method lies in its versatility; it can handle a wide range of tasks well because it facilitates clear definition of desired inputs and outcomes.

1.1.2 Unsupervised Learning

Unsupervised learning refers to a category of machine learning wherein the algorithm is trained on unlabeled datasets. The major objective of this approach is to identify patterns, structures, or relationships within the data without any predetermined target or output variables. Unsupervised learning, in contrast to supervised learning, where models are trained to generate predictions or classifications, is primarily concerned with grasping the underlying structure of the data. Unsupervised learning works in the following manner [5].

1.1.2.1 Data Collection

It starts with gathering a dataset. However, within the context of unsupervised learning, it is important to note that the dataset must lack labeled target values or outputs. The dataset mainly contains processed data points.

1.1.2.2 Model Training

Unsupervised learning methods are utilized to detect and analyze structures and patterns within the dataset. In accordance with the specified categorization system, unsupervised learning falls under the subcategory of Clustering which refers to the utilization of algorithms that aim to group and categorize data points based on their similarities into distinct clusters or groups. The K-means clustering technique is widely recognized as one of the prominent algorithms employed in clustering.

1.1.2.3 Dimensionality Reduction

The primary objective of this technique is to decrease the number of features (dimensions) in the dataset while preserving the most important information. One of the techniques utilized for dimensionality reduction is principal component analysis (PCA).

1.1.2.4 Anomaly Detection

Anomaly Detection refers to the application of unsupervised learning algorithms in order to identify and detect instances within a dataset that deviate from the predicted patterns. One-class support vector machines (SVMs) and isolation forests are frequently used for the purpose of anomaly detection.

1.1.2.5 Insights and Analysis

The unsupervised learning model may find useful information about the structure of the dataset as soon as it starts processing the data. Clustering enables the identification of distinct clusters of data points exhibiting similar characteristics.

1.1.2.6 Utilizations

Unsupervised learning can be used in many different ways in natural language processing, such as to divide customers into groups, compress images, find strange patterns, and create models of people. This approach proves particularly valuable in instances where one seeks to examine and grasp the evidence prior to making a conclusive determination regarding a certain course of action, or in situations where a clear and predetermined objective is not readily apparent.

1.1.3 Reinforcement Learning (RL):

Reinforcement learning is a form of machine learning wherein a computational algorithm engages in interactions with its environment, undergoing training to make a sequence of decisions or actions, and subsequently receiving feedback in the form of rewards or penalties [6]. The primary objective of reinforcement learning is to instruct the agent in the process of making a sequence of decisions that result in the maximization of cumulative benefits over a period of time. This training methodology effectively conveys the ability of the agent to act optimally within the specific environment it operates in.

1.1.3.1 Agent

Refers to an individual or entity that possesses the capacity to make decisions and actively interact with its environment. The system will be provided with an observation,

and it will thereafter react to the training by executing actions and making judgments based on the received answers.

1.1.3.2 The Environmental Context

The environment of an agent pertains to the external system with which it engages in interactions, sometimes denoted as "the environment." The agent receives input that is contingent upon its behaviors, and this input might manifest as either positive reinforcement or negative consequences.

1.1.3.3 State

The state expresses a depiction of the present condition of the ecosystem or its general well-being. The agent's capacity to comprehend its surroundings and make suitable decisions is contingent upon the state. This is the reason why the state holds significant importance.

1.1.3.4 Action

Refers to the assortment of choices available to an agent for the purpose of interacting with the surrounding environment.

1.1.3.5 Policy

Refers to a strategic approach or framework employed by an agency to determine the appropriate course of action in relation to a given circumstance. Both determinism and stochasticity are valid approaches to policy.

1.1.3.6 Total Rewards

The concept of total rewards refers to when the environment provides the agent with a numerical signal that represents the evaluative feedback on the agent's most recent behavior in the present state. The signal mentioned is commonly known as the total

rewards. The primary goal of the agent is to maximize its cumulative reward over a given period of time.

1.1.3.7 The Value Function

The value function provides an estimation of the anticipated total reward that an agent can obtain from a given state or state-action pair. This tool aids the agent in evaluating the desirability of different states and behaviors within the environment.

1.1.3.8 Q-Learning

Q-Learning is a reinforcement learning strategy used to acquire knowledge on the value associated with performing a specific action in a given environment. The utilization of this approach is frequently observed in Q-learning algorithmic frameworks. Reinforcement learning encompasses two major categories of algorithms: model-free and model-based.

1.1.3.8.1 Model-Free Learning.

These algorithms do not depend on a pre-existing model to facilitate their learning process. Instead, they acquire knowledge directly through their experiences and interactions with their surroundings. Deep neural networks serve a purpose within both Q-learning and deep reinforcement learning to approximate value functions.

1.1.3.8.2 Model-Based

The decision-making process for these algorithms commences by constructing a dynamic model of the environment, which is subsequently employed. The model depicts the manner in which the environment reacts to the actions performed by the agent. The utilization of model-based reinforcement learning can prove beneficial in scenarios where

the computing cost or potential hazards associated with environment modeling are significant.

1.2 Traditional K-means algorithm.

In order to get a comprehensive understanding of the optimal approach for data clustering and classification, it is imperative to initially familiarize ourselves with the fundamental structure of the k-means method. The k-means algorithm is an iterative procedure that aims to reduce the sum of squared distances between data points and their respective centroids. Calculating the squared Euclidean distances between each data point and the given centroids completes the task. The technique utilized is an unsupervised learning approach utilized for the purpose of data clustering. The technology has an assortment of possible applications, including but not limited to consumer segmentation, image compression, and anomaly detection. The subsequent section presents a delineation of the architectural framework along with a comprehensive elucidation of the sequential procedures entailed in the k-means algorithm [7].

1.2.1 The process of initialization

The process of initialization requires multiple steps, the first being determining how many clusters will be utilized.

1.2.1.1 Determining the Number of Clusters (k)

Prior to proceeding, it is necessary to establish the desired number of clusters into which the data will be divided. The determination of this hyperparameter necessitates the utilization of preexisting domain information or the implementation of techniques such as the elbow approach or silhouette analysis.

1.2.1.2 Initialization of Centroids

The next step is to randomly select k data points from the dataset to be used as the initial centroids for each cluster.

1.2.2 Assignment Step

The objective of this assignment phase is to implement and comprehend the K-means++ clustering method. The main goal is to investigate the subtleties of this technique, which enhances the conventional K-means strategy by optimizing the initialization of cluster centroids. This stage entails examining the algorithmic intricacies, implementing it on pertinent datasets, and evaluating its effectiveness in terms of convergence and clustering accuracy. Furthermore, we will focus on the practical aspects of parameter adjustment and result interpretation, aiming to develop a thorough comprehension of the K-means++ method and its practical applications in real-world situations.

1.2.2.1 Applying Euclidean Distance

Typically represented as a Euclidean distance, is computed to determine the separation between every k data point in the dataset and each of the centroids.

1.2.2.2 Assignment of Data Points

The assignment of a data point to a cluster should be based on its proximity to the centroid that represents that cluster.

1.2.3 Update Step

During the update phase of the assignment, the primary focus is to enhance and streamline the existing model or method. This entails fine-tuning variables, integrating input, and optimizing the overall efficiency. Within the framework of K-means++, the update phase entails the process of reassigning data points to the cluster centroid that is closest to them, and subsequently recalculating the centroids based on the revised assignments. The iterative procedure is to modify the clustering solution, with the goal of achieving convergence and enhancing accuracy. The update step is an essential phase in the assignment, which enhances the ongoing enhancement and efficiency of the applied method.

1.2.3.1 Arithmetic Mean

The next step is to compute the arithmetic mean of the data points assigned to each cluster, followed by the reevaluation of the cluster centroids. The recently identified centroids serve as representations of the cluster centers for each respective category.

1.2.4 Convergence Check

The convergence check is a crucial step in the assignment, verifying that the iterative procedure has achieved a stable solution. inside the framework of K-means++, the evaluation entails determining if the method has effectively minimized the sum of squares inside each cluster, suggesting that additional iterations are unlikely to result in substantial alterations. This stage eliminates superfluous computational endeavors and indicates that the algorithm has effectively recognized stable cluster assignments. The convergence check is an essential quality control method that guarantees the

effectiveness and dependability of the K-means++ clustering algorithm in producing precise and significant outcomes.

1.2.4.1 Optimal Number of Clusters

It is important to determine whether there has been a significant change in the centroids between each consecutive repeat. If the centroids haven't changed much or if a certain number of iterations have been completed, the algorithm has converged, and the process can be ended.

1.2.4.2 Dynamic Iterations

If the focuses have moved, the next step is to keep running Assignment and Update until they converge again.

1.2.5 Output

The "User Output" phase entails the presentation of the conclusive outcomes and discernments derived from the executed procedure. Within the framework of K-means++, this involves presenting the improved cluster allocations and centroid positions. Users can analyze these outputs to acquire significant insights into the intrinsic organization of the data, enabling informed decision-making or additional investigation. This pivotal stage connects the technical components of the algorithm with the real-world uses, offering customers a concise and significant depiction of the grouped data.

1.2.5.1 Centroid Representations

The ultimate centroids represent the centered positions of the clusters.

1.2.5.2 Cluster Labels

The assignment of data points to clusters determines the cluster labels for each data point.

1.2.6 Variations and implications

Examining diverse algorithmic variants and their consequences is crucial for comprehending the subtleties of various methodologies. Modifications, whether in the settings of parameters or in the algorithms themselves, can have a substantial effect on performance and outcomes. Examining these discrepancies enables researchers to determine the most favorable setups for particular tasks and datasets, hence improving the flexibility and efficiency of the selected algorithms. Furthermore, analyzing the consequences of these adjustments provides insight into the wider range of situations where the algorithms can be applied and the constraints they may have, thereby enhancing our overall understanding of their usefulness in different settings.

1.2.6.1 Limitations of k-means algorithm

The conventional k-means algorithm exhibits certain limitations, including its susceptibility to the selection of initial centroids and its underlying assumption that clusters are spherical and of equal size. These aforementioned criteria represent only a subset of the total constraints outlined earlier. Several limitations can be overcome by using modified versions and improvements of the k-means method, including k-means++, mini-batch k-means, and hierarchical k-means.

1.2.6.2 Preprocessing and Scaling

Performing preprocessing and scaling on data is crucial to ensuring that all features have equal influence on distance computations.

1.2.6.3 Distance Metric

The selection of a proper distance metric is of utmost importance in various domains. Although the Euclidean distance metric is often used, alternative metrics may be better appropriate for specific data sets.

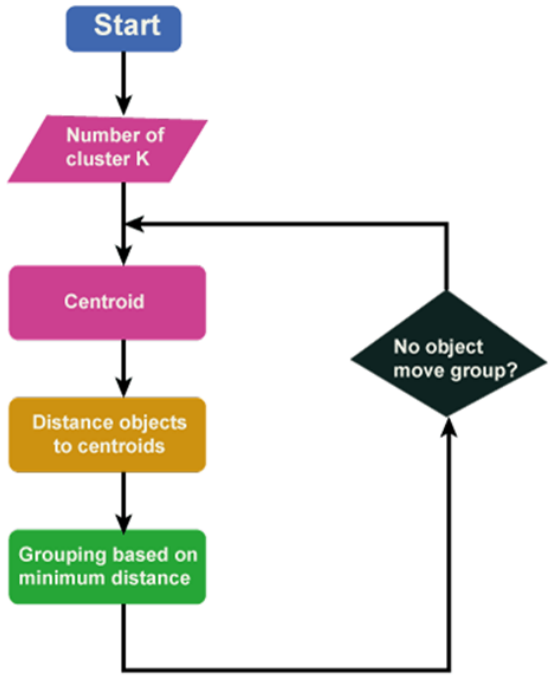

Figure 1 Shows the k-means algorithm architecture.

The k-means algorithm is a widely recognized clustering methodology used in the fields of machine learning and data analysis. The primary objective of this method is to divide a dataset into distinct groups or clusters, taking into consideration the similarity among its constituent pieces.

1.2.7 Objective Function

The primary goal of the k-means algorithm is to minimize the within-cluster sum of squares (WCSS), which is alternatively known as inertia or distortion. The Within-Cluster Sum of Squares (WCSS) metric computes the mean squared distance between each data point and its corresponding cluster centroid [8]. The measurement of this distance occurs within the confines of the same cluster. The subsequent expression represents the objective function:

$$WCSS = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left\| x_{ij-} c_i \right\|^2 \quad (1)$$

1.3 K-means++ Algorithm

The k-means++ algorithm has been developed as an enhancement to the basic k-means algorithm, with the objective of providing an upgraded initialization method for the cluster centroids [1]. The successful implementation of an initialization technique can lead to accelerated convergence and enhanced clustering results. What comes next is an analysis of the k-means++ algorithm's architectural framework and the steps that are taken in order.

1.3.1 Iterative Centroid Selection:

It is imperative to iterate the following methods until k centroids are obtained, with each centroid representing a distinct cluster.

1.3.2 Calculate Distances:

To determine the distances, it is necessary to calculate the distance between each data point in the dataset and the nearest centroid, which was selected in the preceding

phase. The Euclidean distance is commonly utilized as the distance metric in various applications. Nevertheless, alternative metrics can also be used for this purpose.

1.3.3 Probability Weight:

To apply the probability weight to each data point, it is recommended to use the square of the distance between the data point and its nearest centroid. The data points that are located at a greater distance from the existing centroids will have more likely weights.

1.3.4 Choosing Next Centroid:

The subsequent centroid is selected randomly from the data set, with a probability distribution that assigns greater weight to points with higher probability weights. This ensures that the centroids are distributed around the map and are sufficiently separated from each other.

1.3.5 Final step

After selecting k centroids using the aforementioned process, the next step is to proceed with the standard k-means algorithm by iteratively executing the assignment and update stages until convergence is achieved.

The way that the initial centers of the clusters are found is a big difference between the traditional k-means method and the k-means++ initialization strategy. The primary distinction that can be delineated between the two can be identified. Both methods proceed in a consistent manner thereafter, as the clustering algorithm maintains its fundamental structure once the initial centroids have been identified.

1.4 Elbow method:

The elbow technique is a heuristic approach utilized to determine the optimal number of clusters (k) within a clustering algorithm, particularly in the domain of k-

means++ clustering. Examining the correlation between the quantity of clusters and the within-cluster sum of squares (WCSS) is a reliable approach for ascertaining a suitable value for the parameter k. The following section illustrates the operational principles of the Elbow Method [9].

1.4.1 Choosing a range of values for k.

The first step is to select a range of values for the variable k. One must begin by choosing a specific interval value for the variable k that we wish to assess. In the usual practice, values of k are typically examined within the range of 1 to a predetermined upper bound. For instance, it is advisable to experiment with k values ranging from 1 to 10.

1.4.2 Computing the WCSS

The WCSS (within-cluster sum of squares) should be calculated after performing clustering for each specified value of k. Subsequently, the WCSS should be computed for each clustering outcome. The Within-Cluster Sum of Squares (WCSS) refers to the summation of the squared distances between every individual data point within a given cluster and its corresponding centroid. The metric measures the level of compactness exhibited by the clusters.

1.4.3 Plot the Elbow Curve

Generate a line plot or scatter plot where the independent variable, k, is represented on the x-axis and the dependent variable, WCSS values, are shown on the y-axis. It can be observed that with an increase in the number of clusters (k), there is a general drop in the within-cluster sum of squares (WCSS) due to the reduced distance between each data point and its own cluster centroid [10]. Nevertheless, as the number of

clusters increases, the enhancement in within-cluster sum of squares (WCSS) begins to reduce.

1.4.4 Finding the Elbow

The identification of the "elbow" is a crucial stage in the elbow method, wherein the plot is visually examined to locate the point of inflection known as the "elbow point." This is the junction on the graph where the decline in within-cluster sum of squares (WCSS) begins to exhibit an obvious deceleration. The plot commonly demonstrates a distinctive "elbow" shape, and the value of k corresponding to this elbow point is regarded as a suitable selection for the number of clusters.

1.4.5 Determine the optimal value for k.

The best number of clusters for a given dataset is frequently determined by selecting the value of k at the elbow point. Nevertheless, it is crucial to acknowledge that the elbow technique is a heuristic approach, and determining the appropriate value of k may still necessitate subjective judgment, taking into account the unique characteristics of the situation and domain expertise [11].

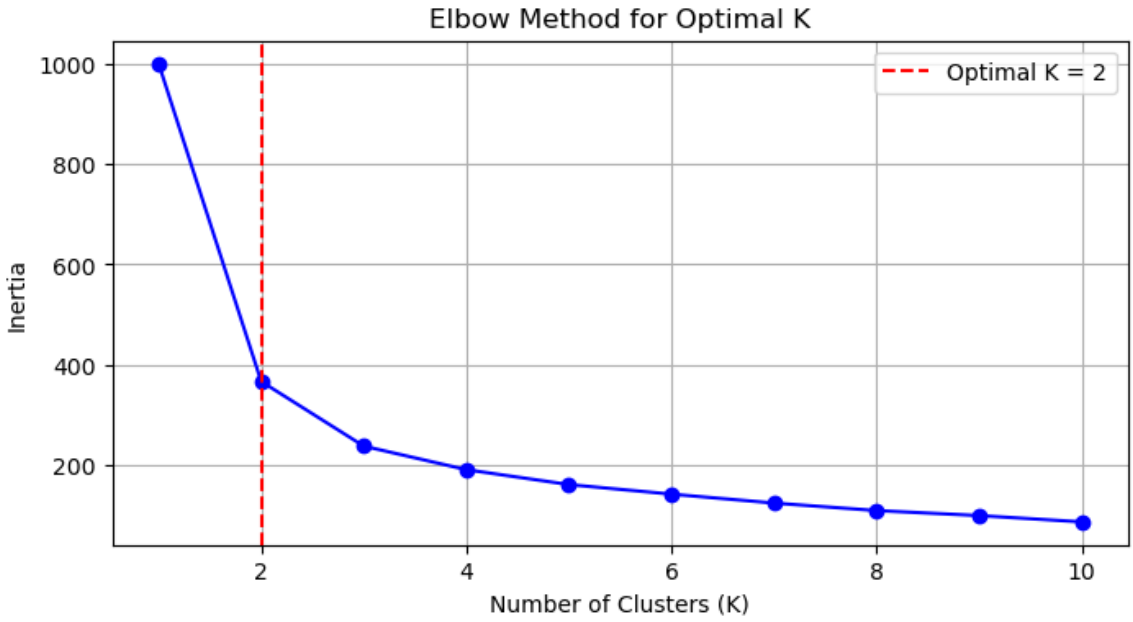

Figure 2 Elbow method for selecting the optimal number of clusters (k).

The elbow method offers a rapid and straightforward approach for approximating the optimal number of clusters in k-means and other clustering algorithms. The process aids in achieving harmony between the intricacy of the model and the effectiveness of clustering.

1.5 Anomaly detection in clustering.

Using clustering techniques to find anomalies means using clustering algorithms to find and separate outliers or instances that don't fit with the rest of the data [12]. Most of the time, the goal of clustering algorithms is to group data points that are similar. This leaves out data points that don't fit into any of the clusters or that are in clusters with few other data points. These are called anomalies. The methodology for conducting anomaly detection using clustering operates as follows:

The approach to performing anomaly detection with clustering works in the following manner [13].

1.5.1 Data Preprocessing:

- Data Cleaning: Start by cleaning and preprocessing the dataset. This step will involve handling missing values via scaling data features and transforming data if necessary.

1.5.2 Choose a Clustering Algorithm:

- Select an appropriate clustering algorithm based on the nature of the data and the objectives. We selected the common clustering algorithm k-means++.

1.5.3 Cluster the data:

- Apply the chosen clustering algorithm (K-means++) to the dataset to group data points into clusters. We used the elbow method to determine the optimal number of clusters.

1.5.4 Anomaly Detection:

- After clustering, we can consider data points that do not belong to any cluster as potential anomalies. These are the points that are far from all cluster centroids and do not fit well into any group.
- We can also identify anomalies within clusters by examining the density of data points within each cluster. Points that are far from the cluster center and are in sparsely populated clusters may be considered anomalies.
- Calculate a measure of dissimilarity or distance (Euclidean distance) between each data point and the cluster it does not belong to.

1.5.5 Thresholding:

- We will set a threshold for the dissimilarity measure. Data points with dissimilarity scores above the threshold are classified as anomalies.

1.5.6 Post-processing and visualization:

- We'll visualize the results by creating scatter plots or other visualizations that highlight the anomalies and their relationships with clusters.
- We can perform additional post-processing steps, such as removing duplicates or refining anomaly labels.

1.5.7 Validation:

- To evaluate the performance of anomaly detection, we need labeled data to confirm whether the identified anomalies are indeed anomalies.

1.5.8 Automated Anomaly Detection:

- For large datasets or real-time applications, we are looking forward to automating the anomaly detection process by periodically retraining the clustering model and reevaluating anomalies as new data arrives [14].

1.6 Scaling and standardization

Scaling and standardization are preprocessing procedures commonly used in machine learning and data analysis to convert numerical features or variables into a uniform scale or distribution. These strategies are crucial in order to provide equitable contributions of various features to the learning process, hence mitigating problems such

as feature dominance and facilitating improved convergence for algorithms that heavily rely on distance or magnitude-based calculations.

$$z = \frac{x-\mu}{\sigma} \tag{2}$$

1.6.1 Scaling:

Scaling is the process of transforming the values of numerical variables into a specific range, typically between 0 and 1 or -1 and 1. This ensures that all variables have the same influence on the model and prevents variables with larger scales from dominating those with smaller scales. Scaling doesn't change the distribution of the data, only its scale.

