Machine-Learning-based User Behavior Classification for Improving Security Awareness Provision

Alaa Al-Mashhour, Dr.Areej Alhogail
Department of Information Systems-College of Computer and Information Science, King Saud University Riyadh, Saudi Arabia

***Abstract*—Users of information technology are regarded as essential components of information security. Users' lack of cybersecurity awareness can result in external and internal security attacks and threats in any organization that has several users or employees. Although various security methods have been designed to protect organizations from external intrusions and attacks, the human factor is also essential because security risks by "insiders" can occur due to a lack of awareness. Therefore, instead of general nontargeted security training, comprehensive cybersecurity awareness should be provided based on employees' online behavior. This study seeks to provide a machine-learning-based model that provides user behavior analysis in which organizations can profile their employees by analyzing their online behavior to classify them into different classes and, thus, help provide them with appropriate awareness sessions and training. The model proposed in this paper will be evaluated and assessed through its implementation on a sample dataset that reflects users' online activities over a specific period to measure the model's accuracy and effectiveness. A comparison between six classification techniques has been made, and random forest classification had the best performance regarding classification accuracy and performance time. After users are classified, each group can be provided with the appropriate training material. This study will stimulate additional research in this area, which has not been widely investigated, and it will provide a useful point of reference for other studies. Additionally, it should provide insightful information to help decision-makers in organizations provide necessary and effective security awareness.**

***Keywords*—Machine learning; user behavior analysis; cybersecurity; classification; security awareness**

I. Introduction

The internet plays a significant role in many aspects of our lives, and many daily tasks have been digitalized and are required to be completed online. Besides this, the number of users and employees with varying levels of security knowledge and different backgrounds who are required to work online has increased, which has, in turn, influenced organizations' security requirements. Because of this, every organization now has internal cybersecurity and data and asset safety as a priority. Organizations that handle sensitive information assets can operate effectively, locally, and globally, exchanging information quickly and seamlessly among their employees, partners, suppliers, and customers. Indeed, many organizations now rely on online information exchange to keep their operations running smoothly in collaboration with other parties. However, confidential information is becoming increasingly vulnerable to internal and external security attacks [1]. Although hardware and software-based technologies have been implemented, such as firewalls, proxy servers, and antivirus software, these solutions have not significantly reduced security attacks.

Security attacks or breaches, when they are carried out successfully in organizations, affect inside assets or data. However, the consequences are frequently financial and reputational, undermining customer trust. Applying technical control and systems in this regard is essential. Still, technical controls are only the first line of defense in cybersecurity, and they cannot prevent insiders with elevated access from violating security policies. Many previous studies in this field have discussed the human factor in cybersecurity and the significant role that employees can play in information security breaches. This has increased organizational focus on human threats [2,3].

As a result, many organizations have started to provide cybersecurity awareness training to their employees to make them conscious of cybersecurity threats or any other related issues. Awareness sessions and training are critical to ensuring that staff members act responsibly and are aware of the potential consequences of their online behavior [4]. Due to the importance of cybersecurity awareness inside an organization, various studies have reported that they can become considerably more secure against both internal and external security threats with improved security awareness programs [5–7]. Ryu et al. [8] outlined that a strong awareness program is essential to guarantee that employees properly comprehend their respective internet technology (IT) security duties and roles to safeguard the IT resources delegated to them. Therefore, to reach this level of awareness and responsibility in this regard, awareness sessions on cybersecurity's importance are vital to ensuring the enhancement of the security culture within an organization.

Many employers provide cybersecurity awareness sessions and frequently send out relevant material and emails, as will be viewed in Section II. Nevertheless, these conventional methods are ineffective because tailored and targeted security awareness materials based on the needs and knowledge of employees is required as the level of awareness varies greatly among employees.

This study proposes a machine–learning-based model that enables organizations to analyze users' online behavior, activities, and actions to target them with appropriate security awareness materials. First, we will investigate six machine learning (ML) classification models to select the most appropriate classifier based on the performance measurements and how accurate each classifier is in forming each user class. We will go through several phases to train and test the models. Additionally, for added validation, we will conduct a cross-validation test to ensure accurate results.

Furthermore, based on the comparison results obtained through the performance calculation using confusion matrix, accuracy, F1, and other measures, including performance time, the best classification technique will be used to classify users into three classes and subsequently target them with suitable awareness sessions. The user classes are the malicious, suspicious, and normal (which require the fewest targeted awareness sessions) behavior classes.

Users' online behavior can reveal much about their knowledge level about cybersecurity and what type of security threats they may cause for their organization, as well as what type of security awareness training must be provided to them. Therefore, we used a dataset, which will be discussed later, that consists of web links that users have visited to show their web-behavior. After user classification, the organization can choose the suitable cyber security awareness materials and session content for each user's class, and it will be saved for subsequent users in the backend database to be sent again by the machine to each particular class without any human interaction.

The proposed model can be implemented as a plug-in for the security operations center dashboard. Therefore, in addition to having the ability to monitor network traffic, endpoints, logs, and security events, the organization will also be able to classify its employees into specific classes to send them classified training materials and take the required action in this regard. In addition, these classes can benefit decision-makers in assessing the organization's weaknesses regarding employees' behavior to define new awareness strategies, IT usage policies, and, if required, new tasks and responsibilities.

The remainder of this paper is organized as follows: The literature review will take place in Section II. Section III comprises the proposed methodology. Section IV presents the result, then a discussion and comparison of the results achieved in Section V, followed by the conclusion Section VI, which concludes the proposed model and presents directions for future work.

II. Literature Review

The use of Internet technology (IT) has increased dramatically since its advent. The rapid increase in internet traffic has led researchers to consider the significance of cybersecurity, and research on the values and methods of cybersecurity awareness has attracted substantial attention. Nevertheless, only a few studies have been conducted on the use of machine learning in cybersecurity awareness. This section covers background knowledge and related work regarding the proposed method.