Common scaling methods include [15].

1.6.1.1 Min-Max Scaling: It scales the data to a specific range between 0, 1 by linearly mapping the minimum and maximum values of the variable to the desired range.

1.6.1.2 Z-Score Standardization: Also known as Standard Scaling, is a data transformation technique. After the transformation, the data are centered around a mean of zero and exhibit a standard deviation of one. This is particularly advantageous in scenarios where the data adheres to a Gaussian distribution.

The process of scaling holds significant significance for algorithms such as K-means++ and Support Vector Machines (SVMs), since they heavily depend on accurate distance estimates for their correct functioning [16].

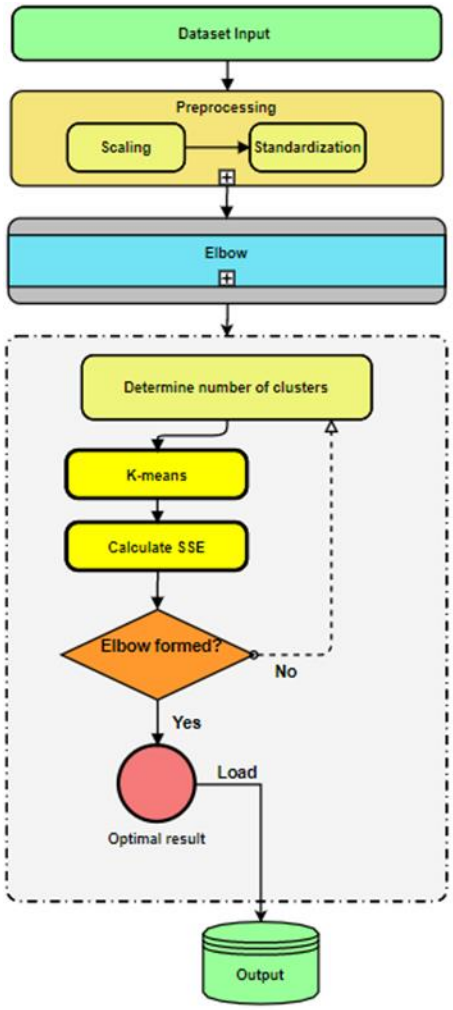

Figure 3. The process flowchart for the scaling and standardization steps

1.7 Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a statistical technique commonly employed in the domains of data analysis and machine learning. The main goal is to preserve trends and patterns while reducing the complexity of data with a high number of dimensions. The initial features, regarded as variables, undergo a transformation resulting in a distinct set of variables that are independent of each other. The new variables are known as principal components, which are obtained by linearly combining the original features.

1.7.1 PCA Objectives:

The utilization of dimensionality reduction techniques in data analysis and pattern identification is widely recognized [17]. This approach is effective in reducing the dimensions of datasets, thereby enabling algorithms to process the data more efficiently and with reduced complexity. Consequently, the running time of algorithms is improved. Additionally, dimensionality reduction methods ensure the preservation of data granularity and are recommended for efficient detection of data anomalies in machine learning applications. **Objective:** The main purpose of principle component analysis (PCA) is to perform a transformation on the variables of a dataset, resulting in a new form that exhibits greater sophistication. This approach will establish a linear lack of correlation among the variables, ultimately resulting in a reduction in the number of fractures within the dataset.

1.7.2 Steps:

- Standardization**:** Prior to conducting principal component analysis (PCA), it is crucial to do dataset scaling, as it ensures that all variables are brought within a standardized range of 0 to 1. This stage will be executed by computing the discrepancy between the average and each individual value in the dataset and subsequently dividing the resulting outcomes by the standard deviation. This procedure will standardize all datasets to a common range of values between 0 and 1 [18].
- Covariance Matrix: The initial phase of principal component analysis (PCA) involves the initialization of the CCM (Computation of Covariance Matrix). During this step, the standardized dataset will be utilized to provide a comprehensive representation of the variables contained within the normalized dataset.

- Eigenvalue Decomposition: The subsequent phase of this procedure will focus on the decomposition of the eigenvalue of the matrix, followed by the assignment of the resulting output to its corresponding value.
- Selecting Principal Components: The procedure will commence by allocating each value, and thereafter, the eigenvalues will be denoted as "k" once they have been arranged in decreasing order according to their magnitudes.

1.7.3 Creating New Features: During the ongoing evaluation process, it is possible for us to choose the suitable dataset to project the new dataset into axes.

1.7.4 Applications:

- Data Visualization: Data visualization is considered to be a significant application within the field of principal component analysis (PCA). To accomplish this task, it is necessary to restrict the variables inside the dataset, thereby guaranteeing the extraction of the most confidential information pertaining to the dataset. However, it is not advisable to utilize a data set with a high number of dimensions when performing data analysis.
- Noise Reduction: Noise reduction is an additional machine learning application that leverages principal component analysis. This application aims to reduce noise by reducing the dataset to its two most significant main components. This will allow the algorithm to focus more on the fine details of the data [19].
- Feature Engineering: Principal Component Analysis (PCA) is a valuable tool within the domain of data engineering, as it facilitates the exploration of datasets in novel and innovative ways. Additionally, PCA aids in the establishment of rules and restrictions for the effective utilization of data.

1.7.5 Limitations:

- Interpretability: There may be difficulties in elucidating certain variables that were initially chosen for inclusion in the dataset.
- Linearity: With respect to the linear and nonlinear interactions present among the variables within a given dataset, principal component analysis (PCA) proves to be a highly effective technique for analyzing linear datasets. However, it may not always be the optimal approach for handling datasets that exhibit nonlinearity and complexity.
- Information Loss: In the context of dimension reduction and translating the original dataset variables to a new dataset in Principal Component Analysis (PCA) format, there exists a certain cost associated with this process. However, it should be noted that this cost is relatively minimal. Hence, the ultimate consideration is to the number of components required for data reduction. In my scenario, the dataset was subjected to a mapping process, resulting in two principal component analyses (PCAs) [20]. Consequently, the percentage loss incurred amounts to 2%, with 1% attributed to each component.

1.7.6 PCA Equations:

- Covariance Matrix: In order to compute the covariance matrix for the dataset x, consisting of n samples, the following equation can be utilized:

$$Cov(X) = \frac{1}{n-1} \sum_{i=1}^{n} (xi - \overline{x})\,(xi - \overline{x})^{T} \quad (3)$$

- Eigenvalue Decomposition: The determination of eigenvectors and eigenvalues for the covariance matrix is an essential step in the use of principal component

analysis (PCA). The equation presented below demonstrates the eigenvalue decomposition of the covariance matrix.

$$Cov(X) = Q\Lambda Q^T \tag{4}$$

- Projection: The initial dataset can be mapped onto the selected principal components to effectively decrease the dimensionality of the data. The equation that governs the projection of each data point x_i onto the first k principal components can be expressed as follows:

$$x_{i}^{!} = x_k^T \; x_i \tag{5}$$

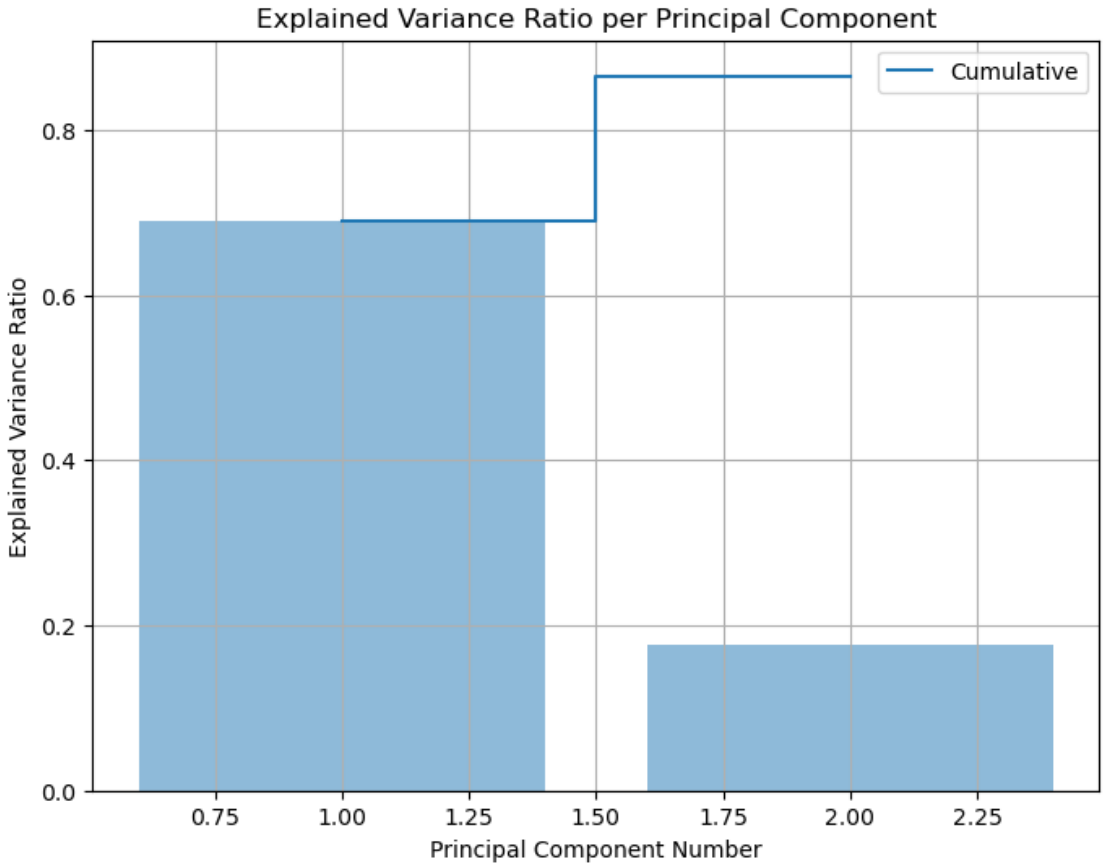

Figure. 4. Explained Variance Ratio per Principal Component

- Explained Variance Ratios:

The explained variance ratios refer to the proportions of the overall variation in the data that can be attributed to each principal component. In this particular case, we are addressing two main constituents. The initial principal component accounts for around 68.85% of the overall variance, whereas the subsequent principal component accounts for roughly 17.65% of the total variation.

- Cumulative Explained Variance:

The cumulative explained variance refers to the summation of deviations that have been accounted for up to a specific component [21]. The reason for this is that the cumulative explained variance serves as a cumulative metric. Upon an in-depth review of the initial component, it becomes evident that it accounts for approximately 68.85% of the overall variation independently. However, when both the first and second components are taken into consideration, their combined contribution amounts to roughly 86.50% of the total variation. This stands in contrast to the findings derived solely from the first component.

The PCA analysis plot indicates that, by considering the first two principal components, approximately 86.50% of the overall variance in the dataset may be retained. The availability of information regarding the reduction of dataset dimensionality through principal component analysis (PCA) is valuable as it facilitates comprehension of the extent to which information is preserved during this procedure. The determination of the number of core components to retain may be guided by the cumulative explained variance. However, this decision is contingent upon the objectives of the research and the balance between reducing dimensionality and preserving information.

CHAPTER TWO

LITERATURE REVIEW THESIS CONTRIBUTION

The K-Means++ algorithm is a highly regarded clustering technique renowned for its efficacy in partitioning data into distinct groups. Due to its efficacy, it has garnered significant popularity. This digital study extensively explores the K-Means++ approach, providing a comprehensive explanation of its mechanisms, the advantages it provides, and its significance in the domains of data analysis and machine learning.

2.1 Overview of the Algorithm:

This section provides an overview of the algorithm. K-Means++ is an enhanced version of the original K-Means clustering method, specifically designed to overcome many limitations inherent in the original technique. The mechanism employed by K-Means++ for initializing cluster centroids is a significant factor in its recognition as an innovative algorithm [1]. The approach employed involves the utilization of an intelligent initialization technique instead of randomly choosing centroids [24]. This approach yields notable improvements in both the rate of convergence and the general quality of the resulting clustering.

2.2 Process of initializing the K-Means++ model:

Select a random data point as the first centroid for the cluster. Compute the squared Euclidean distance (D(x)) between each of the remaining data points and its nearest existing centroid. The subsequent centroid for data representation can be determined by employing a probability distribution that is directly proportional to the

value for the data points. Continue to iterate steps 2 and 3 until a desired number of ‘k' centroids is achieved.

2.3 K-Means++ algorithm possesses advantages:

Enhanced Convergence The K-Means++ initialization method ensures that the initial cluster centroids are evenly distributed, hence mitigating the possibility of the algorithm being stuck in a suboptimal solution and enhancing its convergence capabilities. This finally leads to a more rapid convergence. Enhanced Clustering Accuracy: The K-Means++ algorithm exhibits superior cluster assignment accuracy by employing a sophisticated approach to centroid initialization, resulting in an overall enhancement of the clustering quality.

The K-Means++ algorithm is considered to be a more robust option for clustering in comparison to the standard K-Means algorithm due to its reduced sensitivity to the initial centroid configuration. Scalability: The K-Means++ technique exhibits scalability to large datasets and generally achieves convergence in fewer iterations compared to the original K-Means algorithm. Significance in Data Analysis: The significance of data analysis in academic research and decision-making processes cannot be overstated. The K-Means++ algorithm has exhibited its efficacy in diverse domains, including but not limited to consumer segmentation, image compression, document clustering, and recommendation systems. Due to its high degree of reliability and efficacy, this approach is commonly employed for unsupervised learning tasks.

2.4 Challenges and Considerations:

Despite the numerous advantages it offers, the K-Means++ algorithm does impose certain limitations and prerequisites that necessitate careful study. The underlying assumption of this hypothesis is that clusters exhibit spherical symmetry, have uniform sizes, and are isotropic in nature. Alternative clustering algorithms may be better suited for analyzing data sets that have irregularly shaped or overlapping clusters.

In summary, the K-Means++ algorithm is a valuable adaptation of the K-Means clustering algorithm that effectively tackles the challenges encountered during its initialization and substantially enhances the results of clustering. The technology in question is a versatile instrument commonly utilized by data scientists, facilitating the extraction of valuable insights from data in a more streamlined manner. The K-Means++ algorithm exhibits promising outcomes across several applications, contingent upon its careful implementation and integration with complementary clustering techniques.

Enhancing the Detection of Cybersecurity Threats with the Application of the K-Means++ Clustering Technique

Cybersecurity breaches targeting network infrastructures pose a progressively grave threat to individuals, enterprises, and even entire nations [23]. Due to the dynamic and evolving nature of these attacks, there is a pressing need for detection techniques that exhibit enhanced sophistication and efficacy. This thesis examines the utilization of K-Means++ clustering for enhancing cybersecurity threat detection. It also contributes by presenting a new methodology for identifying abnormal patterns in network traffic data. The selection of K-Means++ clustering was motivated by its capacity to enhance the identification of potential hazards.

2.5 Thesis Contribution:

The primary contribution of this thesis lies in the novel utilization of K-Means++ clustering for the identification and mitigation of cybersecurity threats. The subsequent enumeration outlines the fundamental contributions that this scientific endeavor has yielded: The engineering and selection of characteristics is a crucial component of the research, involving a comprehensive analysis of network traffic data to identify relevant properties. These qualities offer significant insights into the behavior of the network, hence facilitating the development of a dependable detection model.

The present study focuses on the initialization process of the K-Means++ algorithm for the purpose of anomaly detection. Conventional approaches for identifying anomalies generally depend on conventional statistical methodologies. The results of this study present a novel approach in which cluster centroids are initialized using the K-Means++ algorithm. The detection model exhibits enhanced capability in distinguishing between normal and abnormal activity subsequent to the application of the K-Means++ clustering technique to the network traffic data.

The thesis introduces a mechanism for dynamic clustering that is capable of adapting to changes in network activity over time. This approach employs K-Means++ reinitialization and retraining techniques to ensure the model's relevance in response to evolving threats. The fundamental goal of this project is to design a system for detecting threats in real-time. In the electrical power grid, the system demonstrates the capability to identify and address potential risks in real-time by leveraging the amalgamation of the K-Means++ clustering algorithm with streaming data processing techniques. This measure aids in mitigating potential system vulnerabilities, safeguards against critical

infrastructure, and conducts a comprehensive analysis of the proposed methodology's efficacy through the utilization of diverse datasets. A comparative analysis of the K-Means++-based approach and conventional methodologies for anomaly detection demonstrates the superiority of the K-Means++-based method in terms of both detection accuracy and false positive rate. Moreover, the proposed system addresses the issues of scalability and efficiency in relation to the proposed technique, ensuring its ability to handle large amounts of network data via PCA while maintaining detection accuracy. The concept of scalability pertains to the capacity of a technique to effectively manage the volume of data created within a network. Parallelization and optimization techniques are employed to achieve this level of scalability.

This thesis makes a significant contribution to the subject of cybersecurity threat identification by introducing a novel approach to address the issue. The proposed methodology utilizes the K-Means++ clustering algorithm to improve accuracy and enable real-time detection capabilities. The research findings possess the capacity to enhance the security of digital systems and networks, hence aiding in the mitigation of diverse forms of assault. The suggested technique has high effectiveness, scalability, and efficiency, making it a crucial tool in combating cybersecurity attacks.

2.6 Leveraging K-Means++ Clustering for Malware Detection

The increasing dependence on digital technology within electrical power networks has led to a notable concern regarding the prevalence of malware assaults. The reliance on technology has resulted in a pressing concern regarding the potential for malware attacks [24]. This research investigates the application of K-Means++ clustering in conjunction with the analysis of physical data obtained from power grid systems, with

the aim of enhancing malware detection. The research study introduces an innovative approach that adds value to safeguarding critical infrastructure by means of its discoveries.

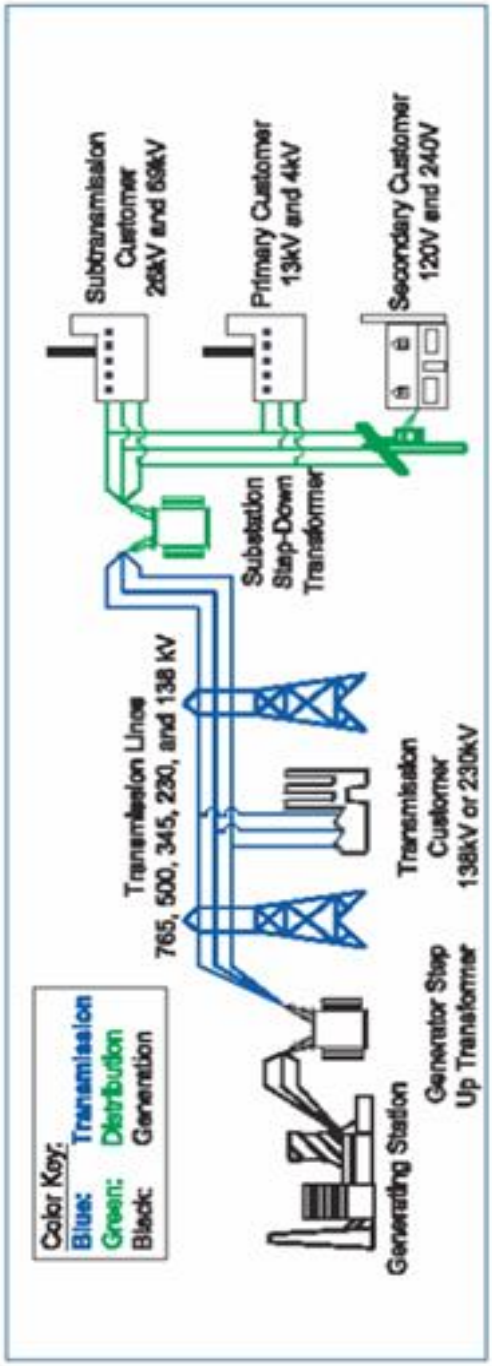

Figure. 5. Electrical Power Grid

2.6.1 Data Collection and Feature Extraction:

The process of gathering empirical evidence for the thesis from a diverse range of sensors and devices collected via Internet of Things (IoT) devices is given by the Kaggle platform, which will be introduced in the next chapter [26]. The aforementioned data is further examined in order to ascertain pertinent attributes, including voltage, current, frequency, and phase angle. Understanding the fundamental attributes outlined above is crucial for gaining insight into the standard functioning of the power grid.

2.6.2 Anomalous Behavior Detection:

The concept of anomalous behavior detection refers to the identification and analysis of atypical patterns or actions within a given system or dataset. The K-Means++ algorithm is a technique used in clustering analysis [27]. Clustering is employed as a means to depict the typical behavior exhibited by the electrical power system. This enables the identification of any atypical conduct. The utilization of the clustering methodology proves advantageous in identifying recurrent patterns and characteristics that are associated with grid activity. Any activity that diverges from these established patterns can be construed as an abnormality, perhaps suggesting the existence of malicious software.

2.6.3 Dynamic Model Update:

The study presents a novel approach for dynamic model updating, wherein the K-Means++ clustering model is adapted over time to accommodate variations in the power grid's behavior [28]. The achievement was attained by employing an evolutionary

algorithm. This ensures that the detection system will maintain its efficiency despite the ongoing evolution of techniques adopted by malware.