As we are looking to enhance the user awareness level due to its importance, In fact, traditional training methods, such as classroom discussions and exercises, have demonstrated their efficacy in increasing trainee awareness and, consequently, their ability to detect issues such as phishing or hacking attempts [9]. However, due to the high cost and number of trainees, traditional class sessions are rendered insufficient and cannot provide the information that individual employees need. Bernaschina et al. [10] studied some security training sessions that concentrated on phishing emails. At the end of each session, the trainees were given a survey to complete the evaluation of usefulness of the previous session as a learning opportunity. These trainees reported that they already had prior knowledge of phishing emails, which demonstrates that nontargeted sessions that are not based on specific behavior lead to the wastage of both time and money, as well as a reduction in benefits for the organization and its employees. Therefore, targeted sessions based on behavior analysis must be created.

Crume et al. [9] found that targeted employee awareness programs based on web behavior can aid in preventing the misuse of an organization's assets. Furthermore, implementing this training will result in numerous benefits for organizations, including improved resource utilization, employee knowledge and performance, and organizational policies and procedures.

Current research on awareness has tended to focus on analyzing users' behaviors based on qualitative data collected through interviews, scales, questionnaires, and surveys. User behavior analysis related to cybersecurity awareness, however, focuses on analyzing users' activities, such as accessing websites and files and user identity. User behavior analysis has successfully identified usage patterns that may indicate unusual or anomalous internet behavior.

A study carried out by Gartner has been mentioned in the work of Kumar and Singh [12] that defined user behavior analysis as outlining and incongruity recognition, which depends on a variety of analytic methodologies, typically combining fundamental analytical methods. Examples of this are policies that influence signatures, pattern recognition, mapping, basic rules of statistics, and advanced analytics tools. However, these methods do not provide accurate data regarding users' real online behavior.

As shown in some of the previous research on the impact of online behavior, this emphasizes the need for organizations to target their employees with specific awareness sessions based on an analysis of those behaviors. Targeted security awareness refers to the provision of training based on the threat that some employees' online behavior may pose. These employees can be identified using behavioral analysis of each user within an organization, using a range of qualitative and quantitative data. Multiple scales are used to assess employee awareness. For example, a Portuguese healthcare institution case study assessed employees' professional awareness of information security by assessing their attitudes and behavior related to cybersecurity [13]. The study consisted of applying and validating scales, such as the risky cybersecurity behaviors (RScB) scale, which is a questionnaire for employees that evaluates behaviors that may lead to poor cybersecurity practices and human vulnerability within enterprises, particularly in healthcare organizations. The RScB scale has a score range of 0 to 120, with higher values indicating riskier behavior, which is frequently associated with a lack of cybersecurity awareness.

Moreover, in this regard, several machine-learning techniques, such as sequence clustering, can be used to analyze and study user behavior, such as grouping web users with common interests and behaviors. For example, clustering analysis creates a user cluster from web log files. For instance, Facebook's machine-learning algorithms track every user's activity on the network to predict their interests, recommend articles, and post notifications on the news feed based on the user's previous behavior [14].

Fong Tsai showed [16] how collaborative filtering recommendations, which are widely used in recommendation systems on shopping websites, form cluster ensembles. This assumes that people who share the same preferences on certain items also tend to share the same choices on other items. Therefore, clustering based on user logs is done to identify users with similar choices, and it provides recommendations based on the preferences of these "similar neighbors."

Jiang et al. [17] demonstrated that different machine-learning techniques can be used to extract meaningful data from a huge dataset, including extracting information to analyze user behavior. Callara and Wira [15] suggested an algorithm for user classification based on their dataset and found that it could distinguish 108 groups of users with similar online behavior, which meant they could classify each group with similar behavior as a separate group. They proved that classification techniques are useful in analyzing and labeling test data into known types of classes. Hence, employers can benefit from this classification by providing awareness sessions suited to each class to enhance their employee's level of security knowledge and keep their assets safe.

Efficient classification techniques have been used by Niranjan and Nitish [11] to enable users to distinguish between phishing and normal websites, classify users as normal users or criminals based on their social media activities (crime profiling), and prevent users from running malicious code by labeling them as "malicious." However, classifying users into two categories only offers limited options. Concerning the provision of security awareness sessions, a larger number of categories is needed to be more accurate and provide what is needed based on user experience and behavior.

III. Methodology

The user classification model is a multi-classification problem that aims to classify users into three classes based on the analysis results of their online behavior. To achieve the desired goal of classifying users based on their online behavior and delivering dedicated awareness material to them, a machine–learning-based classification model has been proposed. Assume D, a dataset of website instances, where domain di is defined using a set of n features, $F = \{f1, f2, \ldots, fn\}$, and each domain $di \in D$ is either malicious, suspicious, or normal behavior. The supervised machine-learning algorithm must be trained using D so that the resulting model M can classify a new domain *dnew* that has not been seen before by *M*.

The research process has three main phases. The data are collected from users' records and then prepared using data cleaning and preprocessing. Subsequently, the researchers take various steps to evaluate the classification methods to construct the most effective model of user behavior classification. A diagram describing the workflow of the research procedure is shown in Fig. 1.

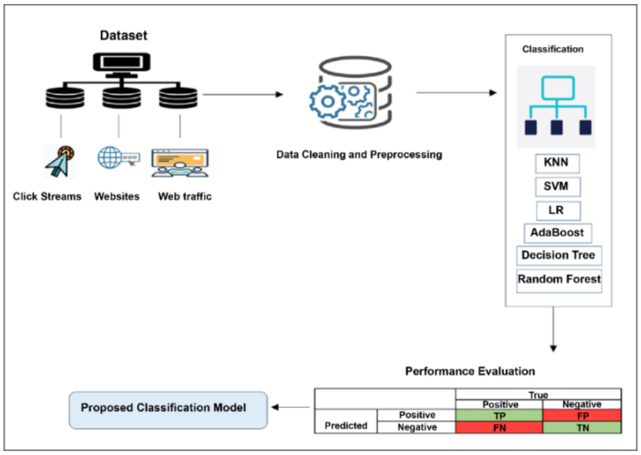

Fig. 1. Security awareness provision based on the user behavior model framework.

A. Data Description

The dataset that was used in this study is from Irvine's Machine Learning Repository of the University of California [19]. The same dataset was used to investigate and validate the observations. It is an imbalanced multivariate dataset by nature, which has 8,118 instances, each with nine integral forms of attribute characteristics. The data are classified into three user classes to help provide suitable security awareness sessions. These data contain website references/sources which are legitimate or malicious besides the normal references.

Therefore, the dataset contains 8,118 website instances, of which 4,602 are authentic, 2,670 are malicious, and 846 are suspicious. The nine distinct features provided in the dataset that can be used to classify any website as malicious or authentic are server form handlers (SFH), popup window, SSL final state, request URL, URL anchor, web traffic, URL length, domain age, IP address, and the labeled class. They are briefly described in Table I.

B. Model Description

Six well-known classifiers were compared in a supervised learning environment with prior knowledge of the output target set. The classifiers were K nearest neighbor (KNN), support vector machine (SVM), logistic regression (LR), adaptive boosting (AdaBoost) classifier, decision tree classifier, and the random forest classifier. Chosen algorithms have been selected as they are commonly used by researchers and in practice for user classification in different fields, as in the work of Kotsiantis *et al.*, [42], Osisanwo *et al.*,[43], and other studies mentioned in this work [11,15,16,17,34,34]. They are described in the following subsections.

TABLE I. DATASET ATTRIBUTES DESCRIPTION

Feature	*Description*
SFH	SFHs that contain an empty string or “about: blank” are considered doubtful because action should be taken based on the submitted information. In addition, if the domain name in an SFH is different from the domain name of the webpage, then this reveals that the webpage is suspicious because the submitted information is rarely handled by external domains.
Popup window	This is considered a feature, particularly when the website is asking the users to submit any information through a popup window. It is unusual to find a webpage that requests personal information from users via a popup window.
SSLfinal stated	SSL is used to secure communication between a web browser and a web server. This turns a website’s address from HTTP to HTTPS. The “S” stands for “secure.”
Request URL	A request URL examines whether the external objects contained within a webpage, such as images, videos, and sounds, are loaded from another domain.
URL anchor	An anchor is an element defined by the <a> tag. This feature is treated exactly like a “Request URL.”
Website traffic	This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit.
URL length	A URL length can show whether a URL is a suspicious or phishing URL, where specific calculations should be made to determine whether it is a safe URL or a suspicious or phishing URL.
Domain age	Most phishing websites exist for only a short period.
IP address	If an IP address is used that is different from the domain name in the URL, such as “http://125.98.3.123/fake.html,” then someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code, as shown in the following link: “http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html.”
Class	This is the class of the domain (malicious behavior =–1, suspicious behavior = 0, and normal behavior = 1).

1) *The KNN classifier:* This is among the most basic classifiers. It works based on a supervised training method, and its technique is based on similarity. The KNN algorithm can perform regression and classification, and it is nonparametric by nature, as it does not make assumptions regarding non-available data. The basic principle is measuring the Euclidean distance from the new point to the nearest previous points, which are the KNNs. The class that has the nearest neighbors is assigned to the given query point.

2) *Support Vector Machine (SVM):* This machine-learning classification method uses supervised learning, and it is based on the margin or decision boundary, as the SVM selects the optimal margin for classification. However, in this research, we applied the SVM one-vs-Rest (OvR) method of multiclass classification, which was used to create a multiclass SVM classifier. Here, for each class, we created three OvR classifiers. Each classifier should predict a class probability, and the data will be assigned to the highest-probability class.

3) *Logistic Regression (LR):* This approach works based on the probabilistic prediction of any specified variable and performs the estimation of parameters related to the logistic model. We classified data into more than two classes. Therefore, we had y = {0,1 … n}. A one-vs-all strategy was used, in which we trained three distinct binary classifiers, each designed to recognize a specific class. Subsequently, we used these classifiers to predict the correct class.

4) *AdaBoost:* This approach can perform both classification and regression. The working mechanism is based on the meta-estimation and ensemble method. Through this method, weak learning is converted into stronger learning. In the beginning, it uses a basic learning method model and performs repetitive adjustments of the data distribution to increase the accuracy of the next model based on the existing model performance.

5) *Decision tree:* This is a tree-type classifier, and it has nodes, branches, and leaf nodes. The internal nodes are the dataset and features, whereas the branches represent the decision-making rules, and the leaf is the outcome. Thus, it is fundamentally a graphic illustration depicting all possible outcomes of a problem and its conditions. The classification and regression tree algorithm is used to form the tree structure. It is nonparametric by nature and classifies nonlinear data efficiently. It classifies each branch using the decision rules.

6) *Random forest:* The random forest approach is extremely efficient, and its training requires little time. Its accuracy and other performance measures are very high, even when datasets are large or contain missing data. It has parallel decision trees. Thus, it is a type of bagging ensemble. For the classification task, the output is considered the data found at the bottom of the node, while for the implementation of the regression, the mean of all the trees is considered the final output. Let the trees be denoted by h1 (x), h2 (x), … AdaBoost, hk (x); the training data are given as X, Y, and the margin function can be defined as the equation given below:

$$mg(X,Y) = av_k\, I(h_k(X) = y) – \max_{j \neq y} av_k\, I(h_k(X) = j) \quad (1)$$

The classification models are implemented on an unbalanced dataset. Each classifier is trained and tested on the dataset.

C. Feature Importance

Feature importance refers to techniques that calculate a score for all the input features for this model–the score represents the importance of each feature. In other words, it indicates strategies for valuing input features depending on their predictive power for a target variable (rank features based on their effect on the model’s prediction.). Feature importance is essential in the context of understanding the data that go into a model, model improvement, or model simplification, which means, in the case of reducing the model dimensionality, high-scoring features could be kept, and the features with the lowest scores could be deleted because they were not necessary.