2.6.4 Feature Engineering for Malware Signatures:

The Development of Feature Engineering Techniques for Application in Malware Signatures: The objective of this thesis is to explore the feasibility of creating feature sets that can effectively capture established malware signatures inside physical data, with the aim of enhancing the system's capacity to identify and detect malicious activities. The incorporation of the following features into the clustering model has been undertaken with the aim of enhancing the detection capabilities for specific types of malware programs.

2.6.5 Case Studies and Performance Evaluation:

The topic of interest pertains to case studies and performance evaluation. The proposed methodology is comprehensively scrutinized by utilizing empirical data derived from operational power grids alongside data obtained from simulated instances of malware assaults. The results demonstrate the system's capacity to effectively identify anomalies related to malware and highlight the practicality of employing K-Means++ clustering in this specific context [29].

2.6.6 Real-time alerting and response:

The objective of this study is to develop a real-time alerting system that can promptly inform grid operators about potential malware incidents as they occur. As a result, the prompt implementation of swift responses can mitigate the impact of malware attacks on the operational efficiency and stability of the power system.

In conclusion, it can be inferred that this thesis contributes to the field of cybersecurity in critical infrastructure by introducing a novel approach that utilizes K-Means++ clustering to detect malware in electrical power grids. The utilization of physic's data analysis is a viable approach in the identification of anomalies and potential assaults that may evade detection through conventional cybersecurity methods. The reason for this is because conventional cybersecurity protocols prioritize the safeguarding of the logical elements within a network. The study provides evidence of the practicality and effectiveness of the suggested system, emphasizing its capacity to enhance the safety and resilience of electrical power grids in response to rising cybersecurity risks.

CHAPTER THREE

KAGGLE PLATFORM

Kaggle is a widely recognized online platform that provides opportunities for anyone with a keen interest in the fields of data science and machine learning. There exists an online community of data scientists and engineers who are engaged in the field of machine learning [26]. The platform was initially established in 2010, and over time, it has evolved into a prominent hub for holding data science competitions, facilitating the exploration of data, and fostering the creation of machine learning applications. Kaggle has played a substantial role in fostering the expansion of the data science and machine learning areas. This is achieved through the provision of a platform that facilitates collaboration between individuals at various skill levels, enabling them to collectively engage in problem-solving activities related to practical data science scenarios. Additionally, Kaggle serves as a platform for knowledge exchange, allowing participants to share their expertise and acquire new insights from their peers. This resource should not be disregarded by those with even the slightest interest in these domains.

The repository of datasets accessible on Kaggle encompasses an extensive variety of subject domains. The datasets are made available in downloadable formats to facilitate their utilization in applications pertaining to data science, machine learning, and data analysis.

Participation in Kaggle entails engaging in contests within the realm of data science, wherein corporations and organizations provide datasets and pose problem statements. Participants on Kaggle's platform have the opportunity to engage in these

competitions. These problems serve as a catalyst for global collaboration among data scientists, machine learning experts, and practitioners, fostering the joint creation of models and algorithms aimed at effectively addressing the identified problem. The ability to participate in data science competitions presents an exceptional opportunity to demonstrate and highlight one's proficiency in the field, with the added incentive of potentially securing monetary rewards.

Kaggle curates a substantial repository of datasets that are openly available to the general public, encompassing diverse fields such as the social sciences, economics, and healthcare. Users possess the capability to publish and disseminate datasets for the purpose of facilitating use by other users. One may engage in an exploration of the Kaggle Datasets website in order to enhance their understanding of Kaggle's comprehensive collection of datasets. Datasets can be located through the utilization of keywords or by navigating through different categories and tags, enabling the identification of datasets that align with our specific interests or project prerequisites.

3.1 Essential features of Kaggle:

Kaggle is a comprehensive platform catering to data science amateurs, professionals, and scholars. It cultivates a cooperative and competitive environment that promotes learning and problem-solving in the domain of data science and machine learning. The combination of Kaggle's attributes renders it a comprehensive platform.

3.1.1 Participation

Kernels refer to an interactive computing environment known as Jupyter Notebooks, which facilitates the execution of Python or R applications directly within web browsers. This tool proves to be highly advantageous in conducting data analysis and

constructing models. The act of exchanging kernels with other individuals is indeed a feasible endeavor.

3.1.2 Training and Education

Kaggle provides a variety of courses and tutorials aimed at facilitating users' acquisition of knowledge in the fields of data science and machine learning, as well as enhancing their proficiency in these domains. This course encompasses a range of topics, including Python programming, machine learning, deep learning, and various other areas.

3.1.3 Engagement and discourse

Kaggle offers its members a platform for community interaction, enabling them to inquire, deliberate on data science subjects, and collaborate on initiatives pertaining to the domain. This venue provides an excellent setting for discussing the field of data science and seeking support for any challenges encountered.

3.1.4 Prospects for Securing Employment

Pathways for Career Advancement Kaggle serves as a valuable platform for cultivating one's data science portfolio, hence facilitating access to diverse avenues for career progression. The assessment of candidates' talents and level of experience by employers in the domains of data science and machine learning sometimes involves the examination of their Kaggle profiles.

3.1.5 Datasets and Notebooks

Kaggle offers users a platform for publishing and sharing datasets, along with Jupyter Notebooks that demonstrate data analysis and machine learning projects. The utilization of Kaggle's collaborative and sharing capabilities facilitates the acquisition and

dissemination of information among community members, hence emphasizing its significance.

3.1.6 Downloading Datasets

Kaggle provides users with the capability to access and get datasets directly from their personal accounts. Most datasets are available for download in widely used file formats for data, such as CSV, JSON, or Excel. In order to access datasets, it is typically necessary to possess a Kaggle account, although a considerable number of datasets are available at no cost. To ascertain any usage constraints or licensing agreements associated with certain datasets, it is advisable to refer to the metadata of the dataset [30].

3.1.7 Dataset Metadata

The dataset metadata for each entry in Kaggle's database includes information that describes its contents, such as the source of the data, its definition, the number of rows and columns, the document type, and any relevant license information. The pertinent information can be located within the "Dataset Metadata" section of the Kaggle website. The acquisition of this information is crucial for comprehending the whole content of the dataset and determining its potential applications.

3.1.8 Apps and APIs

Kaggle offers a range of applications and application programming interfaces (APIs) that facilitate the programmatic access of its data and services by developers and data scientists. This development facilitates enhanced automation and enables increased interaction with many platforms and tools.

Figure 6: Distributed Transformer Monitoring Dataset

3.2 Advantages of Kaggle

Kaggle offers a multitude of benefits to individuals who possess proficiency in machine learning and data analysis and those with a vested interest in subjects pertaining to data analysis.

3.2.1 Provision of Access to a Wide Array of Datasets

Kaggle offers users the opportunity to access a substantial collection of datasets that are categorized based on a diverse range of subject domains. Due to the diverse array of difficulties present in the real world, this program offers the chance to engage with and evaluate data from several disciplines, such as healthcare, economics, and the social sciences, among others.

3.2.2 Data science competitions

Data science competitions have emerged as a popular platform for individuals and teams to showcase their skills and expertise in the field of data science. These competitions provide a competitive environment. Kaggle serves as the platform for several data science challenges, a subset of which provide financial rewards to the winners. By engaging in these challenges, participants will have the opportunity to demonstrate their talents, tackle complex tasks, and engage in competition with data scientists globally.

3.2.3 Kaggle Kernels

The Kaggle Kernels feature provides users with the ability to create and execute code within a Jupyter Notebook environment, accessible through their web browsers. The utilization of data analysis and machine learning facilitates the efficient creation and

distribution of projects. Utilizing kernels generated by other individuals presents an additional avenue for acquiring knowledge and engaging in collaborative endeavors.

3.2.4 Collaborative Efforts in Open-Source Development

Kaggle promotes collaborative efforts among users by facilitating the utilization of open-source tools. The community has undergone a transformation into an interactive platform for the sharing of information and code, owing to the substantial contributions of kernels and datasets.

3.2.5 The Concept of Awareness and Recognition

Achieving favorable outcomes in Kaggle competitions and developing frequently utilized projects might contribute to increased visibility and acknowledgement within the data science industry. This opportunity presents a platform for you to showcase your proficiencies and specialized knowledge, so it is advisable to capitalize on it. The topic of interest pertains to the concept of problem solving and the various challenges associated with it. Kaggle's competitions encompass a diverse array of problem domains, spanning from image classification to natural language processing. Enhancing problem-solving abilities and staying updated on contemporary business strategy.

3.3 Kaggle Constraints and Restrictions

Individuals with an inclination towards data analysis and related subjects can get considerable advantages by employing Kaggle, a platform that offers diverse avenues for exploration. This includes professionals specializing in data analysis, individuals well-versed in machine learning techniques, and any other individuals with a keen interest in data analysis and its associated themes. Nevertheless, it is important for Kaggle users to

acknowledge some constraints and restrictions, notwithstanding the platform's multitude of favorable qualities and advantages.

3.3.1 Internet Dependency

The necessity of a reliable internet connection is a prerequisite for users to access the functionalities offered by Kaggle, given its online nature. Utilizing Kaggle efficiently can pose challenges for individuals with little or no access to internet connectivity.

3.3.2 Data Privacy and Sensitivity

Kaggle datasets are regularly provided to a wide range of users; yet, it is important to note that not all datasets are universally applicable to every use case. Given that Kaggle is a publicly available website, it is advisable for users to use caution while handling sensitive or confidential data, as datasets posted on the platform may be visible to other individuals.

3.3.3 Limited Hardware Resources

Although Kaggle offers free GPU and CPU resources for data analysis and machine learning, these tools are concurrently utilized by a substantial user base. There exists a potential for a scarcity of resources during instances of popular challenges or periods characterized by high levels of consumption.

3.3.4 Niche Target Audience

The primary focus of Kaggle is centered around fields such as data science, machine learning, and other closely related disciplines. If one's primary interest lies in alternative subfields of computer science or domains, such as web development, it is advisable to consider alternative platforms rather than Kaggle.

3.3.5 Platform Modifications

Both the services provided by Kaggle, and the governing rules are susceptible to modification. It is advisable for users to stay updated on platform updates and develop a strategic approach to address any potential changes.

3.3.6 Public data overfitting

The dataset provided to the public in Kaggle contests generally constitutes a partial representation of the complete dataset. If the model is excessively tailored to fit the public data, there is a potential drawback of insufficient generalization to the private test data, which will only be accessible after the competition concludes.

3.4 Kaggle Learning:

Students have the opportunity to acquire and enhance their proficiency in data science and machine learning through the use of several resources accessible on Kaggle. Kaggle provides a diverse range of educational opportunities suitable for individuals at various levels of experience, whether they are beginners seeking to enter the sector or experienced professionals aiming to enhance their knowledge and skills. Figure 6 shows the variety of opportunities that Kaggle encompasses.

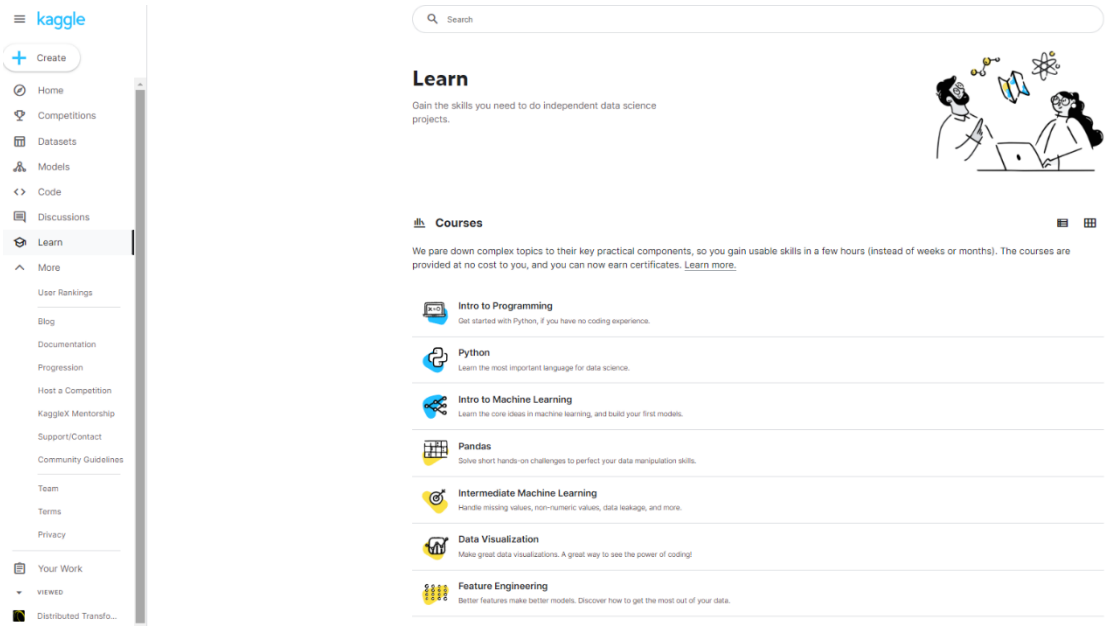

Figure 7. Kaggle Learning hub

In order to optimize the utilization of Kaggle's educational resources, it is advisable to consider establishing a Kaggle account and engaging with the diverse array of courses, datasets, and community functionalities accessible on the platform. In order to optimize one's learning experience on Kaggle, active participation in tournaments and engaging in discussions with fellow community members are recommended strategies. Kaggle is a highly interactive platform that possesses the potential to serve as a valuable tool for the professional advancement and ongoing education of individuals in the field of data science.

3.5 Kaggle in the academic and research domain

Kaggle is an indispensable resource in the domains of research and education, particularly for individuals engaged in data-driven disciplines such as computer science, statistics, and machine learning. In the context of research publications, it is imperative to

comply with ethical principles, acquire appropriate permits for data usage, and accurately cite the source while utilizing Kaggle datasets. Kaggle datasets have the potential to serve as a valuable resource for academic and research endeavors. Nevertheless, it is imperative to exercise caution and prudence when utilizing them. Furthermore, it is imperative for researchers engaging in Kaggle contests for academic purposes to familiarize themselves with the competition regulations and license agreements prior to their participation in any events. Below are many examples that illustrate the utilization of Kaggle in academic and research environments:

3.5.1 Research Data

Kaggle offers academics a wide range of datasets that can be highly valuable for their different research endeavors. The datasets accessible on Kaggle offer valuable opportunities for research and analysis across several sectors, encompassing the social sciences, healthcare, and numerous other domains. Engaging in Kaggle competitions and projects provides academics with the chance to acquire hands-on experience in tackling data-driven challenges and enhancing machine learning models. When conducting research in either the business or academic domains, this knowledge may be of significant value.

3.5.2 Incorporation of real-world problems

Characterized by extensive data sets, for the purpose of benchmarking is a common practice in Kaggle competitions. Competitions of this nature possess the capacity to function as benchmarks for the algorithms and models devised by scholars. The evaluation and enhancement of research methodologies can be achieved by leveraging Kaggle tournaments. Kaggle is a renowned and extensively utilized

instructional resource within the realm of academia. Students have the opportunity to acquire practical experience in the fields of data science and machine learning through enrollment in courses that utilize Kaggle tournaments, datasets, and kernels as instructional resources.

3.5.3 Ethical Utilization of Data

The requisite information for the dissemination of study findings can be accessed on the Kaggle platform. Researchers must use caution and diligence when retrieving datasets from Kaggle, as it is imperative to ascertain the data's origin in order to uphold appropriate citation practices and ensure ethical utilization of the data.

3.5.4 The Investigation of Novel Concepts

Academic researchers can leverage Kaggle contests as a means to explore novel ideas and methodologies for problem-solving within their respective domains through active engagement in these competitive events. It is conceivable that players engaged in Kaggle tournaments may generate novel solutions to the assigned problems.

3.5.5 Ensuring Validity and Reproducibility

Kaggle is a platform that affords researchers the option to validate and replicate the work of other people. Academic researchers possess the capability to scrutinize Kaggle kernels and competition solutions for the purpose of assessing the efficacy of previously formulated methods.

3.5.6 Promoting Interdisciplinary Collaboration

Kaggle promotes interdisciplinary collaboration by virtue of its comprehensive dataset collection. Interdisciplinary research endeavors facilitate collaboration among scholars from diverse disciplines in order to collectively tackle a broad spectrum of intricate and distinctive challenges.

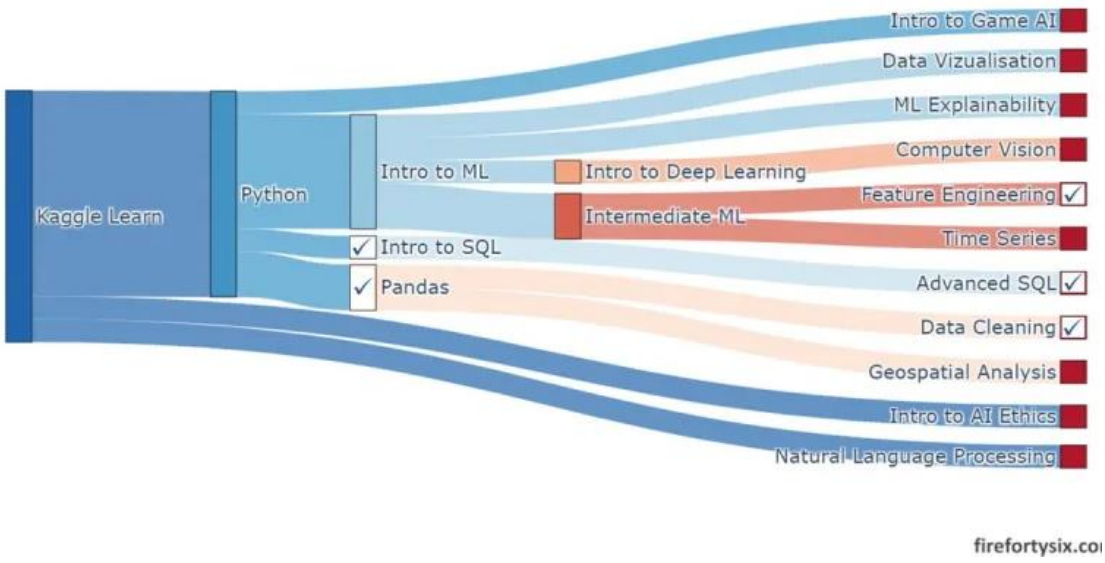

Figure 8. Kaggle Collaboration in Data Since and Deep Learning

3.6 The mythology of Kaggle

The platform utilized by Kaggle incorporates a diverse array of technologies and construction methodologies. Although specific frameworks and technologies may see modifications over time, Kaggle has been observed to utilize the following technologies.

3.6.1 Python

Python is the predominant programming language utilized on Kaggle, exhibiting the highest frequency of usage. The utilization of Kaggle, a platform that offers users a

Python environment for coding and executing operations in Jupyter Notebooks, is a prevalent approach within the domains of data science and machine learning.

3.6.2 The Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing environment that facilitates the sharing of documents that contain executable code, equations, and visual Kaggle kernels. These kernels, which are designed based on Jupyter Notebooks and facilitate users in developing, executing, and exchanging code, constitute a fundamental component of the Kaggle platform.

3.6.3 GPU'S and CPU'S

Kaggle offers computational resources for both graphics processing units (GPUs) and central processing units (CPUs), facilitating enhanced efficiency in training machine learning models and performing data analytic tasks. Nvidia GPUs are employed for a range of machine learning projects.

3.6.4 API

The Kaggle platform provides users with the capability to interact with it programmatically through the utilization of an application programming interface (API). This application programming interface facilitates a diverse array of functionalities, encompassing tasks such as retrieving datasets and submitting entries to competitions, among other capabilities.

3.6.5 Cloud Services

Cloud services refer to the delivery of computing resources, such as storage, processing power, and software applications, over the internet. Kaggle employs a cloud

computing infrastructure, such as Google Cloud Services, to facilitate the hosting of its platform and provide users with diverse computer resources.

3.6.6 GitHub

Kaggle users have the option to establish a connection between their accounts and GitHub, thereby facilitating the seamless sharing of code and projects between these two platforms. The capability is accessible to Kaggle users.

3.6.7 SQL Databases

Kaggle utilizes SQL databases for the purpose of storing metadata as well as additional information pertaining to datasets and user activity on the platform.

3.6.8 Privacy Measures

Kaggle implements various privacy and security measures to preserve the data of its users and enforce compliance with its data usage policies. Kaggle is an online competition platform that mostly centers around the domains of data science, machine learning, and data analysis. This platform offers a wide range of valuable tools and chances for individuals at various levels of expertise, including both newbies and those with more experience in the industry. The utilization of this platform is prevalent within academic, research, and industrial settings, serving as a foundation for machine learning, data science, and other domains. Moreover, this platform has a progressive nature, thereby making a significant contribution to the advancement of the data science community.

CHAPTER FOUR

POSSIBLE ATTACKERS AND INFRASTRUTURE

Infrastructure security pertains to the implementation of comprehensive procedures aimed at safeguarding important systems and assets from potential threats, including both physical and digital attacks. From an information technology perspective, the scope of asset management typically encompasses both software and hardware resources, including personal computers, smartphones, tablets, data center infrastructure, networking systems, and cloud-based resources [31]. The preservation of an organization's assets, electronic records, reputation, adherence to regulations, and sustainability heavily relies on the criticality of infrastructure security.