Because of the points, feature importance scores are a critical component in predictive modeling, as they provide enlightenment of the data and the model. Let *D* be a dataset of *m* classes; a represents a feature that takes *V* possible values {*a1*, *a2*, … *av*} in *D*. Let *Dv* be the subset of samples from *D* that takes the value of *av* for feature *a*, and let *pi* be the probability that a sample belongs to class *i*. In the proposed model, the following measures are used: information gain, gain ratio, Gini index, and Pearson product—moment correlation coefficient. These measures have been chosen because they are easy to understand and execute, have light computational requirements, and are frequently successful with various datasets. They are described as follows [19, 20]:

1) Information gain: This metric identifies the features that provide the most information about a class and must highlight that entropy plays a crucial role in measuring information gain. Entropy measures the uncertainty of the data. From a different perspective, entropy measures how difficult it is to guess the label of a random sample from a dataset, where low entropy indicates that the data labels are quite uniform, and high entropy indicates that the labels are in confusion [21]. Information gain computes the difference between the entropy before and after a split and specifies class element impurity. The information gain metric investigates the information content of messages; the information gain can be determined by separating dataset D by features, as follows:

$$\text{Gain(D, a) = Ent(D)–}\sum_{v-1}^{V} \frac{|Dv\,|}{|D\,|} Ent(Dv) \quad (2)$$

where *Ent(D)* is the entropy. By dividing *D* based on feature *a*, a high information gain value indicates that the archived data is of greater purity.

2) Gain ratio: The gain ratio attempts to reduce the bias of information gain by introducing a normalizing term known as intrinsic information (II). II is the level of difficulty in guessing the branch in which a randomly selected sample is placed. The feature gain ratio is calculated as Gain ratio = information gain/II, which means mathematically:

$$\text{Gain\_Ratio (D, a)} = \frac{Gain\ (D,a)}{IV(a)}, \qquad (3)$$

where *IV(a)* denotes the intrinsic value of a feature *a* and is calculated as follows:

$$IV(a) = -\sum_{v-1}^{V} \frac{|Dv\,|}{|D\,|} \log 2\ \ \frac{|Dv\,|}{|D\,|} (4)$$

3) Gini Index: This is also known as Gini impurity and measures the degree or probability of a variable being incorrectly classified when randomly selected. It measures the dataset impurity. If all the elements in a class belong to a single class, then it can be called pure. In the calculation of impurity, the weight of the feature based on the class label has been calculated. The degree of the Gini index varies between 0 and 1, and a lower Gini index means a higher dataset purity [22]. It can be calculated as follows

$$Gini_{Index}(D,a) = \sum_{v-1}^{V} \frac{|Dv\,|}{|D\,|} Gini\ (Dv) \quad (5)$$

where

$$Gini\ (D) = 1 - \sum_{i=1}^{m} Pi^2 \qquad (6)$$

4) Mattheus correlation coefficient: Brian W. Mattheus developed the Mattheus correlation coefficient (MCC) in 1975 using Karl Pearson's phi coefficient, and it has become a widely used metric for evaluating the effectiveness of machine-learning techniques, with extensions for multiclass cases [23]. It has a value range between [–1 and 1] that measures the strength and direction of the relationship between two variables as a strong correlation, no correlation, or an inverse relationship.

5) Kappa: Cohen's kappa builds on the idea of measuring the concordance between the predicted and true labels, which are regarded as two random categorical variables [24]. Two categorical variables can be compared by constructing a confusion matrix and determining the marginal row and column distributions. Therefore, we can begin using Cohen's kappa indicators as ratings of the dependence (or independence) between the model's prediction and actual classification.

In the multiclass case, the calculation of Cohen's kappa score is as follows [25]:

$$K = \frac{c \times s - \sum_{k}^{K} pk \times tk}{s^2 - \sum_{k}^{K} pk \times tk}, \qquad (7)$$

where

- C = $\sum_{k}^{K} C_{kk}$ the total number of elements correctly predicted
- S = $\sum_{i}^{K}\ \sum_{j}^{K} C_{ij}$ the total number of elements
- p_k = $\sum_{j}^{K} C_{ki}$ the number of times that class k was predicted (column total)
- t_k = $\sum_{j}^{K} C_{ki}$ the number of times that class k was predicted (rows total)

D. Data Preprocessing

The original data must be preprocessed to remove irrelevant and redundant log entries. The following preprocessing techniques were applied to the collected data before they were trained and analyzed. Each technique is described next.

1) Check and remove null or missing entries: This step is considered one of the most essential steps in data cleaning. All missing data are identified and then removed. It should also clean the data of all irrelevant information, such as "Nan," "n/a," or any other irrelevant values having a number in the URL attribute. These are removed using Python Regex. Empty entries are removed as well.

2) Data normalization and standardization: The process in which the data is cleaned is known as data normalization. This cleaning makes the data regular for all the values of features, which leads to improved segmentation. It removes all the unstructured and redundant data to provide logical data storage. This type of data management is considered particularly crucial for large databases. The raw data hinder the achievement of high efficiency. This problem is dealt with through data normalization. All the feature values are compressed between [0, 1]. The mean is shifted to 0, and the standard deviation is maintained at 1, so the data can be standardized and easily manipulated. Most machine-learning algorithms display noticeable increments in efficiency after the implementation of normalization.

E. Principal Component Analysis (PCA)

Essentially, the algorithm follows the data relationships to a base field and then sequentially applies mathematical functions in the data in different columns and rows along this path to generate the final feature [26]. The performance of classification algorithms may be compromised because of redundant or highly correlated features. Thus, we implemented dimensionality reduction using PCA, as it reduces the size of the feature space while retaining a significant amount of the information [27]. In this regard, many studies have indicated that PCA is less noise-sensitive than other dimension reduction methods [28, 7].

F. Experimental Setting

The proposed model was implemented using Python and Pandas library. A personal computer was used for this experiment, with the following specifications: operating system: macOS Monterey; chip: Apple M1 Pro; total number of cores (processors): eight (six performance and two efficiency); and OS Loader, version: 7459.141.1. In addition, the programming language Python was used.

Based on the parameter settings, the performance of various algorithms can vary. In this work, the algorithms were run using the following parameters:

1) KNN model classifier: K = 5, weights = “uniform”, algorithm = “auto” “fit method is model1.fit(X_train,y_train), leaf_size = 30, p = 2 (Euclidean distance), metric = “minkowski”.

2) SVM classifier: The regularization parameter is set to 1, with a linear kernel, no class weights, and a shrinking heuristic.

3) LR classifier: The norm of the penalty = L2. No class weights, fit intercept is set to true, maximum iterations = 100, and for multi_class = “auto”.

4) AdaBoost classifier: integer value = 42.

5) Decision tree classifier: Decision tree classifier (random_state = 42) with no maximum depth, which means nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples, and the splitter is the “best”.

6) Random forest classifier: One hundred trees, with no maximum depth and a minimum number of splits = 2.

The experiments were designed using different machine-learning and data-analytics libraries, including scikit-learn [29], Numpy [14], and Pandas [31]. Six machine-learning algorithms (described previously) were employed along with the PCA-based feature importance measure with reduced dimensions. Standard 10-fold cross-validation [32] train/test trials were run by partitioning/splitting the entire dataset into training and testing (proportions of 70% and 30%). We ensured that the test data contained a fair distribution for all classes. The following experiments were designed with consistent classifier configurations:

1) Train and test the seven machine-learning algorithms over the individual datasets.

2) Train and test the five machine-learning algorithms over the PCA-based dimension-reduced datasets using a 10-fold CV to compare the performances.

G. Performance Measures

After performing classification, its performance and results must be gauged without specific markers. Therefore, to evaluate a classifier’s capabilities, various performance measures can represent the classification quality of different classifiers on any given data. This provides a deeper insight into the classification techniques’ efficiency than that which using basic accuracy percentages can achieve. The performance evaluation is accomplished using performance metrics such as confusion matrix, precision, recall, and F1 score, as well as basic accuracy. Brief descriptions of each of the performance measures are as follows:

1) The confusion matrix represents the relationship between the actual and predicted values. The following briefly describes the confusion matrix with its four basic elements:

2) True Positive (TP): A vector that gives a count of correctly classified data (presence of condition). Mathematically, this can be calculated by TP/(TP+FP).

3) False Positive (FP): A vector that gives the incorrect classification of data (e.g., the detection of a condition that is not present). Mathematically, this can be calculated by TN/(TN/FN).

4) True Negative (TN): A vector that shows the number of correctly classified data that do not possess the condition (absence of condition).

5) False Negative (FN): A vector that gives the count of wrongly classified data (detected the absence of a condition when it was present).

a) Accuracy: The most basic and extensively relied upon measurement is accuracy, as calculated in Eq. 8 below. It represents the accuracy of the classification results and is the fraction or percentage of a classifier’s total correct identifications against the classifier’s total outcomes, both correct and incorrect.

$$\text{Accuracy} = \text{Correctly classified samples/total number of classifications} \quad (8)$$

b) Precision: This measurement tells us how precise the classifier results are. It gives the percentage of correctly identified positive outcomes against total positive outcomes, which includes false positives.

c) Recall: Recall measures the sensitivity of the classifier. It gives the recognition rate of a classifier. A recall is the proportion of correct positive outcomes against the total number of actual positives present in the dataset. Therefore, it includes false negatives.

d) F1 score: The F1 measurement is an amalgamation of both precision and recall. It is essentially the subjective average of both, namely the recall and precision values. It provides more precise estimations of incorrect outcomes than accuracy when the dataset is imbalanced.

e) Receiver operating characteristic (ROC) area: The ROC metric is used to evaluate the quality of multiclass classifiers. The true positive rate is typically plotted on the Y axis and the false positive rate (FPR) on the X axis. For multiclass problems, ROC curves can be plotted by comparing one class against the others. Applying this OvR to each class will give results in the same number of curves as classes. The ROC score can also be calculated separately for each class. ROC values range between 0 and 1. A model with 100% incorrect predictions has a value of 0.0 while one with 100% accurate predictions has a value of 1.0.

f) Precision-recall curve (PRC) area: PRC can be referred to as the relationship between precision and recall (sensitivity) and is regarded as a more suitable metric for unbalanced datasets. PRC can be calculated by integrating the piecewise function. Consequently, the PRC tends to intersect significantly more frequently than the ROC. The primary distinction between the two is that the number of true negative results is not factored into the PRC because the precision-recall curves are only affected by true positives in most cases. The PRC is generally a tortuous curve, fluctuating upwards and downwards [33].

IV. Model Results

In this model, we were looking to classify users into three classes using each of the six best classifiers regarding the performance measurements that were applied to their evaluation and selection. Each classifier was trained and tested separately to evaluate it in a different portion of the dataset for each classification model with different testing options. We had 70% of the dataset for training and 30% for testing the model besides applying PCA to the dataset. In addition, we performed cross-validation to improve the effectiveness and accuracy of the classification.

A. ML Classification Results

The following Table II and Table III, illustrate the classification performance of the six classifiers used in this work. The tables show the evaluation measure for all six classification models trained on 70% of the dataset and tested on 30%.

TABLE II. Performance Measures for the Six Classification Models Trained on 70% of the Dataset

Classifier	***KNN***	***SVM***	***LR***	***AdaBoost***	***Decision tree***	***Random forest***
Accuracy	93.54%	88.21%	86%	96.07%	95.40%	96.58%
Recall	93.5%	88%	86%	96.1%	95.5%	96.6%
Precision	93.6%	87%	85%	96.1%	95.6%	96.6%
F1 measures	93.6%	87%	85%	96.3%	95.5%	96.6%
MCC	88%	80%	76%	93%	91%	94%
Time (seconds)	1.25	0.01	0.03	0.02	0	0.13

TABLE III. Performance Measures For The Six Classification Models Tested on 30% of the Dataset

Classifier	***KNN***	***SVM***	***LR***	***AdaBoost***	***Decision tree***	***Random forest***
Accuracy	92.89%	88.%	87%	95.8%	94.62%	96.09%
Recall	92.8%	88.8%	87%	95%	94.6%	96.1%
Precision	93%	88.4%	85%	95%	94.7%	96.1%
F1 measures	93%	88.4%	86%	95%	94.6%	96.1%
MCC	87%	81%	77%	92%	91%	93%
Time (seconds)	0.56	0.01	0.01	0.01	0	0.05

As presented in the tables previously, we can see that all the classifiers have been applied to evaluate each classifier's performance. Training data helps construct a machine-learning model and teaches it what the expected outcomes should look like, while the model examines the dataset repeatedly to understand its characteristics and optimize its performance. In contrast, after a machine-learning model is constructed using the training dataset, it must be tested to evaluate the performance of each classifier to select the optimal classifier from those included.