In contemporary times, it is feasible for an individual to effortlessly transmit and receive diverse forms of data, encompassing electronic mail, videos, and audio files, by a simple act of pressing a button. However, it remains pertinent to inquire whether any contemplation has been given to the users' input on the degree of security in the transfer and communication of their information to ensure its integrity and confidentiality. The remedy can be found in the realm of cyber security. The Internet has emerged as a rapidly expanding infrastructure in contemporary daily life. [32].

4.1 Cyber-attacks on Critical Infrastructure

The energy "grid" consists of critical infrastructure systems such as those that power generation, treatment of water, electricity production, and other platforms. This grid, while useful to the public, is vulnerable to cyberattacks by "hacktivists" or terrorists.

According to Idan Udi Edry, the Chief Executive Officer of Nation-E, a company specializing in cyber security solutions, their services facilitate secure connections between customers' infrastructure and the internet. This allows for remote access and control of vital assets in a safe manner.

The energy sector is a prominent target for cyberattacks on important networks; however, it is not the only industry susceptible to such threats. Transportation, government services, telecommunications, and vital sectors such as manufacturing are also susceptible to potential risks.

4.1.1 Complexity of critical infrastructure

The complexity and interconnectivity of critical infrastructure, such as power generation and distribution systems, are steadily increasing. A few decades ago, power grids and other essential infrastructure systems functioned in a state of isolation. Currently, there exists a significantly higher level of interconnectivity, encompassing both geographical and cross-sectoral dimensions. [33]

According to Edry, the US power grid scenario serves as a notable example, illustrating how the breakdown of a single essential infrastructure component might potentially trigger a catastrophic sequence of events. The issue about the susceptibility of vital infrastructure to cyberattacks and technical failures is not surprising. Recent incidents have provided validation for the concerns expressed. According to Internet Crime Complaint Center, there was a notable rise of 20% in cyber investigations throughout the year 2015, accompanied by a significant surge in attacks targeting important manufacturing sectors in the United States, which doubled in frequency.

4.1.2 Infrastructure Cyberattacks Target Control Systems

According to the Organization of American States, it is more likely for cyberattacks on critical infrastructure and manufacturing to focus on industrial control systems rather than data breaches [31]. According to their study, it was discovered that 54% of the 500 suppliers of essential services in the United States who were polled reported instances of attempts to manipulate their systems, while 40% encountered efforts to disrupt the functioning of their systems. Moreover, 50% of respondents indicated that they had observed a spike in attacks, whereas 75% expressed the belief that these attacks were getting increasingly sophisticated.

Table I. Recent Cyber Attacks on Critical Infrastructures

Attack/Year	Attacker	Attacked	Type of Attack
Ukraine Power Grid Attack No. 2 (2016)	Russia	(Ukraine Power Grid)	Same as Ukraine Power Grid Attack No. 1 (2015) but with a TDos/More organized than first
TRITON on Saudi Oil and Gas facility (2017)	XENOTIME	Saudi Arabian Oil and Gas facility	TRISIS malware framework [1]/Error in attack-no damage
Symantec Dragonfly attacks (2017)	Dragonfly	US and EU Energy Sector	Range of malware tools. RAT software. [2]/Raised awareness
Cryptocurrency Mining at EU wastewater site (2018)	Unknown	EU wastewater site and networks	Mining Malware [4]/System was down for an unknown amount of time
Firewall Flaw in western US Grid (2019)	Novice hacker Running a script	Grid sites in California	Automated bot scanning for vulnerable networks" [5] /Grid was blind for 10 hours
Colonial Pipeline Hack (2021)	DarkSide	Colonial Pipeline	Brute Force Attack/Ransomware. [6] Pipeline was shut down for six days

As previously said, cyberattacks are a subject of significant concern. Despite the abundance of warnings, reports, and events, threat actors persistently engage in the development of attacks. The frequency and sophistication of attacks perpetrated by threat actors have been observed to escalate annually [31][32]. Table I displays instances of historical and contemporary cyberattacks that have occurred between 2016 and 2021. This table provides details on the perpetrator responsible for each attack, the specific

infrastructures that were targeted, the sort of attack used, and any discernible consequences resulting from these incidents. The Western US grid attack involved a more sophisticated strategy, wherein the attackers used an automated bot to systematically search for networks with vulnerabilities [32].

Due to the enhanced techniques and tools utilized by threat actors, it is imperative to adapt security protocols and strategies accordingly. The utilization of proactive digital forensics (ProDF) can assist in addressing these sophisticated approaches. Before the occurrence of assaults, investigators must assure the appropriate implementation of plans, tools, and methodologies, to enhance efficiency in terms of time, effectiveness, and cost.

4.1.3 Infrastructure for enterprises, organizations.

Based on the existing national legislation and international legal agreements approved by Ukraine, the category of critical infrastructure items encompasses firms, institutions, and organizations that hold significant strategic importance for the state, regardless of their ownership structure.

In response to the growing menace of cybercrime, there has been a significant focus on digital forensic investigations to address critical infrastructure incidents. Currently, investigations heavily depend on reactive digital forensics, which involves conducting inquiries after the occurrence of an incident [34]. The investigation process is characterized by its significant consumption of time and financial resources. Additionally, this technique carries the potential for data loss. Following the occurrence of an incident, it is imperative to promptly obtain data before the system replaces volatile data. Incorporating a proactive methodology in digital forensic investigations has the potential

to yield time and cost savings while simultaneously bolstering the system's protection mechanisms. [102]

4.1.4 Challenges and Techniques in Cyber Security

Over the past few years, there has been a significant change in the characteristics of IT security incidents. Previously, these incidents primarily consisted of isolated attacks on information systems. However, there has been a discernible transformation towards deliberate, focused, and intricate cyber threats that have been targeted towards individuals, institutions, and even entire nations [33]. The interdependent functionalities of digital technologies afford several advantages, yet they also generate a plethora of novel risks that have significant implications. Despite the apparent interchangeability of these phrases, it is important to note that cybersecurity and information security are distinct concepts.

4.1.4.1 Access control and password security

Usernames, passwords, PINs, biometric scans, and security tokens have always been key components of information security. This could be one of the initial cybersecurity measures.

4.1.4.2 Authentication of data

Before downloading any attachment that we receive via email or any other manner, it must always be validated, meaning that all documents have to be examined to ensure that they are not coming from a untrusted source [35]. Antivirus software is used to authenticate these documents.

4.1.4.3 Malware detectors

Usually, malware detectors check all the system's files and documents for harmful viruses or malicious code, where malicious software, which is sometimes referred to as malware, includes viruses, worms, and Trojan horses.

4.1.4.4 Anti-virus software

Antivirus software, usually referred to as AV software, which is a computer program intend to identify and find harmful software's from the PC or network. Where is malware is any software that can damage the computer or corrupt or manipulate data, which includes viruses, worms, Trojan horses, spyware, adware, and other similar programs.

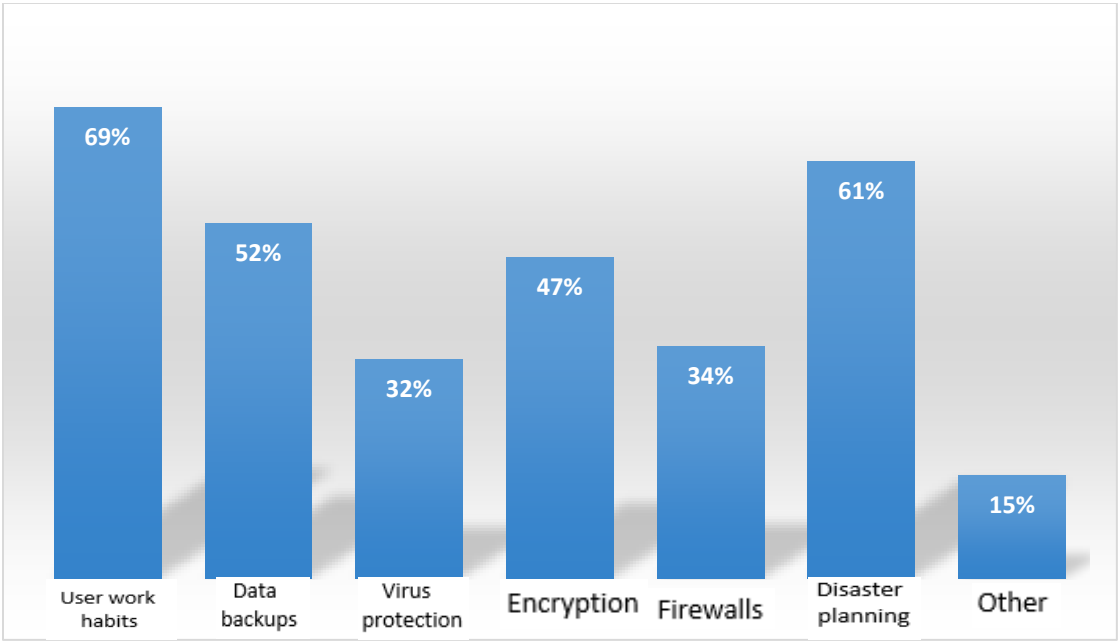

Figure. 9. Cybersecurity techniques

4.2 The Phenomenon of Cybercrime on Computer Networks

Any illegal activity that might use a computer as its main tool for commission and theft is referred to as cybercrime [37]. The U.S. Department of Justice broadens the definition of cybercrime to encompass any illegal action that keeps evidence on a computer. According to the Internet Crime Complaint Center (IC3), an average of 652,000 complaints have been reported to the IC3 over the course of the previous five years. The grievances encompass a wide array of internet fraudulent activities that affect individuals worldwide [IC3].

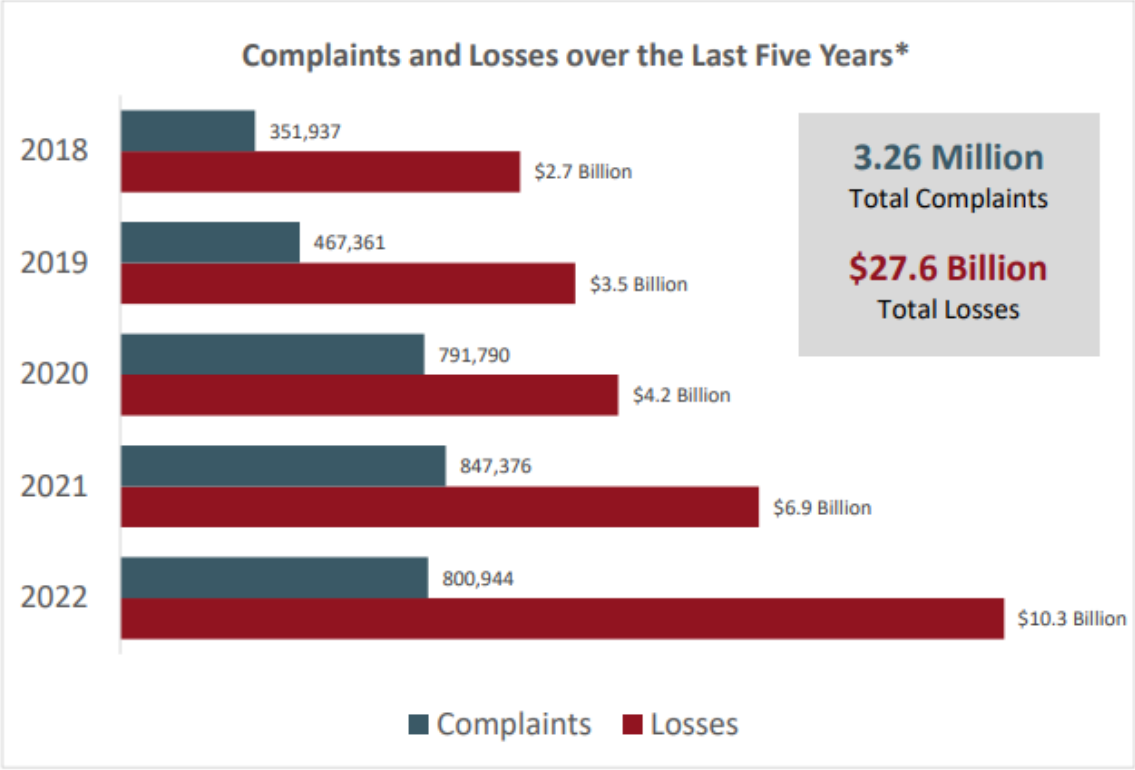

Figure. 10. Data for complaints and losses over the years 2018 to 2022. [36]

The chart presents both annual and cumulative data related to complaints and losses from the years 2018 through 2022. During the specified period, the IC3

accumulated a cumulative count of 3.26 million complaints, which corresponded to a reported financial loss amounting to $27.6 billion. The user's text is already academic in nature.

The data that is shared with IC3 can have a significant impact on individual complaints, but its greatest impact is observed when analyzed collectively. When these individual complaints are aggregated with additional data, it enables the FBI to establish connections between complaints, conduct investigations into alleged crimes, monitor patterns and potential threats, and, in certain instances, impede the flow of misappropriated monies. Equally significant is the fact that the IC3 disseminates crime reports across its extensive network of FBI field offices and law enforcement collaborators, thereby enhancing our country's unified reaction at both local and national levels.

4.3 Strategies for Mitigating Critical Infrastructure Cyberattacks

It is important to promote the development and maintenance of a cybersecurity-oriented culture. Ultimately, it is human beings who safeguard the integrity and security of businesses. The compromise of a system can occur through several means, such as phishing and zero-day attacks. In some instances, the system's security is breached due to the actions of a lone employee, who may inadvertently download a file containing malware, mistakenly disclose their credentials to a cybercriminal, or neglect to apply necessary patches or updates to their equipment [36]. The strength of the system is only determined by the weakness of the weakest password, as it can be compromised using brute force assaults and password spraying techniques.

4.3.1 Cyber Hygiene.

Multi-Factor Authentication. Employees require more than a password to enter a network or system.

Data Encryption. Smart grids, smart meters, and other IoT devices should encrypt data and communication.

Firewall. A digital firewall that scans, assesses, and filters incoming traffic.

Security Information and Event Management (SIEM). Protects against malware and monitors network access.

Anti-malware software. Scans devices, finds risks, and eliminates malware.

4.3.2 Securing physical assets:

Physical asset security is essential for individual, company, and organizational security. Infrastructure, equipment, data centers, and other physical infrastructure must be protected from numerous threats and risks. Most common approach has been used in companies and organizations which is access cards, biometrics, and surveillance cameras, etc. set stringent access controls and allow only approved users to access the software. A software that has been used widely within organizations called is called lightweight directory access protocol (LDAP), with which organization can store data in the company local directory and authenticate certain users to access the directory.

Figure. 11 Lightweight directory access protocol [38].

CHAPTER FIVE

DECOY

In the world of computer security, a decoy refers to an artificial system or network that is strategically established with the intention of attracting and misleading malicious hackers. Referred to as a "honeypot," this simulated system is strategically developed to redirect and ensnare malicious actors in the cyber realm, thereby enabling cybersecurity professionals to actively observe their actions, accumulate valuable intelligence, and safeguard the authentic network infrastructure.

Honeypots have gained widespread acceptance and have been extensively studied as a method for monitoring the occurrence of vulnerabilities across various internet services. Researchers are enabled to identify the emergence of novel exploits and monitor the extent to which vulnerabilities are being exploited. Adversaries possess a vested interest in identifying and evading honeypots, as they aim to refrain from attracting scrutiny towards their distinct methods of exploitation and associated resources [39].

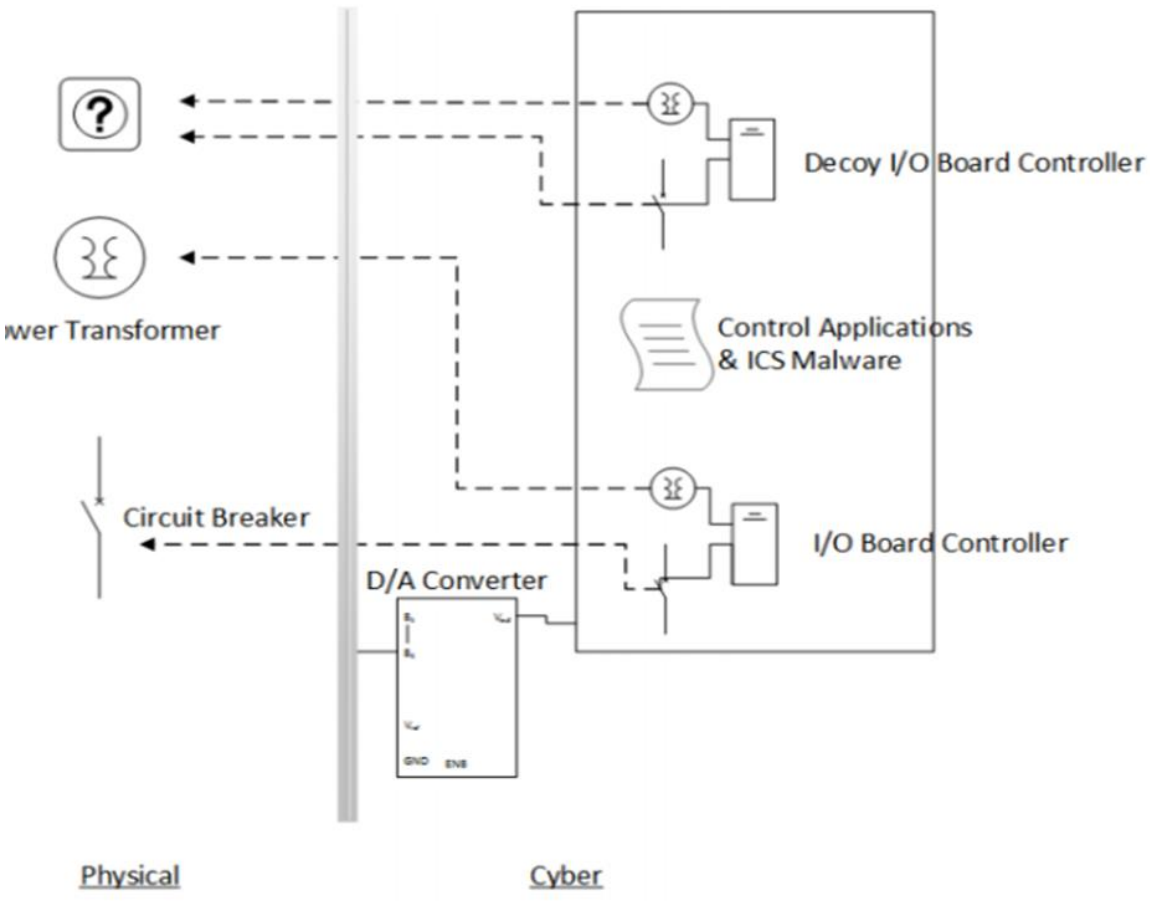

Figure. 12. Decoy physical equipment [39].

The prevalence of malware has been steadily increasing due to the expanding volume of data being transmitted via contemporary communication networks. Malware has significant effects on both general-purpose computing systems and industrial control systems. In response, several defensive tools and techniques have been devised to counteract their actions. Honeypots are considered to be one of the defense tactics. Honeypots possess a high level of efficacy in detecting malware due to their passive nature, as they refrain from initiating any autonomous behavior. However, malware that is knowledgeable about decoys examines its targets for inconsistencies before proceeding

with its actions. In the event that malware identifies a potential victim as a decoy, it will retreat and remove itself, disappearing prior to the defender's detection of any indications. [40]

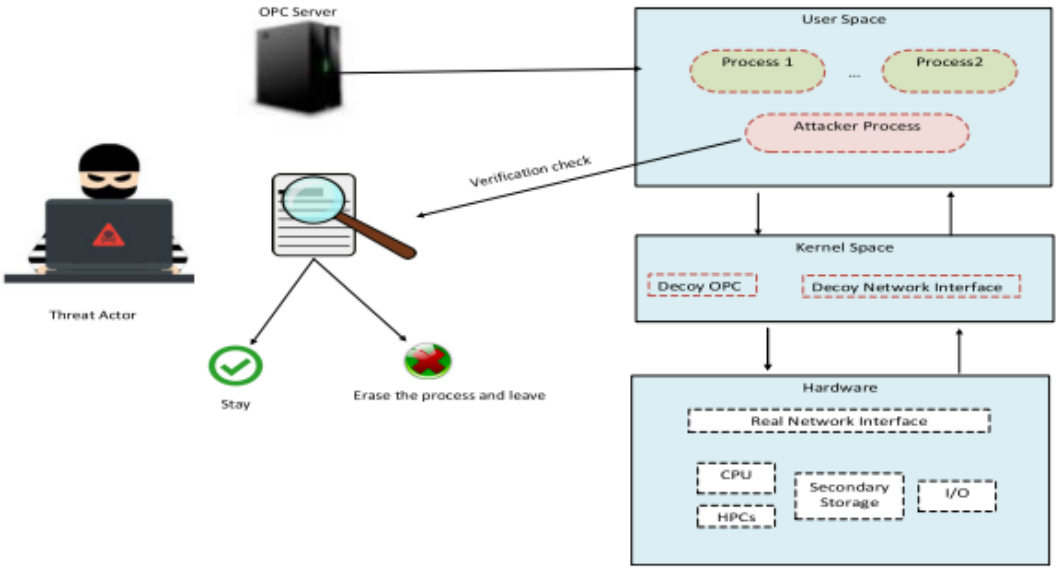

Figure. 13. Probing of decoy processes on a compromised machine.[39]

5.1 Decoy utilization

The utilization of decoy tactics has proven to be a successful strategy for protecting our computer systems and identifying possible attacks. One distinguishing feature of decoys is their deployment within the authentic system. Decoys possess the inherent benefit of efficiently unveiling covert attacks while obviating the need for legitimate users to authenticate themselves upon accessing the decoys. Park et al. generated Java-based code as a software decoy to use for obfuscation techniques. [105]. Lee et al. introduced a novel approach for generating decoy files through an analysis of the Ransomware.[41]

```
# Pseudo code for a decoy file

# Define a fake decoy file with a tempting name and content
decoy_file_name = "important_document.docx"
decoy_file_content = "This is a highly confidential document. Access restricted."

# Monitor access to the decoy file
while True:
    user_action = getUserAction()  # Get the action performed by a user

    # Check if the user tries to access the decoy file
    if user_action == "access" and user_requested_file == decoy_file_name:
        logAccessAttempt(user, decoy_file_name, "decoy")
        alertSecurityTeam(user, "Attempted access to decoy file")

    # Continue monitoring other user actions
    # ...

# Define functions for logging access attempts and alerting the security team
function logAccessAttempt(user, file_name, file_type):
    log_entry = "User " + user + " attempted to access " + file_name + " (" + file_type + " file)"
    writeToSecurityLog(log_entry)

function alertSecurityTeam(user, message):
    security_alert = "ALERT: User " + user + " - " + message
    sendAlertToSecurityTeam(security_alert)

# Function placeholders for getting user actions, writing to security logs, and alerting the security team
function getUserAction():
    # Implement code to get user actions, e.g., reading user input
    pass

function writeToSecurityLog(log_entry):
    # Implement code to write log entries to a security log file
    pass

function sendAlertToSecurityTeam(alert_message):
    # Implement code to notify the security team, e.g., via email or a message
    pass
```

Figure. 14. Decoy Pseudo Code Simulating the Behavior of a Decoy

5.2 Decoy processes.

This section will outline the step-by-step procedure for creating a dynamic display specifically designed for decoy processes. The objective of this study is to simulate a deceptive procedure, referred to as a decoy process, in a manner that convincingly presents the process to potential attackers as a fully operational system with performance characteristics identical to those of its legitimate counterparts. The achievement of this objective is realized through the projection of the decoy process as a novel process [42]. Due to its nonexistence, our procedure does not utilize any of the tangible resources

present in the system. The initiation of our process differs from the establishment of our procedure. When a genuine process is initiated, there is a requisite overhead that must be incurred to allocate resources, data, code, and libraries for the purpose of storing the simulated process in the primary memory. This phenomenon is commonly referred to as the cost of reproduction.

In order to initiate a new process, it is necessary to first halt the ongoing execution. This guarantees that the CPU scheduler will refrain from initiating the new process. Nevertheless, as previously mentioned, the act of halting execution to produce a stable state proves to be inefficient due to the continued utilization of CPU cycles and secondary storage.

The task at hand presents us with a dilemma as we strive to distinguish our technique from any possibly deleterious activity. Malicious actors employ techniques to obfuscate their processes by manipulating the usual execution of system calls.

Modifications are implemented to the system calls and process manager in a manner that suppresses the inclusion of information pertaining to malicious processes within the returned list. Consequently, the process manager effectively circumvents the alterations. Nevertheless, there are established techniques that can reveal the presence of a concealed process using this approach. Various methods can be employed to accomplish this work, including the utilization of tools such as the task management tool. [43]

5.3 The influence of input/output data on the consistency of decoy processes

A suitable procedure will encompass input/output (I/O) read and write operations, enabling it to both retrieve data and generate output. The input/output operations are produced by various devices, including files, keyboards, screens, hard disks, and network

interface devices. Despite the availability of performance counters for monitoring I/O activities, malevolent entities retain the capability to clandestinely check the processes in which a process engages with I/O devices. The researchers possess the capability to analyze every input/output (I/O) data flow and see the total lack of I/O operations involving a specific target process. Consequently, in order to enhance the efficacy of the decoy process in deterring malware, it is imperative for the dependability of resource utilization by decoy processes to align with the input/output (I/O) traffic associated with I/O devices.

5.4 Defining an input data set

This section will describe the process of defining the input data sets X and Y in order to incorporate the two key variables, namely transition probability and transition time frame. Multiple models can be generated by utilizing the specifications of input data sets. The aforementioned data sets pertain to a shared model but exhibit differences in temporal aspects and transition probabilities.

The capabilities of the OPCExplorer process were divided into many jobs. The input data X and Y are defined through the decomposition of the process functionality into individual tasks, resulting in the aforementioned conclusion. In a more precise manner, X consists of input data values that are expressed through the identical model, utilizing the same time and transition probability parameters. Similarly, Y is composed of input data values that are characterized by the identical model, employing the same transition probability parameters. The data values in Y necessitate recalculating the time and probability due to their lack of correspondence with the previous parameters. To provide further clarification, the specific task of obtaining OPC server properties outlined in Table

1 will include a unique code path. Consequently, this will lead to varying probabilities and transition durations in both the outer and inner automatons. This is in contrast to the actions of removing a tag from a collection or retrieving information from a tag [44].

Nevertheless, our model takes into consideration the variability that exists among different occupations in terms of the durations and likelihoods of transitioning. At some intervals, the rate of transitions between states may exhibit greater frequency compared to other intervals. Our technique remains unaffected by variance as we meticulously record each state change, including its chance of transitions and length.

5.4.1 Structured Data Collection:

A structured collection of data refers to a systematically ordered assemblage of data, typically arranged in a structured manner, such as in rows and columns or in a suitable alternative format. In a tabular format, each row typically corresponds to an individual data point or observation, while the columns of the table often pertain to different attributes or features associated with the data points [46].

5.4.2 Raw Data:

Raw data refers to datasets that do not require extensive processing and are either unprocessed or have only undergone minimum processing. The datasets presented above are unprocessed and unaltered information derived from the real world.

	A	B	C	D	E	F	G	H	I	J
1	VL1	VL2	VL3	IL1	IL2	IL3	VL12	VL23	VL31	INUT
2	249.1	247.5	247	62.8	29.4	71	430.9	427	430.8	38.2
3	249.1	247.5	247.2	58.1	32.6	69.2	430.9	427.2	430.9	31.9
4	249.2	247.7	247.4	62.3	31.9	64.8	431	427.6	431.1	31.5
5	249.5	247.9	247.7	55.6	25.9	60.1	431.6	428.1	431.6	32.7
6	249.7	247.9	247.9	54.9	42.6	58.2	431.6	428.4	432	14.1
7	249.8	248.3	247.8	53.8	24.7	65.2	432.2	428.2	432.3	36.5
8	249.9	248.2	248	54.2	40.6	57.4	432.1	428.7	431.9	15.5
9	250.1	248.7	248.2	55.5	24.9	59.2	432.9	429.1	432.7	32.7
10	250.1	248.5	248.1	45.8	31.7	55.8	432.5	429	432.5	20.8
11	250.1	248.5	248.4	48.5	34.3	56.3	432.5	429.2	432.8	19.4
12	250.5	248.7	248.4	42.1	27.1	54.5	433.1	429.4	433.3	23.4
13	250.6	249	248.7	44.9	29.4	55	433.4	429.9	433.6	22.2
14	250.7	249	248.7	44	32.2	54.9	433.5	429.8	433.4	19
15	250.8	249.2	248.9	44.9	29.1	55.9	433.7	430.2	433.8	23.8
16	250.8	249.1	248.8	42.4	22.7	53.9	433.7	430	433.9	27.1
17	250.7	248.9	249	43	35.4	50.7	433.4	430.2	433.8	13.3
18	251	249.2	249.1	41.1	29.7	51.9	433.9	430.4	434.2	19.4
19	251.1	249.3	249.1	38.8	35.1	53.6	434	430.6	434.2	17
20	251.1	249.5	249.2	38.7	29.2	54.5	434.3	431	434	22

Figure 15. Row Data Set.

5.4.3 Initial Data Source

In a broad sense, input datasets often include unprocessed or minimally processed data. The datasets encompass unaltered information collected from the real world prior to any analysis or manipulation.

5.4.4 Specific to Use

The layout of an input dataset frequently has a purpose-driven nature, indicating that it is tailored to the particular problem or task at hand. When conducting an analysis of data related to client sales, the input dataset may include various pieces of information, such as customer names, purchase quantities, and purchase dates.

5.4.5 Data Types and Variable Features:

The input datasets have the potential to encompass a diverse range of data types, such as numeric data, text data, categorical data, image data, audio data, or a combination thereof, among other possible data types. The selection of acceptable data types should be guided by the problem being addressed. The Diverse and Multifaceted Attributes in a given input dataset, the columns, which are often referred to as variables, often represent diverse attributes, qualities, or features associated with the individual data points. As an illustration, the variables inside a dataset concerning residential properties may encompass the aggregate area in square feet, the quantity of bedrooms, and the corresponding selling price.

5.5 Conclusion:

In the area of data-driven decision-making, analytics, and machine learning, the significance of input datasets cannot be overstated. Initially, it is important to collect the data, engage in preprocessing, and ultimately transform the data into a suitable format. It is only at that point that one can derive perceptive judgments or make accurate predictions. Data scientists and analysts often allocate a significant amount of their effort to the task of cleansing and preparing input datasets, ensuring their suitability for analysis or modeling purposes.

CHAPTER SIX

APPLICATION AND FUTURE RESEARCH

Protecting against cyberattacks with computer-generated intelligence (CGI)

The significance of security has been greatly amplified due to the rapid advancement of technology. The primary factor contributing to this phenomenon is the increase in malware incidents detected in industrial computer systems, resulting in potential harm to both the computer infrastructure and the individuals or organizations involved. To maintain the engagement of the stakeholders. The term "malware" encompasses a set of malevolent programming codes designed to inflict harm upon computer systems, software, or web applications. Based on the results of an investigation, it has been observed that software programs exhibit an inability to discern between valid system calls and those that are malicious in nature. To ensure the protection of stakeholders, it is imperative that computer systems and web apps are designed with the capability to identify and distinguish between malicious activities and approved operations performed by applications. Artificial intelligence (AI), machine learning (ML), and deep learning (DL) represent groundbreaking concepts that can be effectively employed by various algorithms aimed at identifying harmful software activities. This article presents a proposal for employing artificial intelligence (AI) techniques to identify and mitigate malware activity occurring in computer memory. The objective is to safeguard sensitive data kept in memory by preventing unauthorized access by malware. These techniques enable the identification of malicious software operations occurring in computer memory. This study introduces a revolutionary technology that utilizes the k-means algorithm and

the elbow technique, together with other functionalities, to effectively classify physical data stored in memory. The proposed technology aims to achieve prompt detection of such data.

6.1 Electrical Power Grid (EPG)

The growing number of malware incidents in industrial computer systems has generated substantial apprehension regarding security. The primary reason for concern over this issue derives from the possibility of physical harm to computer systems as well as the negative repercussions it may have on various stakeholders, including hospitals, corporations, and dwellings. The electricity sector is a highly susceptible industry that experiences a significant frequency of cyberthreats. The expanding scope of threats has resulted in their infiltration into industrial control systems.

The energy sector is one of the top three industries in the United States that are vulnerable to deliberate attacks. In 2016, the energy sector saw a cumulative count of 59 occurrences. The energy sector has been subject to significant targeting attacks, not only within the United States but also across Europe and Japan. [38]

To overcome these challenges, the power sector must allocate significant resources towards implementing resilient cybersecurity protocols, regularly maintaining and updating systems, conducting thorough risk evaluations, and collaborating with governmental entities, industry counterparts, and cybersecurity experts to fortify its safeguards against cyber hazards. Furthermore, there is a growing focus on the implementation of norms and regulations to aid in safeguarding critical infrastructure against cyberattacks.

There are multiple vulnerabilities caused by the following factors:

6.1.1 Critical infrastructure.

This refers to the essential systems, networks, and assets that are vital for the functioning of a society, economy, or organization. The generation, transmission, and distribution networks of electrical power exemplify crucial infrastructure, playing a vital role in facilitating the functioning of contemporary society. Disruptions within the electricity industry can yield significant consequences, extending beyond the realms of the economy and society and encompassing national security as well. As a result of this, threat actors may find cyberattacks targeting this industry attractive for the purpose of causing harm or gaining leverage.

6.1.2 High Stakes.

The potential consequences of a successful cyberattack against the electrical sector might be quite detrimental. These attacks have the potential to cause power outages, leading to the deprivation of energy access for significant populations, causing financial repercussions, and, in certain cases, endangering lives.

6.1.3 Complex Systems.

The electricity sector relies on intricate systems comprising diverse equipment and networks. The interconnectivity of these systems is increasing as a result of the integration of digital technology, hence rendering them susceptible to potential cyber threats. Due to its intricate nature, the system may be vulnerable to attacks originating from diverse sources.

6.1.4 The Focus of Interest.

Cyberattacks targeting the power sector can be executed with diverse motivations, encompassing the pursuit of economic benefits, political objectives, or industrial espionage. The sector may be targeted by many entities, such as state-sponsored actors, criminal organizations, and hacktivists, due to a range of motives.

6.1.5 Insufficient Implementation of Security Measures.

Efforts have been made by the power sector to enhance cybersecurity, yet the presence of vulnerabilities persists owing to issues such as the utilization of outdated technologies, insufficient allocation of resources towards cybersecurity, and the inherent difficulty of effectively safeguarding extensive, interconnected networks. Notwithstanding these endeavors, vulnerabilities persist.

6.1.6 The dynamic nature of the threat landscape.

The dynamic nature of the threat environment is characterized by constant evolution as novel and increasingly intricate techniques of cyberattack continue to emerge. Threat actors consistently engage in the exploration of innovative methods and tactics to exploit weaknesses within the electrical sector's infrastructure.

6.2 Malware.

The evolution of these viruses was also modified due to developments in technological innovation [39]. The Morris worm, ILOVEYOU, and Melissa represent a selection of malicious software instances that have been identified within the last few decades.

These viruses possess the capability to infect software applications utilized in business, enterprise, or organizational settings. Their propagation can occur through

routine usage or downloading, the installation of commercial software, malicious intent, or simply by clicking on a predetermined hyperlink. The potential impact of these software applications extends to government systems, data centers, laboratories, and hospitals. According to a study [3], it has been determined that in contemporary society, the predominant mode of operation revolves around applications, with a significant portion of services being delivered through online platforms. This stands in stark contrast to the historical context, wherein the majority of services were rendered through face-to-face interactions. As a result of this, there is a substantial likelihood of encountering malicious software. Moreover, hazardous applications possess the capacity to rapidly propagate through wireless networks, infecting any interconnected devices within those networks. In the event that a single device inside a network becomes compromised and unauthorized individuals gain control over it, they can exploit the infected device as a host and utilize the network's Wi-Fi capabilities as a modem to establish communication with other devices within the network [41]. If we use a hospital as a representative case for the potential consequences of malware attacking the electrical transformer of a city, several outcomes may arise, encompassing a multitude of possible scenarios. The availability of a comprehensive array of electrically operated medical devices is crucial for effectively preserving the lives of patients.

This category encompasses various medical devices such as ventilators, incubators, suction units, and equipment utilized for dialysis therapy. The loss of electrical power in a residential care home or healthcare facility poses a significant risk to the lives of numerous individuals. The occurrence of an abrupt power outage during the execution of a delicate treatment can lead to severe challenges of a catastrophic nature. The patient's

life is at risk when essential monitoring devices fail to operate efficiently due to a power shortage, hence increasing the probability of patient mortality. Hospitals and other medical institutions typically maintain a disaster recovery plan to mitigate the risk of power outages and their adverse consequences. However, the installation of backup generators serves as an additional measure to ensure uninterrupted power supply and the ability to sustain essential services, particularly during emergency situations. Notwithstanding the fact that hospitals and other medical institutions consistently maintain preparedness through the implementation of disaster recovery plans, despite the potential for mitigating power disruptions, the current state of affairs persists.

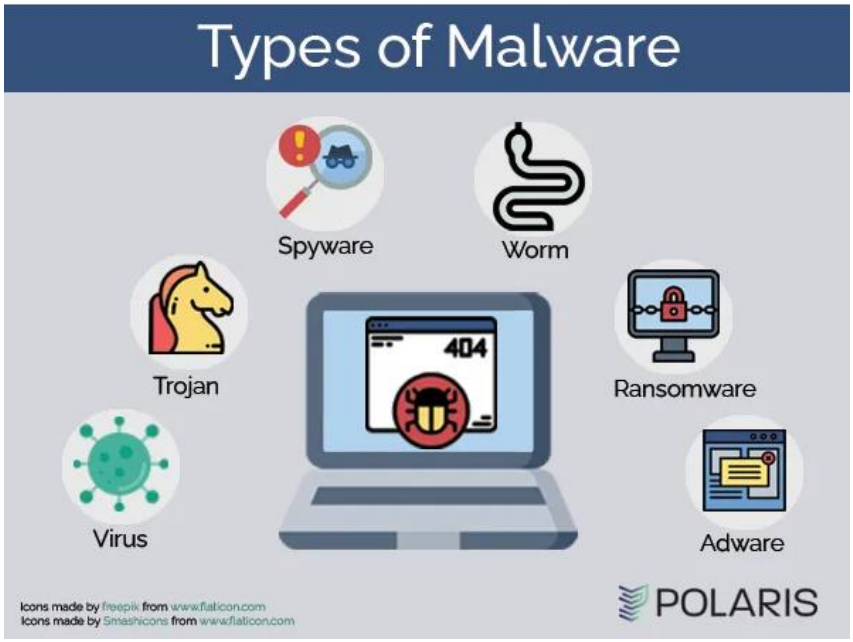

Figure. 16. Types of malware

6.3 Threads

The process encompasses many instances of an execution unit referred to as threads, with each thread having its own program counter, stack, and registers. Threads

cannot exist in isolation from any process. Moreover, it should be noted that each thread is exclusively linked to a single process. The many threads possess the capability to exchange information, including code segments, data segments, and file segments, among themselves [46].

Utilizing threads in programs is a prevalent approach to enhancing performance through the implementation of parallelism. The central processing unit (CPU) is capable of executing only one thread at a given moment; however, it employs rapid thread switching to create the illusion of concurrent operation of several threads. In the context of a computational process, a thread functions as the fundamental entity of execution. The number of threads linked to a singular process might vary from a minimum of one to a maximum of multiple threads.

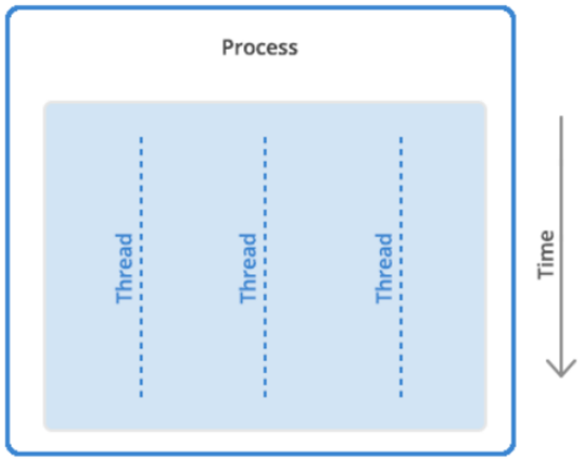

Figure. 17. Thread in a process

Various applications across a wide range of contemporary operating systems extensively use threads, which play a crucial role in concurrent programming. These apps encompass both web servers and desktop programs. When engaging in the development of

multi-threaded applications, developers are obligated to confront several complexities and obstacles associated with the synchronization and management of resources. The complications and obstacles arise due to the presence of multi-threaded programs.

The utilization of threads can lead to improved program performance and responsiveness, especially in computing systems that feature several cores and processors. Threads can be utilized to do many tasks, such as parallel processing, concurrent I/O activities, and the development of responsive user interfaces. A process has the ability to generate threads and one or more threads may be used in the execution of a process. The generation of new threads is enabled and supported by several system calls and libraries in many contemporary operating systems. For instance, the thread library is frequently utilized in a Linux-based system for the purposes of creating and managing threads.

The scheduler, which is an integral component of the operating system, assigns CPU time to various threads within each process. Threads possess different priorities, and the scheduler determines the order in which threads are run by considering these priorities and the currently employed scheduling mechanism.

It is much easier for threads within the same process to talk to each other and share data than for threads within different processes, which need to use inter-process communication (IPC) protocols. However, it is imperative to provide data access synchronization as a precautionary measure to mitigate the occurrence of data races and other concurrent issues. Termination of a thread can occur either deliberately or after the associated process concludes. Upon the termination of a thread, it is imperative to release the resources connected with it and perform the necessary cleanup of any associated data.