Table II and Table III show the training and testing results regarding the performance matrix evaluation. Comparing the results of all classifiers using part of the dataset, the final results show that the best accuracy is for the random forest classifier, although some of the classifiers, such as AdaBoost and decision tree, have results close to the random forest classifier. Additionally, the LR classifier achieves the lowest accuracy value in both testing and training the model compared to the other classifier models. In this study, AdaBoost was a combination of J48 and decision tree, where the J48 algorithm is closer to the random tree algorithm even in the time it requires for execution. J48 is an algorithm that C4 (one of the decision tree classifiers) employs to generate a decision tree (an extension of ID3). Also referred to as a statistical classifier [30], the J48 algorithm is used to classify various applications and produce accurate classification results, to produce more accurate and fairer comparison results.

The random forest algorithm has the highest accuracy but requires significantly more time to generate a model than the decision tree and AdaBoost algorithms. Besides measuring each classifier's accuracy, because we have an imbalanced dataset, another measurement could assist us in deciding which classifier would perform the best and enable us to have more accurate evaluation results.

We also considered MCC because this indicator is viewed as an effective solution to overcoming the class imbalance issue [34]. In the evaluation, we also considered the F1 measurement, as it is widely used in most application areas of ML, particularly in multiclass cases [35]. Because we had close results for accuracy and time for some of the classifiers, for additional evaluation indicators, we added MCC results to the previously presented tables as well as included them and the F1 results in selecting the best classifier for this proposed model. The random forest classifier has the highest MCC and F1 result among all the machine-learning classifiers.

B. Feature Importance

Next, to understand to what degree each feature contributes to model prediction, which will affect its performance and accuracy in the model, we analyzed feature importance using the four-feature importance measures. The following Fig. 2 shows each feature's rank and score; the scores represent the "importance" of each feature. A higher score indicates that the feature will have more impact on the model used to predict a particular variable.

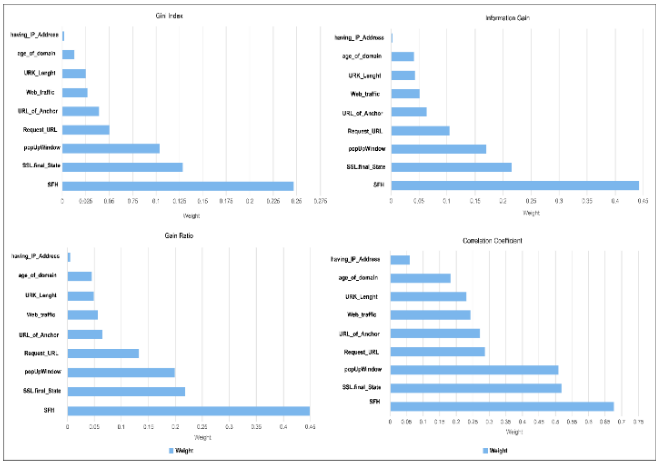

Fig. 2. Top four features and their corresponding weights using information gain, gain ratio, Gini index, and correlation coefficient.

The results show that the top four features are SFH, SSL_final_state, popup window, and requested_URL.

C. PCA

PCA, in this context, is the concept of reducing the number of variables of the dataset while retaining as much information as possible. Accuracy naturally suffers when a dataset's variables are reduced, but the aim of dimensionality reduction is to sacrifice a little accuracy in return for greater simplicity because machine-learning algorithms can analyze data much quickly and easily with smaller data sets as there are fewer extraneous variables to process. Table IV and Table V. show the results of applying PCA to each classification model.

TABLE IV. PERFORMANCE MEASURES WITH PCA (70% TRAINING DATASET)

Classifier	***KNN***	***SVM***	***LR***	***AdaBoost***	***Decision tree***	***Random forest***
Accuracy	95%	86%	85%	97%	95%	98%
Recall	95%	86%	85%	97%	95%	98%
Precision	95%	86%	85%	97%	96%	98%
F1 measures	95%	86%	85%	97%	95%	98%
MCC	93%	79%	77%	95%	93%	96%
Time (seconds)	1.27	0.42	0	0.56	0	0.19

TABLE V. PERFORMANCE MEASURES WITH PCA (30% TESTING DATASET)

Classifier	***KNN***	***SVM***	***LR***	***AdaBoost***	***Decision tree***	***Random forest***
Accuracy	93%	85%	85.18%	95%	93%	96%
Recall	93%	85%	85%	95%	94%	96%
Precision	93%	90%	89%	95%	94%	96%
F1 measures	93%	86%	86%	95%	94%	96%
MCC	87%	79%	78%	92%	89%	93%
Time (seconds)	0.58	0.2	0.1	0.10	0.03	0.07

According to the previous presented tables (Table IV and Table V), we can see the improvement in accuracy when the PCA was applied to the dataset because of dimensionality reduction where the redundant and irrelevant data have been removed; in other words, the data that have no significant effect on the classification results have been removed. Additionally, the improvement in the MCC results is noticeable.

For further investigation, and as the final results of all six classifiers were similar, a 10-fold data split was constructed, as shown in the following Fig. 3, to understand how the algorithms performed.

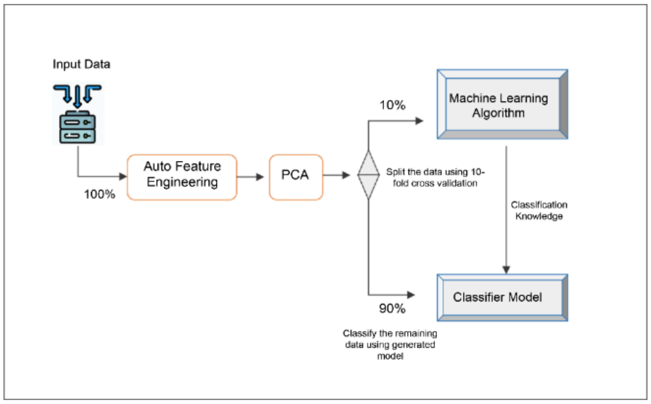
Input Data
100%
Auto Feature Engineering
PCA
Split the data using 10-fold cross validation
10%
Machine Learning Algorithm
Classification Knowledge
90%
Classifier Model
Classify the remaining data using generated model

Fig. 3. Cross-validation process model.

The classifier constructed nine identical instances of the dataset and then split the data in each of these instances into 10% for training and 90% for testing. Each of these nine instances was trained/tested with a unique split. Finally, the result from each of these instances was combined into a final result. Because nine combinations of 10% of the data were used to classify the data, a reasonably realistic result could be obtained using this 10-fold cross-validation split.

Using cross-validation emphasizes that, as previous Fig. 4 and Fig. 5 shown, although all the classifier results are similar to each other, the random forest classifier shows the best performance regarding all performance measures and, in particular, the lowest FPR (2.2%), with incorrectly classified instances of 4% in the cross-validation test.

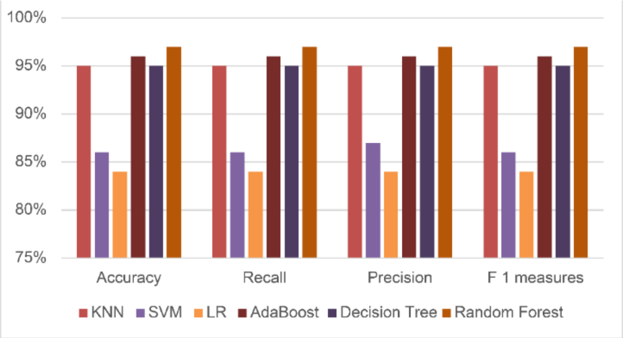

Fig. 4. Cross-validation.

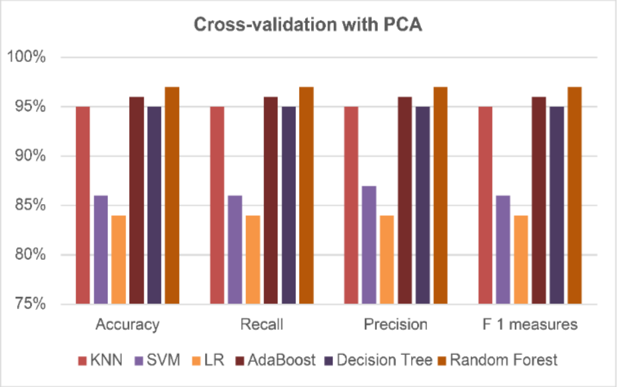

Fig. 5. Cross-validation with PCA.

D. Proposed Model

This study aimed to propose a model that can assist organizations in providing dedicated and targeted cybersecurity awareness sessions to their employees based on an analysis of their online behavior. The problem at hand was formulated as a multiclass problem. We differentiated between three classes: malicious, suspicious, and normal. Based on influential features and the best-performing classifier we identified, we propose an ML-based classification model. Fig. 6 shows the proposed model.

The dataset was first fed into the classifier, which was then used to extract features. Following that, a few preprocessing techniques were applied to ensure that the dataset was clean. After that, we applied the machine-learning classification models to the dataset.

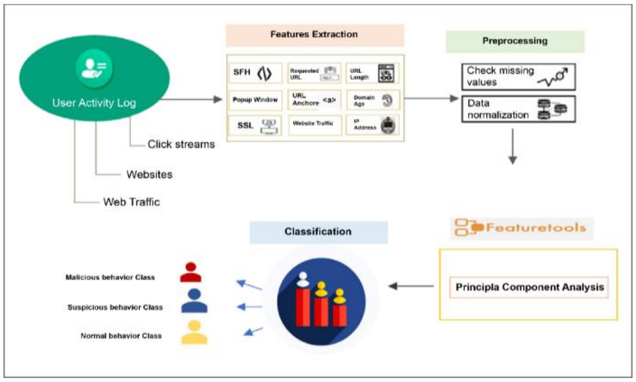

Fig. 6. Machine–learning-based classification model.

The Kruskal–Wallis test was used to compare the performance of the various models in the study. The Kruskal–Wallis test is a nonparametric statistical test that is computed based on the rank and the sum of ranks. The null hypothesis assumes that the performance measures of the models are drawn from the same distribution and that any differences are due to chance.

The hypothesis of the test is given below:

H_0: The performances of the models are equal (i.e., there are no statistically significant differences in model performances).

H_A: At least one model performance is different (i.e., there are statistically significant differences in model performances).

1) Test statistics: The test statistic of the Kruskal–Wallis, H measures the differences among the performance of the groups and is given by the following:

$$H = \frac{12}{N(N+1)}\left(\sum \frac{R_i^2}{n_i}\right) - 3(N+1)$$

where n_i total number of observations in the model i

R_i the sum of the ranks of model i

N the total number of observations across all models.

The Kruskal–Wallis test statistic approximates a chi-square distribution with k-1 degrees of freedom, where k is the number of groups (models).

The observations for the test are obtained from the classification accuracy of each of the models from 10-fold cross-validation. Hence, this ensures that each classifier used is evaluated on the same splits of the dataset via the 10-fold cross-validation. These observations (classification accuracies from the 10-fold cross-validation) are provided in the appendix below. The Kruskal–Wallis test is then used to compare whether there is a statistically significant difference among the performance of these models. All analyses were implemented using Python software.

2) Test results: The test statistics and the associated p-values are given below in Table VI.

TABLE VI. TEST STATISTICS AND THE ASSOCIATED P-VALUES

	H statistics	***p-value***
Training set	45.4512	1.1745×10^{-8}
Testing set	48.9979	2.2215×10^{-9}
PCA with Training set	49.0362	2.1818×10^{-9}
PCA with Testing set	49.3816	1.8544×10^{-9}

Decision Rule

Reject H_0 if the p-value ≤ 0.05; else, fail to reject H_0.

Because the p-value associated with any of the H statistics is less than 0.05, we reject H_0. Hence, enough evidence supports the alternative hypothesis that at least one of the model performances is different. Therefore, there are statistically significant differences in model performances.