6.4 Thesis contribution:

Taking inspiration from the concept of software decoy, it is worth considering the possibility of developing an operating system that periodically executes a replication of such a process in the form of a decoy thread. The proposed approach involves the creation of a replica in the form of a thread, commonly known as a decoy thread or a shadow thread. This shadow thread can be combined with decoy sensors and actuators, thereby providing a simulated environment for potential malware that may be operating within the host's operating system process. However, in this particular scenario, we have the opportunity to conduct intrusive observations, as this is an accurate replication of the authentic process. Therefore, including our analysis of the ongoing events will not adversely affect the actual process. Nevertheless, the creation of a decoy thread within the operating system process allows for the observation of malware instructions within the same environment as the main operating system process. It is important to note that our objective is not to directly target the original structure of the malware. We are actively pursuing the identification and containment of a malicious software entity embedded under a deceptive facade. When the malware attempts to execute and gain access to the physical data structure, potentially reading buffers that include sensor data, our objective is to identify and isolate a replica of the malware structure within a decoy in an intrusive manner. Our objective is to identify and neutralize a malicious software structure that has infiltrated a deceptive entity. When the malware attempts to execute and gain access to the physical data structure, potentially reading buffers that include sensor data, our objective is to identify and isolate a replicated instance of the malware structure within a deceptive intrusion. The Decoy thread is eliminated subsequent to a period of non-observance of any

anomalous activity. Subsequently, a new decoy thread is created by duplicating the existing address space of the primary process. This iterative process is continued until any form of activity is detected, such as the execution of instructions or the accessing of decoys.

6.5 Decoy Thread.

The term "decoy thread" is not often used within the domain of computer science or in the context of operating systems. In some security or cybersecurity contexts, the use of a decoy or fake thread may be utilized as a strategic measure to deceive or redirect potential threats or assailants. This is achieved by using a simulated thread. The concept in concern is occasionally deliberated in correlation with the broader subject matter of honeypots or honeynets. In the past few years, there has been a notable convergence of artificial intelligence and machine learning within the area of cybersecurity, resulting in their heightened significance as indispensable instruments. The findings of the present investigation indicate a deficiency in the ability of current programs to differentiate between legitimate system calls and those that are intended to be harmful. Therefore, it is crucial to develop computer systems and online applications that possess the ability to differentiate malicious behaviors from regular program activities.

The following examples are some of the decoy threads used in cybersecurity.

6.5.1 Honeypots and Decoys.

Honeypots refer to intentionally configured computer systems, networks, or resources designed to attract and deceive potential hackers with the aim of inducing them to launch attacks. Honeypots serve the purpose of conducting research and assessment on

the actions of malicious actors while simultaneously acting as a deterrent against potential attacks on authentic production systems [47].

6.5.2 Decoy Thread in Honeypots.

When examining the concept of honeypots, the phrase "decoy thread" might be applied to denote a simulated or false sequence of events occurring within the honeypot. This activity has the potential to replicate the actions of a legitimate user or system process, creating the illusion of authentic network traffic or user interaction. Using fake threads gives potential thieves the impression that the honeypot is more real and convincing than it really is.

6.5.3 Deception and Monitoring.

Security professionals possess the capacity to gain valuable knowledge regarding the interactions between malicious actors and a system, the strategies they use, and the particular vulnerabilities they aim to exploit through the utilization of deceptive elements within a honeypot environment. The collected data holds potential use in the realms of threat intelligence and security analysis.

It is crucial to recognize that the use of honeypots and decoy threads is a specialized technique in the field of cybersecurity that security experts frequently employ to gain insightful knowledge into the strategies and methodologies used by malicious actors. It is crucial to keep in mind that honeypots and decoy threads hold significant importance. While these approaches can provide valuable guidance, it is crucial to establish and monitor them diligently to prevent any inadvertent risks that may undermine the security of your network.

6.6 Machine learning approach.

This study aims to utilize artificial intelligence (AI) and machine learning (ML) techniques for the purpose of identifying and mitigating malware attacks against power transformers, with the ultimate goal of protecting the physical integrity of these critical infrastructure components. The aforementioned attacks aim to mitigate potential physical harm to the power transformer by incapacitating the operators, disabling the protection algorithms, and manipulating controllable functionalities of the power transformer, ultimately leading to its failure [48].

The utility of my approach extends to several circumstances, encompassing both pre- and post-construction phases of a decoy thread, as well as other relevant scenarios. It is advisable to periodically duplicate the decoy thread, with a suggested frequency of every two minutes. In the event that the primary thread becomes compromised during these intervals, the malware will not be found in the replicated decoy thread. Therefore, the cyber deception probing conducted within the decoy thread is fundamentally futile and lacks potential for meaningful outcomes. Within the framework of my research, the detection of an anomaly serves as a trigger for the creation of a decoy thread in this moment in time.

6.6.1 ML Methods and Functionality

The term "machine learning" (ML) encompasses several methods and techniques that enable computers to learn from data and make predictions or decisions without explicit programming. Below is the ML model I have employed in this research.

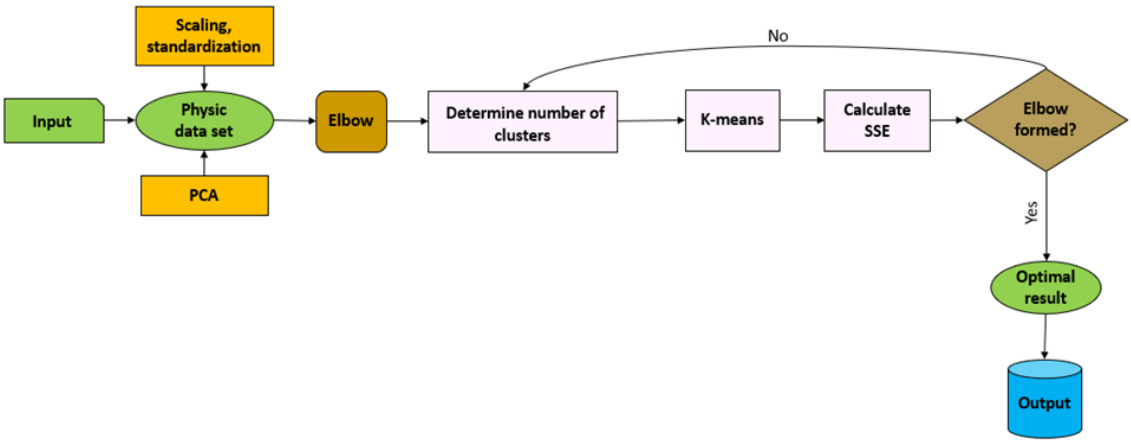

Figure. 18. Proposed System Model

6.6.1.1 Preprocessing

Scaling is a common need for numerous machine learning estimators when dealing with datasets that contain independent variables with varying ranges. Inadequate scaling of the variables could cause some factors to dominate the objective function, which would make it more difficult for the estimator to effectively learn from other characteristics. This can lead to inaccurate predictions in certain scenarios. One widely recognized approach for standardizing and normalizing data is implemented through the utilization of the Scikit-Learn function in the Python programming language. In the event that the data set lacks scaling and standardization for each specific characteristic, it is imperative to address this by applying a mean of zero and ensuring that there is no unit variance. Consequently, the estimator may demonstrate unfavorable characteristics in this particular situation. From a mathematical perspective, the methodology for distributing and centralizing the dataset involves initially subtracting the average value of each specific attribute. Subsequently, the non-constant features should be scaled by dividing them by

their respective standard deviations. The subsequent depiction pertains to the standard deviation that is applicable to this particular procedure [49].

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

In this context, the symbol "z" represents the standardized or normalized value, "x" represents the individual feature value of the data point, "μ" represents the mean of the population, and "σ" represents the population standard deviation for the dataset.

6.6.1.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a technique utilized for the purpose of reducing the dimensionality of large datasets. This process involves the conversion of a substantial dataset comprising multiple variables into a more compressed dataset while aiming to retain the maximum amount of information possible within human limitations. PCA is often used to expedite machine learning methods by representing them as linear combinations or mixtures of the original variables. This utilization of PCA has considerable importance.

- **Covariance Matrix:** The covariance matrix of a dataset X with n samples and m features can be calculated as:

$$Cov(X) = \frac{1}{n-1} \sum_{i=1}^{n} (xi - \overline{x})\,(xi - \overline{x})^T \quad (2)$$

- **Eigenvalue Decomposition**: PCA involves finding the eigenvectors and eigenvalues of the covariance matrix. The eigenvalue decomposition of the covariance matrix is given by:

$$Cov(X) = Q\Lambda Q^T \quad (3)$$

- **Projection:** To reduce the dimensionality of the data, you can project the original data onto the selected principal components. The projection of data point x_i onto the first k principal components is given by:

$$x_{i}^{!} = x_k^T \; x_i \tag{4}$$

6.6.1.3 K-Means++ Method

The K-means++ initialization method is one of the most popular options available when it comes to the K-means clustering algorithm. It was initially discussed in the book titled "k-means++: The Benefits Obtained by Planting Seeds with Great Care," which was written by David Arthur and Sergei Vassilvitskii [1]. K-means++ has made a substantial contribution to the field of clustering, and it continues to be an essential method for initializing K-means clustering algorithms. K-means++ has had a considerable impact on the clustering field. K-means++, which was first introduced, has quickly established itself as the method of choice for initializing centroids in K-means clustering since its launch due to the major advantages it offers over random initialization. As a result of these benefits, the K-means++ technique has become the norm in practice for initializing centroids in the K-means clustering algorithm. Researchers and practitioners typically utilize it as a starting point for further studying more complicated clustering algorithms and improving the quality of their clustering findings.

The equation W (C_k) represents the sum of the within-cluster variances.

$$W(C_k) = \sum_{x_i \epsilon C_k} (x_i - \mu_k)^2 \tag{5}$$

Here, x_i denotes a data point belonging to a certain cluster, C_k represents an indicator variable indicating the cluster membership for each data point, and μ_k denotes the mean value of the data points assigned to the cluster C_k. The measurement of data compactness can be quantified using the term "total within-cluster sum of squares" (TW). This term is defined as the sum of the squared distances of all data points within their respective clusters.

$$TW = \sum_{K=1}^{K} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2 \quad (6)$$

6.6.1.4 Elbow Method

The elbow method is a commonly used technique in cluster analysis for determining the most appropriate number of clusters in the K-means algorithm [50]. This method involves plotting the variation and identifying the point on the curve where a distinct bend, or "elbow," occurs. This bend represents the optimal number of clusters to use, as adding more clusters beyond this point does not significantly improve the modeling of the dataset. Initially, the addition of clusters contributes a substantial amount of information, but eventually, the marginal benefit of including additional clusters drops, making the elbow point the optimal choice. The elbow criterion is a method used to calculate the optimal value of "k," which represents the number of clusters to be selected. The procedure will commence with an initial value of k equal to 2, and subsequent to each iteration, it will be incremented by one. The sum of squared errors (SSE) is determined by the value of k. The next step is to generate a line chart illustrating the sum of squared errors (SSE) for various values of k. The point at which the line chart displays an elbow-like shape indicates the execution of the elbow method. At this point, the optimal number of clusters, denoted as k, is identified for the K-means algorithm [51]. Equation 4

introduces the elbow technique, also known as the within-group sum of squares (WSS). This strategy quantifies the squared average distance between all data points within a cluster and the centroid of that cluster. It provides a statistical measure of the distance from the group averages to the same cluster centroid.

$$WSS = \sum_{i=1}^{m}(x_i - c_i)^2 \quad (7)$$

The optimal number of clusters, denoted as k, can be estimated by combining the K-means algorithm with the elbow approach. The subsequent statement can be utilized to characterize the elbow technique by utilizing the sum of the squares error [52].

$$SSE = \sum_{k-1}^{k}\sum_{xi\in Sk}\|x_i - c_k\|_2^2 \quad (8)$$

The value of the SSE is equal to zero. When the value of k is equal to the total number of data points in the dataset, each individual data point forms its own cluster, resulting in zero error between the data point and the center of its respective cluster. When the value of k is equal to the number of clusters generated, denoted as C, and i represents the i^{th} cluster, and X represents the total number of data points in each cluster, there is no errors observed between the cluster and its own center.

6.6.1.5 (SSE) sum of the squared error.

The Sum of Squared Errors (SSE) is computed by summing the squared Euclidean distances between each data point and its corresponding centroid, which is determined as the centroid that is geographically closest to the point. The objective of the k-means++ method is to minimize the measure of error, hence achieving the minimum feasible value [53].

$$SSE = \sum_{i=1}^{n}(x - \overline{x})^2 \quad (9)$$

6.7 Dissertation contribution.

The fundamental objective of this dissertation is to detect malware within industrial computer systems, namely those utilized in power grids or power transformers. All forms of malicious software or threat actors present a potential hazard to the availability, control, and security of industrial control systems and processes.

In this dissertation, we have discussed our research, which is aimed at developing unique methods to detect, prevent, and neutralize, in a timely manner, advanced computer attacks that target industrial equipment. Industrial computer systems (ICS), which include the power grid, are extremely valuable pieces of equipment that manage and track the currents and voltages that are supplied to various parts of a city or town. In the event that these electronic devices become attacked or compromised, there is the potential for hazards and risks to human life as well as the environment. We depend on the power infrastructure to supply us with reliable electricity, which serves as the lifeblood of our society. These cyberattacks frequently manifest in the form of self-replicating malicious software, possessing the capacity to infiltrate industrial computer networks at an advanced level. Upon gaining access, individuals have the capability to inflict severe physical destruction on power plant machinery or engage in covert cyber-espionage activities. The compromise of a supply chain facilitates the occurrence of these attacks in an additional manner. The clear skill exhibited by these types of attacks underscores the imperative need for ongoing real-time monitoring of cyber-physical critical infrastructures, encompassing both inadvertent and intentional disruptions. The operators responsible for managing the power grid are going to have the capability to react quickly to potential threats or failures of the power grid as a result of the implementation of such monitoring systems. If left

unchecked, these attacks possess the capacity to do harm to the physical infrastructure and operational procedures of a power system.

In the electrical world, transformers are passive parts that are very important for moving electricity between circuits efficiently through electromagnetic induction. To make electrical power suitable for transmission and distribution, voltage levels are adjusted while ensuring the frequency remains consistent.

To maintain the proper and secure operation of power transformers, a variety of sensors and measurement equipment have been used for live monitoring and control purposes. Each sensor has the capability to produce data related to the attributes of the transformer, including voltage, current, temperature, and oil level. Known as physic data, the data collected from Kaggle is a widely recognized internet-based platform that serves as a host for competitions within the domains of data science and machine learning [29]. These simulation data were generated via IoT devices. I will utilize the provided datasets as input for the machine learning model during development. Afterwards, I will execute the appropriate preprocessing methods to guarantee data accuracy and integrity. Following this, I will systematically describe each step before transitioning to the subsequent discussion segment [55].

VL1	VL2	VL3	IL1	IL2	IL3	VL12	VL23	VL31	INUT
233.2	233.7	233	1.1	0.9	1.1	405.5	405.3	405.2	0.1
233.4	233.8	233.3	1.2	1.1	1.3	403.7	403.6	403.5	0.1
233.4	234	232.8	108.5	98.7	139.8	405	404.4	404.2	37.8
233.3	233.9	233.4	155.3	159.9	148.6	404	404.7	403.7	10
232.5	232.8	232.4	143.8	173.1	157.7	403.9	405.1	404.5	25.5
232.4	232.9	232.3	150	141.5	152	402.6	402.9	402.3	9.4
232.2	233.1	232.1	148.3	124.5	130.9	404.5	404.5	403.2	22.7
232.1	232.7	232	136.1	141	127.8	403.3	403.8	402.9	11.1
232.4	232.9	232.1	132.1	148.2	135.5	402.8	403.1	402.4	15.9
232.8	233	232.4	124.9	139	132.2	402.1	402.4	402	12
232.6	232.9	232.1	114	124.2	131.1	402.8	403.1	402.6	9.8
230.7	231.4	230.6	126.8	124.3	130.1	402.3	402.8	401.9	5.7
232.3	233	232.1	131.3	120.6	151	402.4	402.5	402.2	28.4
232.3	233.1	232.2	127.8	119.7	135.7	403.7	403.8	403.1	13.8
233.2	234	233.1	121.4	109.3	134.5	405.3	405.3	404.7	21.4

Figure. 19. Current Voltage Data Set.

WL1	WL2	WL3	VAL1	VAL2	VAL3	RVAL1	RVAL2	RVAL3
35.3	36.4	33.2	36.1	37.2	34.6	7.6	7.3	0.0096
32.5	39.5	34.9	33.3	40.2	36.1	7.2	7.7	0.0092
32.2	32.2	34.3	32.6	32.8	35.3	5.2	6.3	0.0084
33.9	28	28.9	34.2	28.5	30.2	4.4	5.2	0.0086
30.9	32.2	28.6	31.3	32.7	29.8	4.9	5.5	0.0083
30.1	33.8	30	30.4	34.2	31.2	4.2	5.3	0.0084
28.7	31.8	29.1	28.8	32.1	30.3	2.8	4.4	0.0083
26.3	28.6	28.8	26.4	29	29.9	2.6	4.8	0.0082
28.8	28.2	29	29.1	28.6	30.1	3.8	4.8	0.0082
30.2	27.6	33.6	30.4	27.9	34.5	3.3	4	0.0078
29.3	27.3	30.1	29.4	27.6	31.1	2.9	4.5	0.0077
27.9	25.1	30.3	28.1	25.3	31.2	3.1	3.6	0.0074
25.5	26.5	27.5	25.7	26.9	28.7	3.5	4.7	0.0083
24.5	22.7	26.2	24.7	23.2	27.4	3.1	5	0.008
34.1	31	29.9	34.4	31.6	31.1	4.2	5.8	0.0085
36.1	29.8	31.5	36.4	30.2	32.7	4.4	4.5	0.0087

Figure. 20. Power Data Set.

PFL1	PFL2	PFL3	Avg_PF	Sum_PF	FRQ	THDVL1	THDVL2	THDVL3	THDIL1	THDIL2	THDIL3	MDIL1	MDIL2	MDIL3
0.99	0.99	0.98	0.99	8.1	49.8	1.7	2.1	1.7	7.9	11.6	5.8	189.5	201.3	215.7
0.99	0.99	0.98	0.99	7.5	49.6	1.8	2	1.5	7.7	10.7	4.9	189.5	201.3	215.7
0.99	0.99	0.98	0.99	8	49.9	1.6	2	1.7	5.4	12	5.5	189.5	201.3	215.7
0.99	0.99	0.97	0.99	7.9	49.9	1.7	1.9	1.8	14.7	10.8	5.6	189.5	201.3	215.7
0.99	0.98	0.97	0.99	8.6	49.9	1.7	1.8	1.8	5.7	9.3	6	189.5	201.3	215.7
0.99	0.99	0.98	0.99	7.6	49.9	1.6	1.8	1.8	5.1	10.1	6.2	189.5	201.3	215.7
0.99	0.98	0.98	0.99	7.5	50	1.6	1.9	1.7	4.7	9.4	6.3	189.5	201.3	215.7
0.99	0.98	0.98	0.99	8.6	50	1.6	1.9	1.8	4.4	10.5	5.2	189.5	201.3	215.7
0.99	0.98	0.98	0.99	8.5	50	1.5	1.6	1.7	6.5	9.9	6	189.5	201.3	215.7
0.99	0.98	0.98	0.99	7.6	49.8	1.5	1.9	1.5	4.9	8.8	5.5	189.5	201.3	215.7
0.99	0.98	0.98	0.99	8.1	49.9	1.6	1.7	1.7	5.1	9	5.8	189.5	201.3	215.7
0.99	0.98	0.98	0.98	9.3	49.9	1.5	1.6	1.6	4.7	8.4	5.4	189.5	201.3	215.7
0.99	0.97	0.98	0.99	9.4	50	1.4	1.6	1.6	5.8	8.6	5.8	189.5	201.3	215.7

Figure. 21. Power Factor data set.

KWH	KWH_I	KVARH	KW	KVA	KVAR	MPD	MKVAD
8145.99	14.469	8262.52	64.029	65.62	14.363	130.8	131.6
8162.99	17	8279.87	71.118	72.482	13.993	130.8	131.6
8181.4	18.406	8298.61	74.072	75.522	14.731	130.8	131.6
8199.99	18.594	8317.57	75.527	77.025	15.113	130.8	131.6
8220.18	20.187	8338.15	80.894	82.407	15.722	130.8	131.6
8240.27	20.094	8358.58	81.345	82.583	14.244	130.8	131.6
8260.68	20.406	8379.3	85.85	86.984	14.001	130.8	131.6
8282.3	21.625	8401.3	87.733	89.227	16.264	130.8	131.6
8305.02	22.719	8424.38	93.264	94.617	15.942	130.8	131.6
8328.77	23.75	8448.46	95.178	96.382	15.188	130.8	131.6
8352.58	23.813	8472.61	94.478	95.937	16.67	130.8	131.6
8376.93	24.343	8497.29	103.12	104.341	15.914	130.8	131.6
8402.08	25.157	8522.77	99.88	101.394	17.46	130.8	131.6

Figure. 22. Total Power data set.

The proposed system, as described by the flowchart in Fig. 3, will determine the optimal number of clusters to use during the process of data set evaluation, which will be carried out in four steps. The first step, known as preprocessing, will scale and standardize the dataset. The second step, known as PCA, will minimize the dimension of the dataset. The third step, known as the Elbow technique, will specify the most optimal number of clusters to be delivered to the final step, known as the Kmeans++ clustering algorithm, which will cluster and group data based on their similarities [56].

The initial stage involves commencing the processing of the dataset utilizing the proposed system. Scaling is a common need for numerous machine learning estimators when dealing with datasets that contain independent variables with varying ranges. Inadequate scaling of the variables could cause some factors to dominate the objective function, which would make it more difficult for the estimator to effectively learn from other characteristics. This can lead to inaccurate predictions in certain scenarios. One widely recognized approach for standardizing and normalizing data is implemented through the utilization of the Scikit-Learn function in the Python programming language. In the absence of scaling and standardization of the data set's individual features, it is necessary to process the data by applying a mean of zero and a variance of one. Consequently, the estimator may demonstrate unfavorable characteristics in this particular situation. From a mathematical perspective, the methodology for distributing and centralizing the dataset involves initially subtracting the average value of each specific attribute [57]. Subsequently, the non-constant attributes are scaled by dividing them by their respective standard deviations. Throughout the course of this procedure, As a result

of this, we will be able to commence the procedure of standardizing the dataset and including it within a consistent range.

The output for the dataset, both prior to and following scaling, is presented below.

Table 2. Current Voltage Physic Dataset.

VL1	VL2	VL3	IL1	IL2	IL3	VL12	VL23	VL31	INUT
249.1	247.5	247	62.8	29.4	71	430.9	427	430.8	38.2
249.1	247.5	247.2	58.1	32.6	69.2	430.9	427.2	430.9	31.9
249.2	247.7	247.4	62.3	31.9	64.8	431	427.6	431.1	31.5
249.5	247.9	247.7	55.6	25.9	60.1	431.6	428.1	431.6	32.7
249.7	247.9	247.9	54.9	42.6	58.2	431.6	428.4	432	14.1
249.8	248.3	247.8	53.8	24.7	65.2	432.2	428.2	432.3	36.5
249.9	248.2	248	54.2	40.6	57.4	432.1	428.7	431.9	15.5
250.1	248.7	248.2	55.5	24.9	59.2	432.9	429.1	432.7	32.7
250.1	248.5	248.1	45.8	31.7	55.8	432.5	429	432.5	20.8
250.1	248.5	248.4	48.5	34.3	56.3	432.5	429.2	432.8	19.4
250.5	248.7	248.4	42.1	27.1	54.5	433.1	429.4	433.3	23.4
250.6	249	248.7	44.9	29.4	55	433.4	429.9	433.6	22.2
250.7	249	248.7	44	32.2	54.9	433.5	429.8	433.4	19
250.8	249.2	248.9	44.9	29.1	55.9	433.7	430.2	433.8	23.8
250.8	249.1	248.8	42.4	22.7	53.9	433.7	430	433.9	27.1
250.7	248.9	249	43	35.4	50.7	433.4	430.2	433.8	13.3
251	249.2	249.1	41.1	29.7	51.9	433.9	430.4	434.2	19.4
251.1	249.3	249.1	38.8	35.1	53.6	434	430.6	434.2	17
251.1	249.5	249.2	38.7	29.2	54.5	434.3	431	434	22

Upon performing the scaling process, it becomes evident that the function effectively merges every aspect of the information into a unified range spanning from 0 to 1. The outcome is less than the numerical value.

VL1	VL2	VL3	IL1	IL2	IL3	VL12	VL23	VL31	INUT
-0.99	-1.022	-0.969	-2.974	-2.147	-3.148	-0.865	-0.884	-0.771	-1.952
-0.922	-0.989	-0.868	-2.971	-2.141	-3.141	-1.222	-1.225	-1.111	-1.952
-0.922	-0.923	-1.036	0.515	0.604	1.513	-0.964	-1.064	-0.971	1.564
-0.956	-0.956	-0.834	2.035	2.326	1.809	-1.162	-1.004	-1.071	-1.029
-1.229	-1.319	-1.171	1.662	2.697	2.115	-1.182	-0.924	-0.911	0.417
-1.264	-1.286	-1.204	1.863	1.808	1.923	-1.44	-1.365	-1.35	-1.085
-1.332	-1.22	-1.272	1.808	1.33	1.214	-1.063	-1.044	-1.171	0.156
-1.366	-1.352	-1.305	1.411	1.794	1.11	-1.301	-1.185	-1.231	-0.926
-1.264	-1.286	-1.272	1.281	1.997	1.369	-1.4	-1.325	-1.33	-0.478
-1.127	-1.253	-1.171	1.048	1.738	1.258	-1.539	-1.466	-1.41	-0.842
-1.195	-1.286	-1.272	0.693	1.322	1.221	-1.4	-1.325	-1.291	-1.047
-1.845	-1.782	-1.777	1.109	1.324	1.187	-1.499	-1.385	-1.43	-1.43
-1.298	-1.253	-1.272	1.255	1.22	1.89	-1.48	-1.445	-1.37	0.688
-1.298	-1.22	-1.238	1.142	1.195	1.375	-1.222	-1.185	-1.191	-0.674

Figure. 23. Current Voltage Scaled Dataset

In addition, the application of principal component analysis (PCA) will be performed on the dataset subsequent to the completion of the scaling stage. The primary objective of PCA is to achieve dimensional reduction, thereby reducing the number of dimensions in large datasets. This is accomplished by transforming a dataset comprising numerous variables into a smaller dataset while endeavoring to retain the maximum amount of information. The transformed dataset is constructed through the formulation of linear combinations or mixtures of the original variables. One of the most important applications of PCA is speeding up machine learning algorithms [58].

We have the flexibility to select the number of primary components that should be retained based on the level of dimensionality reduction that we would like to accomplish. For example, if we wanted to reduce the number of features in the data from 100 to 10, then we would keep the 10 major components that rated best in their respective rankings.

PC1	PC2
0.275619	0.00530761
0.325447	-0.00284764
0.0481042	-0.00268216
0.0481042	-0.00268216
-0.17377	-0.00255049
-0.0998115	-0.00259522
-0.0998115	-0.00259522
-0.216706	0.00351433
-0.17377	-0.00255049
-0.210749	-0.00252901
-0.0687896	0.00342721
-0.210749	-0.00252901
-0.0318108	0.00340493
0.0481042	-0.00268216

Figure. 24. PCA Output for the Current Voltage Scaled Dataset

Figure. 24 illustrates the outcome of using the Principal Component Analysis (PCA) function. It is evident that the dataset's dimensionality decreased from 10 features to only 2, while preserving a substantial amount of information through the generation of linear combinations of the original variables. This is evidenced by the observation that the dimensions of the dataset have decreased while retaining a significant amount of information.

Subsequently, the Kmeans++ algorithm will be utilized to partition the dataset into clusters, where the datasets will be assigned to groups depending on their distance to the respective cluster means. The use of this clustering methodology has been implemented to assess the data in a proficient manner and provide the requisite information for the purposes of data mining and pattern identification. This strategy contributes to reducing the discrepancies inside the cluster, namely in terms of squared Euclidean distances, which aligns with the objectives of these two procedures. Cluster analysis has been extensively utilized in various domains, including market research, pattern identification, data analysis, and image processing. The utilization of the Euclidean distance function renders it widely regarded as one of the most straightforward strategies for data mining partitioning and grouping. The value of K represents the number of groups contained in the data, whereas the primary goal of the K-means method is to identify and locate these groups [59]. The algorithm utilizes the specified properties in an iterative manner to allocate each data point to one of K groups. The data points within the dataset demonstrate clustering behavior based on the similarities seen among their respective attributes. In essence, our objective is to identify homogeneous subgroups within the dataset. These subgroups are characterized by data points that demonstrate high similarity to one another,

as determined by a similarity measure such as Euclidean-based distance or correlation-based distance. This clustering process aims to minimize the error sum of squares and the error criterion, which serve as the basis for determining the optimal number of clusters. The functional aims of this approach involve dividing the dataset into k divisions to meet specific requirements, even when dealing with non-numerical data. Because of its numerous advantages such as brevity, efficiency, and swiftness, the k-means++ clustering methodology is used.

```
KMeansPlusPlus(K, data):
    # K: Number of clusters to create
    # data: The dataset to cluster

    # 1. Select the first centroid randomly from the data
    centroids = [randomly select a data point from data]

    # 2. Repeat the following steps for K - 1 more centroids:
    for i = 1 to K - 1:
        # 3. For each data point, calculate its distance to the nearest existing centroid
        nearest_distances = []
        for each data point in data:
            distance = minimum distance to the centroids in centroids
            nearest_distances.append(distance)

        # 4. Calculate the sum of squared nearest distances
        total_squared_distance = sum of squared distances in nearest_distances

        # 5. Create a probability distribution based on squared distances
        probability_distribution = [squared_distance / total_squared_distance for squared_distance in nearest_distances]

        # 6. Randomly select the next centroid using the probability distribution
        new_centroid = randomly select a data point from data using probability_distribution

        # 7. Add the new centroid to the list of centroids
        centroids.append(new_centroid)

    # 8. Return the initial set of centroids for k-means clustering
    return centroids
```

Figure. 25 k-means++ algorithm pseudocode.

Subsequently, after the k-means++ stage has been completed, the second phase that has the most significance is the elbow approach. The aforementioned strategy is a heuristic approach employed for the purpose of determining the most suitable number of clusters within a given dataset. The k-means++ algorithm uses the method of visualizing variation to choose the most suitable number of clusters. The elbow approach involves the

generation of a cost function by systematically tweaking the parameter k and then visualizing its corresponding values [60]. Increasing the value of k will result in a decrease in the average distortion, a decrease in the number of instances per cluster, and a convergence of instances towards the centroids of their respective groups. Conversely, the extent of improvement in average distortion will become progressively less visible with the increasing value of k throughout every stage of the operation. The elbow value, denoted as k, refers to the point at which the reduction in distortion reaches its maximum extent. The value of k has been established, indicating that it is appropriate to discontinue the division of data into more clusters. Consequently, the inclusion of additional clusters does not yield any enhanced representation of the dataset. Nevertheless, at a certain threshold, the incremental increase in the number of clusters will diminish substantially, resulting in a noticeable inflection point on the graph. The earliest clusters will introduce a substantial quantity of data; nonetheless, subsequent to this stage, the angle will be furnished. The ideal value of the "k" parameter, which determines the number of clusters, can be determined using the elbow criterion [61]. The initial stage of the procedure will entail assigning the variable k a value of 2, and future stages will involve incrementing k by one throughout each iteration. The computation of the sum of squared errors (SSE) can be achieved by utilizing the constant k. The subsequent procedure involves commencing the process of generating a line chart that depicts the sum of squared errors (SSE) for every distinct value of the signal parameter, k. When the line chart reaches a certain point, an elbow is observed, indicating the optimal value of k, which represents the number of clusters that may be obtained using the K-means method [28]. Once the line chart reaches a certain threshold, it transitions into an elbow shape. Equation 4 in the context of the

Elbow approach represents the within-group sum of squares (WSS). This metric quantifies the squared average distance between all data points within a cluster and the centroid of that cluster. It is a statistical calculation that measures the distance from the group mean to the corresponding cluster centroid. The distance can be determined by going in the opposite direction from the group mean towards the cluster centroid.

```
ElbowMethod(data, k_range):
    # data: The dataset to be clustered
    # k_range: A range of values for k to evaluate

    wcss_values = []  # To store the within-cluster sum of squares (WCSS) for each k

    for k in k_range:
        # 1. Run the clustering algorithm with k clusters
        clusters = RunClusteringAlgorithm(data, k)

        # 2. Calculate the WCSS for the current clustering result
        wcss = CalculateWCSS(clusters)

        # 3. Append the WCSS value to the list
        wcss_values.append(wcss)

    # 4. Determine the optimal k using the "elbow point" criterion
    optimal_k = FindElbowPoint(wcss_values, k_range)

    return optimal_k

FindElbowPoint(wcss_values, k_range):
    # wcss_values: List of WCSS values for different values of k
    # k_range: The range of k values

    # 1. Calculate the differences between consecutive WCSS values
    differences = [wcss_values[i] - wcss_values[i-1] for i in range(1, len(wcss_values))]

    # 2. Find the index of the "elbow point"
    elbow_index = IndexOfLargestDifference(differences)

    # 3. Determine the corresponding k value
    optimal_k = k_range[elbow_index]

    return optimal_k

IndexOfLargestDifference(differences):
    # differences: List of differences between consecutive WCSS values

    # 1. Find the index of the largest difference
    max_difference = max(differences)
    index = differences.index(max_difference)

    return index
```

Figure. 26 Elbow method algorithm pseudocode.

6.8 Results and discussions

Motivated by software deception, The utilization of the Machine Learning methodology has been implemented to partition extensive data collected from electric power grids' s sensors in the network into smaller, more manageable units referred to as "clusters." These clusters are responsible for executing data aggregation tasks on physical datasets that are processed by the main operating system. In order to conduct our analysis inconspicuously, we will observe the system intrusively by diverting our analysis to a decoy thread rather than the main process. The objective of this research is to periodically generate a replica of the main process as a decoy thread. Consequently, the results and discussion section will be divided into two distinct parts.

The practicality of my work is evident both prior to and after the establishment of the decoy thread. Prior to the creation of the decoy thread, it is imperative to initiate a trigger for this project. It is indeed accurate that we intend to replicate the decoy thread at regular intervals, such as every two minutes. If the main thread becomes infected within this two-minute timeframe, the malware will not be present in the decoy thread. Consequently, any cyber-deception probing occurring within the decoy thread would be rendered futile and unproductive. The occurrence of an anomaly serves as an opportunity to initiate the creation of a decoy thread in the present moment. In order to mitigate any damage and safeguard our network, it is imperative to promptly address the presence of malware. The millisecond plays a crucial role in this particular scenario.

Furthermore, considering this cognitive aspect, the actors involved in this scenario possess knowledge about cyber deception and are cautious enough not to fall for decoys easily. They emulate the behavior of existing malware by attempting to send their own

probes in order to determine if they are operating within a decoy thread or not. During this process, they can use different computational methods than the protection algorithm in order to avoid any contact with the decoy, thus minimizing the risk of immediate detection. By observing the departure of clusters in comparison to the modeled behavior, we can obtain a certain level of confidence that this is indeed a suspicious case.

In the optimal scenario for a machine learning model, the elbow graph will exhibit a consistent and smooth trend, indicating that there are no significant changes in the physical data as it remains inside the established threshold.

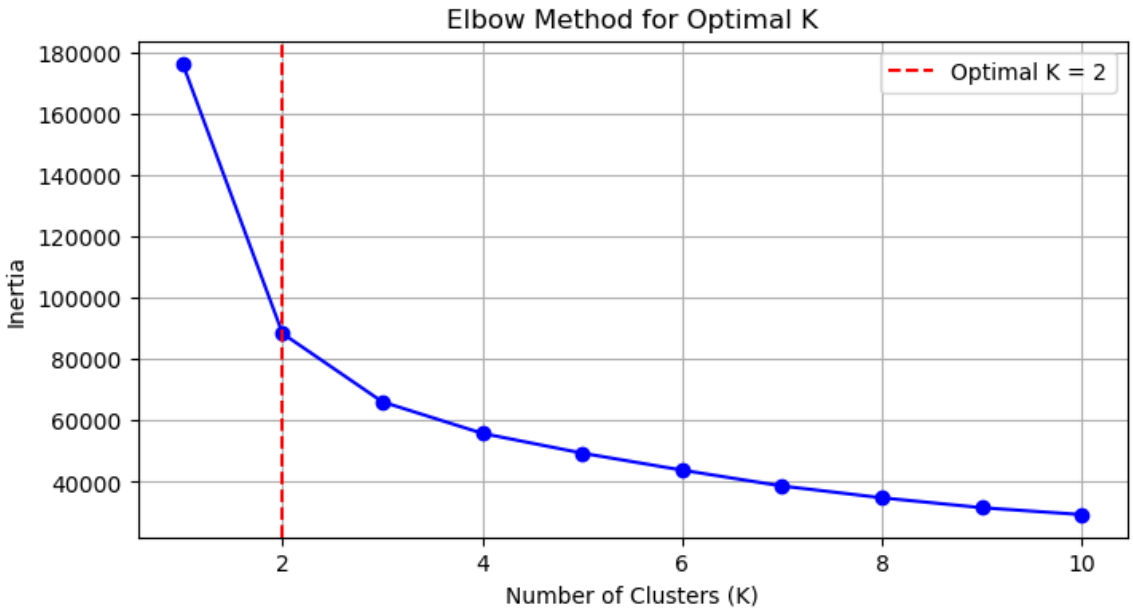

Figure. 27 Elbow method before malware resides in memory.

Figure 28 displays the plot illustrating in the elbow approach, which exhibits a consistent pattern. Prior to the loading of malware into memory, the number of clusters was observed to be 2. The within-cluster sum of squares (WSS) or inertia is around 9000.

Inertia denotes the cumulative squared distances between data points and the cluster centroids they are given in a K-means clustering algorithm.

In this scenario, the physical data obtained through sensors is transmitted to the transformer. Over time, the consistency of the data improves, and all measurements fall within the predefined threshold. Consequently, we can ascertain that this represents a legitimate case. As a result, the model will continue to operate and concurrently assess the physical dataset generated by the sensors.

As soon as we begin to see anomalies in the patterns of the datasets in Figure 28, and physic data begin approaching the threshold, and there are some changes in the inertia, then this is what we refer to as a suspicious case, and in this circumstance, we will make a duplicate of the decoy thread.

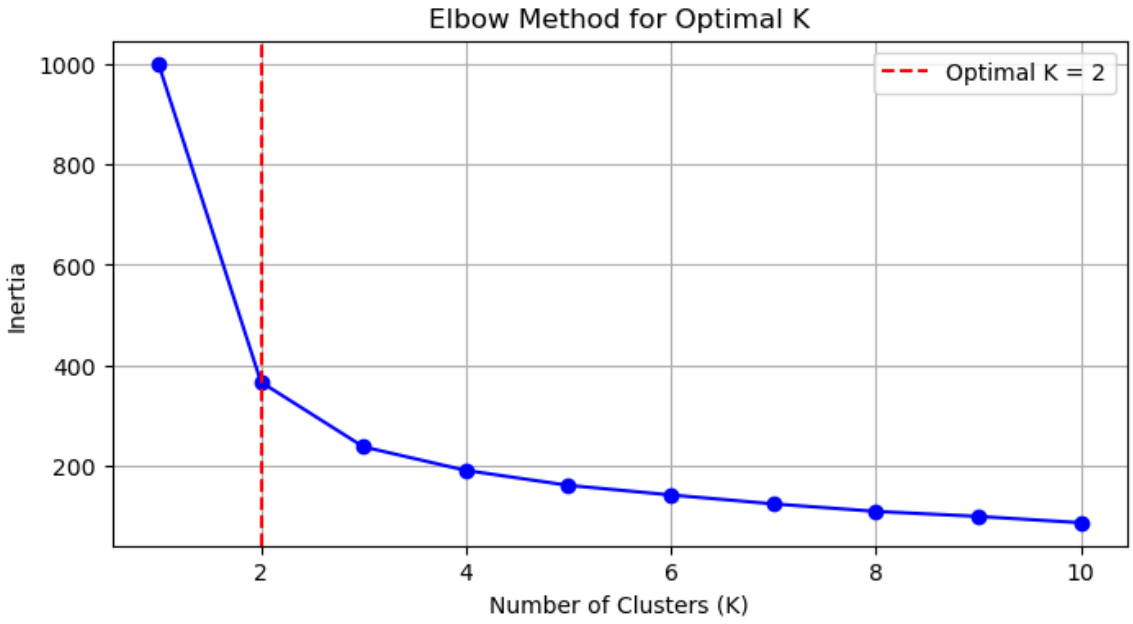

Figure. 28 Elbow method after malware reside in memory.

The plot depicted in Figure 29 demonstrates the use of the elbow method, showcasing a persistent pattern. After the injection of malicious software into the computer's memory, it is noted that there are two clusters present. The value of the within-cluster sum of squares (WSS) or inertia is approximately 400.

In essence, the proposed method involves the creation of a duplicate representation of the process, which can be conceptualized as a thread referred to as a "decoy thread" or "shadow thread." This shadow thread can then be wired with simulated sensors and actuators, serving to mimic the behavior of the original process.

In this context, our primary objective is to establish a suitable host environment for the potential malware that operates within the operating system (OS) process. Consequently, we are able to closely monitor and analyze the activities in a manner that may be considered intrusive.

The replication being discussed here accurately mimics the original process without interfering with its operations. Therefore, any analysis or interventions performed on this replica will not negatively impact the functioning of the actual process. It is important to note that the replica serves as a decoy, and any actions taken on it will not result in any dysfunction for the real process. On the contrary, the creation of a decoy thread, from the perspective of potential malware within the operating system (OS) process, does not result in any changes to the malware instruction. The decoy thread operates within the same environment as the main OS process without directly targeting the original malware structure. Instead, the focus is on intrusively targeting a replicated copy of the malware structure within the decoy.

When this malware attempts to execute and access the physical data structure, it may potentially read the buffer that contains sensor data. Furthermore, it endeavors to manipulate the circuit breaker. It is important to note that both the sensor data and the circuit breaker are emulated by software. The replication of this approach will not have any adverse impact on the functioning of the original process.

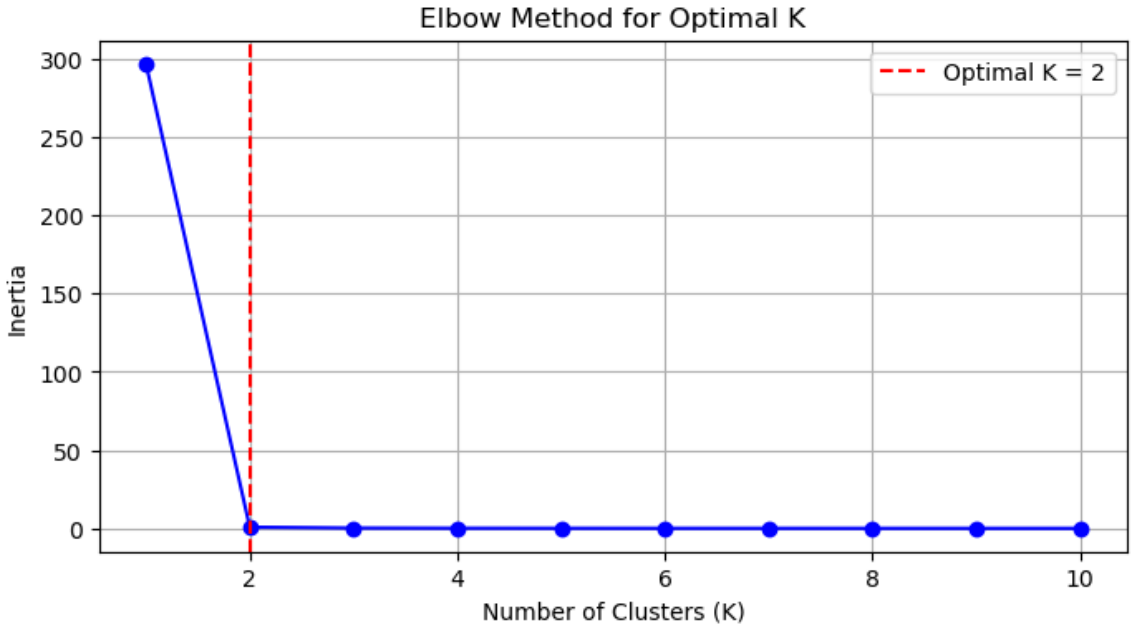

Figure. 29 Elbow method while malware manipulates sensor data.

Considering the identical k value of 2 in both Figure 27 and Figure 28, a significant disparity exists in the inertia of these two figures. The inertia refers to the sum of the squared distances between the data points and their respective cluster centroids within a K-means clustering solution. The disparity becomes evident upon comparing Figure 27 with Figure 28. The malicious software will continue to alter the actual data, and the proposed system will proceed to generate new charts, as can be seen from looking at figure 28. When this occurs, it is the ideal time for investigating and analyzing

the behaviors of malware without interfering with the performance of the main operating system process. This can be accomplished by performing the investigation and study in a context free from interference. This is because we require the primary process to be working normally for the operating system process to be in a position to take necessary action in a timely manner. The duration of time is measured in milliseconds, which are extremely short intervals.

6.9 Exploring K-means++ and SVM for Detection Precision

K-means++ and Support Vector Machines (SVM) serve distinct objectives, where K-means++ is utilized as a clustering technique and SVM is employed for classification or regression tasks. Nevertheless, within the framework of anomaly detection, both methods can be modified to serve this objective. I will now present a concise comparison specifically within this context.

6.9.1 K-means++ for Anomaly Detection

Anomaly identification can be performed using the K-means++ algorithm by identifying instances that are far away from the cluster centroids as probable abnormalities. These remote spots can be recognized as anomalies. An anomaly score can be derived by measuring the distance between a data point and its associated cluster centroid. Points that exceed a specific threshold may be identified as anomalies. Its main advantages is that the implementation is straightforward; it is effective in terms of CPU resources when dealing with huge datasets, and it is capable of detecting local anomalies inside clusters. Its main drawback is its dependency on the initial selection of cluster centers, since it assumes clusters are spherical and have equal sizes.

6.9.2 SVM for Anomaly Detection.

The one-class SVM form of SVM is highly suitable for detecting anomalies. During the training phase, a model is constructed using normal cases. Subsequently, during the testing phase, any departures from this established norm are identified as anomalies. SVM establishes a hyperplane that encompasses the regular instances, while examples falling on the other side of the hyperplane are regarded as anomalies. Some of its advantages included that it demonstrates efficacy in spaces with a large number of dimensions, it exhibits resilience to atypical data points, and has the capability to accommodate non-linear associations using kernel functions. Its main drawbacks are that selecting the appropriate kernel and parameters can pose a difficulty, and the algorithm may require a significant amount of memory when dealing with huge datasets.

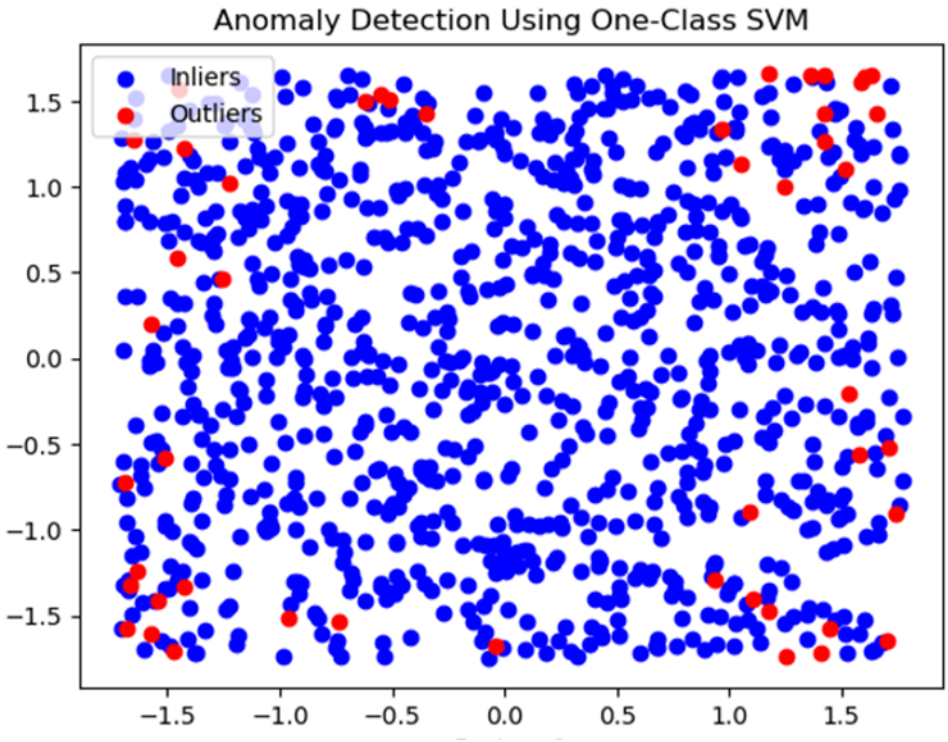

Figure. 30 Anomaly detection using SVM

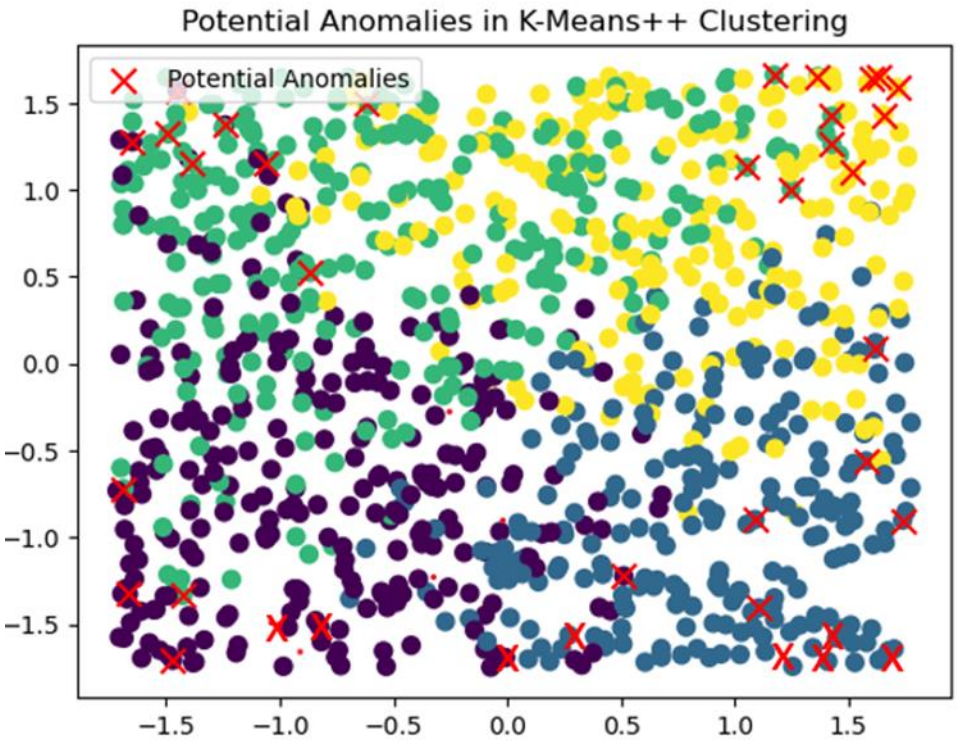

Figure. 31 Anomaly detection using k-means++ model.

The captivating storyline is demonstrated through the comparative visualizations of the SVM and K-means++ models, as depicted in Figures 31 and 32. Figure 31 illustrates the output of the Support Vector Machine (SVM), showing clear decision boundaries that demonstrate how the classifier distinguishes between classes. Simultaneously, Figure 32, illustrating the outcomes of the K-means++ clustering method, exhibits a structure that closely resembles the one presented in Figure 31. This exemplifies the model's ability to identify inherent patterns within the data.

The parallelism observed in the visual representations of the SVM and K-means++ outputs is not coincidental, but rather a visual manifestation of the efficacy of the K-means++ clustering technique. The efficacy and resilience of K-means++ in uncovering

the inherent data structures is evidenced by the alignment of the clustering boundaries with the classification contours depicted in the SVM figure. The concurrence between the output of the K-means++ model and the established classification framework offered by SVM, as demonstrated by the graphical evidence, reinforces the idea that the K-means++ model is highly efficient.

6.10 Future Research:

The potential of future research in the domains of deep learning and k-Means++ hybrid models holds considerable significance as it offers a viable avenue for addressing intricate challenges across diverse fields. The potential for conducting such a study is very exhilarating. The identification of anomalies and outliers holds significant importance within the domains of data analysis and machine learning. The methods stated above find utility in diverse domains, encompassing information security, finance, healthcare, industrial processes, and other related sectors. There are several key factors that require significant attention in relation to these subjects:

6.10.1 Unsupervised Learning Utilizing Hybrid Architectures:

This study aims to explore the feasibility of constructing advanced hybrid models that seamlessly incorporate deep learning with k-means++ clustering techniques. This encompasses architectural concepts that adapt the overall number of clusters based on the level of complexity present in the data.

6.10.2 Transfer learning and pre-trained models:

This study aims to investigate the potential of integrating pre-trained deep learning models, specifically transformers, into hybrid models to facilitate feature extraction. This

approach has the potential to facilitate the transfer of knowledge and improve the generalization capabilities of models.

6.10.3 Anomaly Detection and Outlier Identification:

Hybrid models are employed for the purpose of anomaly detection, aiming primarily to promptly identify outliers and unforeseen data points across diverse domains such as network security, healthcare, and fraud detection, among others.

REFERENCES

[1] Arthur, D., & Vassilvitskii, S. (2007, January). K-means++ the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035).

[2] Tonge, A. M., Kasture, S. S., & Chaudhari, S. R. (2013). Cyber security: challenges for society-literature review. IOSR Journal of computer Engineering, 2(12), 67-75.

[3] Morales, E. F., & Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In Biosignal processing and classification using computational learning and intelligence (pp. 111-129). Academic Press.

[4] Dhanaraj, R. K., Rajkumar, K., & Hariharan, U. (2020). Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning. Business Intelligence for Enterprise Internet of Things, 55-79.

[5] Haldorai, A., Ramu, A., & Suriya, M. (2020). Organization internet of things (IoTs): Supervised, unsupervised, and reinforcement learning. In Business Intelligence for Enterprise Internet of Things (pp. 27-53). Cham: Springer International Publishing.

[6] Mutti, M., Mancassola, M., & Restelli, M. (2022, June). Unsupervised reinforcement learning in multiple environments. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 7, pp. 7850-7858).

[7] R. M. Esteves, T. Hacker and C. Rong, "Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 17-24, doi: 10.1109/CloudCom.2013.89.

[8] Reynolds, A.P., Richards, G., de la Iglesia, B. et al. Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms. J Math

[9] Marutho, D., Handaka, S. H., & Wijaya, E. (2018, September). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In 2018 International Seminar on Application for Technology of Information and Communication (pp. 533-538). IEEE.

[10] Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018, April). Integration K-means clustering method and elbow method for identification of the best customer profile cluster. In IOP Conference Series: Materials Science and Engineering (Vol. 336, No. 1, p. 012017). IOP Publishing.

[11] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 105(9).

[12] Münz, G., Li, S., & Carle, G. (2007, September). Traffic anomaly detection using k-means clustering. In Gi/itg workshop mmbnet (Vol. 7, No. 9).

[13] Kumari, R., Singh, M. K., Jha, R., & Singh, N. K. (2016, March). Anomaly detection in network traffic using K-mean clustering. In 2016 3rd international conference on recent advances in information technology (RAIT) (pp. 387-393). IEEE.

[14] Wang, Z., Zhou, Y., & Li, G. (2020, May). Anomaly detection by using streaming K-means and batch K-means. In 2020 5th IEEE international conference on big data analytics (ICBDA) (pp. 11-17). IEEE.

[15] Paea, S., & Baird, R. (2018). Information Architecture (IA): Using multidimensional scaling (MDS) and K-Means clustering algorithm for analysis of card sorting data. Journal of Usability Studies, 13(3), 138-157.

[16] T. Omar, M. Zohdy and J. Rrushi, (2021). Clustering Application for Data-Driven Prediction of Health Insurance Premiums for People of Different Ages. In 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2021, pp. 1-6, doi: 10.1109/ICCE50685.2021.9427598.

[17] Kurita, T. (2019). Principal component analysis (PCA). Computer Vision: A Reference Guide, 1-4.

[18] Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.

[19] Uddin, M. P., Mamun, M. A., Afjal, M. I., & Hossain, M. A. (2021). Information-theoretic feature selection with segmentation-based folded principal component analysis (PCA) for hyperspectral image classification. International Journal of Remote Sensing, 42(1), 286-321.

[20] Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. Journal of Soft Computing and Data Mining, 2(1), 20-30.

[21] Kim, M. K., Chang, J. W., Park, K., & Yang, D. R. (2022). Comprehensive assessment of the effects of operating conditions on membrane intrinsic parameters of forward osmosis (FO) based on principal component analysis (PCA). Journal of Membrane Science, 641, 119909.

[22] Reddy, G. N., & Reddy, G. J. (2014). A study of cyber security challenges and its emerging trends on latest technologies. arXiv preprint arXiv:1402.1842.

[23] R. M. Esteves, T. Hacker and C. Rong. (2012). Cluster analysis for the cloud: Parallel competitive fitness and parallel K-means++ for large dataset analysis. 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, Taipei, Taiwan, 2012, pp. 177-184, doi: 10.1109/CloudCom.2012.6427553.

[24] Q. Qiu, Q. Zhang and K. Guo, "Grey Kmeans algorithm and its application to the analysis of regional competitive ability," 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, 2014, pp. 249-253.

[25] Putchala, S. (2020, April). Distributed Transformer Monitoring, Version 1. Retrieved April 14, 2020, from https://www.kaggle.com/datasets/sreshta140/ai-transformer-monitoring?rvi=1.

[26] Chang, C. W., Chang, C. Y., & Lin, Y. Y. (2022). A hybrid CNN and LSTM-based deep learning model for abnormal behavior detection. Multimedia Tools and Applications, 81(9), 11825-11843.

[27] D. Charalampidis, "A modified k-means algorithm for circular invariant clustering," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 12, pp. 1856-1865, Dec. 2005, doi: 10.1109/TPAMI.2005.230.

[28] R. M. Esteves, T. Hacker and C. Rong, "Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 17-24, doi: 10.1109/CloudCom.2013.89.

[29] Kaggle: Machine Learning and Data Science Community. (n.d.). https://www.kaggle.com

[30] Tanner, A., Dancer, F. C., Hall, J., Parker, N., Bishop, R., & McBride, T. (2022, December). The Need for Proactive Digital Forensics in Addressing Critical Infrastructure Cyber Attacks. In 2022 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 976-982). IEEE.

[31] Lehto, M. (2022). Cyber-attacks against critical infrastructure. In Cyber Security: Critical Infrastructure Protection (pp. 3-42). Cham: Springer International Publishing.

[32] Kavan, Š., & Skalická, M. Z. F. (2022, June). Security of critical information infrastructure and possible disruption as a crisis. In 2022 11th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5). IEEE.

[33] Tsang, Rose, Cyberthreats, Vulnerabilities, and Attacks on SCADA Networks, 2010.

[34] S. Garfinkel, "Anti-forensics: Techniques, detection, and countermeasures," 2nd International Conference on i-Warfare and Security, 2007, p. 77.

[35] S. L. Garfinkel, "Digital forensics research: The next 10 years," Digital Investigation, vol. 7, pp. S64- S73, 2010.

[36] Internet Crime Complaint Center (IC3) | Home Page. (n.d.).Retrieved October 22, 2023, from https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf

[37] Gallo, K. (2023, April 4). What Is LDAP (Lightweight Directory Access Protocol)? Built In. https://builtin.com/data-science/ldap

[38] Sutton, S., & Rrushi, J. (2023). Decoy Processes with Optimal Performance Fingerprints. IEEE Access.

[39] Park, Y., & Stolfo, S. J. (2012, May). Software decoys for insider threat. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (pp. 93-94).

[40] Lee, J., Lee, J., & Hong, J. (2017, September). How to make efficient decoy files for ransomware detection? In Proceedings of the International Conference on Research in Adaptive and Convergent Systems (pp. 208-212).

[41] Marini, M., Ansani, A., & Paglieri, F. (2020). Attraction comes from many sources: Attentional and comparative processes in decoy effects. Judgment and decision making, 15(5), 704-726.

[42] Franzen, F., Steger, L., Zirngibl, J., & Sattler, P. (2022, June). Looking for honey once again: Detecting RDP and SMB honeypots on the Internet. In 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 266-277). IEEE.

[43] Cui, Y. G., Kim, S. S., & Kim, J. (2021). Impact of preciseness of price presentation on the magnitude of compromise and decoy effects. Journal of Business Research, 132, 641-652.

[44] Wu, C., & Cosguner, K. (2020). Profiting from the decoy effect: A case study of an online diamond retailer. Marketing Science, 39(5), 974-995.

[45] Li, T., Higgins, J. P., & Deeks, J. J. (2019). Collecting data. Cochrane handbook for systematic reviews of interventions, 109-141.

[46] Sun, J., Sun, K., & Li, Q. (2020, June). Towards a Believable Decoy System: Replaying Network Activities from Real System. In 2020 IEEE Conference on Communications and Network Security (CNS) (pp. 1-9). IEEE.

[47] Sethi, T., & Mathew, R. (2021, February). A study on advancement in honeypot based network security model. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 94-97). IEEE.

[48] A. K. Jain, M. N. Murty, and P. J. Flynn. “Data clustering: a review”, ACM Comput. Surv.1999.

[49] Reynolds, A.P., Richards, G., de la Iglesia, B. et al. Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms. J Math Model Algor 5, 475–504 (2006). https://doi.org/10.1007/s10852-005-9022-1

[50] S. Singh and A. Mishra, "Clustering analysis for large scale data sets," International Conference on Computing, Communication & Automation, Noida, 2015, pp. 1-4, doi: 10.1109/CCAA.2015.7148353.

[51] Jasser, J., Ming, H., & Zohdy, M. A. (2017, July). Situation-Awareness in Action: An Intelligent Online Learning Platform (IOLP). In International Conference on Human-Computer Interaction (pp. 319-330). Springer, Cham.

[52] Gao, C., Sun, H., Wang, T., Tang, M., Bohnen, N. I., Müller, M. L., & Dauer, W. (2018). Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in Parkinson's disease. Scientific reports, 8(1), 1-21.

[53] Marutho, D., Handaka, S. H., & Wijaya, E. (2018, September). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In 2018 International Seminar on Application for Technology of Information and Communication (pp. 533-538). IEEE.

[54] Sharma, M., Purohit, G. N., & Mukherjee, S. (2018). Information retrieves from brain MRI images for tumor detection using hybrid technique K-means and artificial neural network (KMANN). In Networking communication and data knowledge engineering (pp. 145-157). Springer, Singapore.

[55] Kumar, J., & Vashistha, R. (2017, February). Estimation of inter-centroid distance quality in data clustering problem using hybridized K-means algorithm. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-7). IEEE.

[56] Yedla, M., Rao, S.S., Pathakota, & Srinivasa, T. (2010). Enhancing K-means Clustering Algorithm with Improved Initial Center.

[57] Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017, August). Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3861-3870). JMLR. org.

[58] Marutho, D., Handaka, S. H., & Wijaya, E. (2018, September). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In 2018 International Seminar on Application for Technology of Information and Communication (pp. 533-538). IEEE.

[59] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 105(9).

[60] Nguyen, Hoang, et al. "A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical K-means clustering and cubist algorithms." Applied Soft Computing 77 (2019): 376-386.

[61] Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018, April). Integration K-means clustering method and elbow method for identification of the best customer profile cluster. In IOP Conference Series: Materials Science and Engineering (Vol. 336, No. 1, p. 012017). IOP Publishing.