3) Post hoc statistical test: Dunn’s Test with the Holm–Bonferroni Correction: Given that the Kruskal–Wallis test showed evidence of statistically significant differences in model performance, the Dunn’s test with Holm–Bonferroni p-value correction was conducted to ascertain which pairs of models differ significantly from each other.

Dunn’s test is a nonparametric pairwise post hoc test used to compute the rank-based Z-values for pairs of the models and convert these values into p-values. The Holm–Bonferroni correction is then applied to these p-values to control for family-wise error rate (FWER). FWER refers to the probability of committing at least one type I error among the pairs of comparisons. All computations are conducted using Python.

4) Hypothesis: The hypothesis of this test for each of the pairs of models is as follows:

H_0: There is no statistically significant difference between the pair of models compared.

H_A: There is a statistically significant difference between the pair of models compared.

5) Decision rule: Reject H_0 if the Holm–Bonferroni Adjusted p-value ≤ 0.05; else fail to reject H_0

TABLE VII. RESULT OF DUNN’S TEST WITH THE HOLM–BONFERRONI CORRECTION ON TRAIN PERFORMANCE

Model 1	*Model 2*	*HB Adj. p-value*	*Result*
KNN	SVM	1	Not significant
KNN	LR	0.263806	Not significant
KNN	AdaBoost	1	Not significant
KNN	Decision Tree	0.899874	Not significant
KNN	Random Forest	1	Not significant
SVM	LR	0.899874	Not significant
SVM	AdaBoost	1	Not significant
SVM	Decision Tree	0.263806	Not significant
SVM	Random Forest	1	Not significant
LR	AdaBoost	1	Not significant
LR	Decision Tree	0.000080	Significant
LR	Random Forest	0.935495	Not significant
AdaBoost	Decision Tree	0.002421	Significant
AdaBoost	Random Forest	1	Not significant
Decision Tree	Random Forest	0.218907	Not significant

TABLE VIII. RESULT OF DUNN’S TEST WITH THE HOLM–BONFERRONI CORRECTION ON TEST PERFORMANCE

Model 1	*Model 2*	*HB Adj. p-value*	*Result*
KNN	SVM	1	Not significant
KNN	LR	1	Not significant
KNN	AdaBoost	1	Not significant
KNN	Decision Tree	1	Not significant
KNN	Random Forest	0.003238	Significant
SVM	LR	0.004285	Significant
SVM	AdaBoost	1	Not significant
SVM	Decision Tree	0.004285	Significant
SVM	Random Forest	1	Not significant
LR	AdaBoost	0.211045	Not significant
LR	Decision Tree	1	Not significant
LR	Random Forest	0.0000003	Significant
AdaBoost	Decision Tree	0.211045	Not significant
AdaBoost	Random Forest	0.045905	Significant
Decision Tree	Random Forest	0.0000003	Significant

TABLE IX. RESULT OF DUNN’S TEST WITH THE HOLM–BONFERRONI CORRECTION ON TRAIN PCA

Model 1	*Model 2*	*HB Adj. p-value*	*Result*
KNN	SVM	1	Not significant
KNN	LR	1	Not significant
KNN	AdaBoost	1	Not significant
KNN	Decision Tree	0.031372	Significant
KNN	Random Forest	1	Not significant
SVM	LR	1	Not significant
SVM	AdaBoost	1	Not significant
SVM	Decision Tree	0.092881	Not significant
SVM	Random Forest	1	Not significant
LR	AdaBoost	1	Not significant
LR	Decision Tree	0.018019	Significant
LR	Random Forest	1	Not significant
AdaBoost	Decision Tree	0.001791	Significant
AdaBoost	Random Forest	1	Not significant
Decision Tree	Random Forest	0.62984	Not significant

TABLE X. RESULT OF DUNN’S TEST WITH THE HOLM–BONFERRONI CORRECTION ON TEST PCA

Model 1	*Model 2*	*HB Adj. p-value*	*Result*
KNN	SVM	0.056299	Not significant
KNN	LR	0.692211	Not significant
KNN	AdaBoost	1	Not significant
KNN	Decision Tree	1	Not significant
KNN	Random Forest	0.007251	Significant
SVM	LR	0.0000086	Significant
SVM	AdaBoost	1	Not significant
SVM	Decision Tree	0.000714	Significant
SVM	Random Forest	1	Not significant
LR	AdaBoost	0.001994	Significant
LR	Decision Tree	1	Not significant
LR	Random Forest	0.0000003	Significant
AdaBoost	Decision Tree	0.056299	Not significant
AdaBoost	Random Forest	1	Not significant
Decision Tree	Random Forest	0.000047	Significant

The results above show that there exists at least one instance where pair of models are statistically different regarding performance.

V. Discussion

Currently, user behavior is one of the most critical factors in organizations' cybersecurity, and it can put the organization's safety, data, assets, reputation, and individuals at risk. Thus, providing cybersecurity training for users or employees plays a vital role in improving their attitude and behavior when online, particularly when the training is directed and targeted based on user needs and deficiencies.

Due to the large number of attributes and high volume of online data, we employed machine-learning techniques in the context of providing cybersecurity awareness by analyzing online user behavior. In this context, the main objective was the enhancement of people's cybersecurity awareness through the provision of targeted cybersecurity awareness programs that would lead to a decrease in cybersecurity issues and intrusions inside an organization.

Although user behavior analysis and the use of machine-learning techniques for analyzing user behavior are not new, the novelty of this paper lies in the fact that it is among the first few research that analyses human online behavior and applies ML to target employees with suitable awareness materials, the primary objective of this study differed from those of previous models and other studies. The concept of user behavioral analysis has been included previously in a number of fields and domains, such as marketing applications, to adopt new and efficient marketing strategies that are based on user data (i.e., utilizing recorded information of the past activities of potential clients in data-based behavioral marketing) [36]. It has also been included in recommendation systems by predicting user interests from a user's last browsing and searching activities, for example, by recommending specific articles for readers or an item of clothing during shopping [34].

Moreover, ML is used to classify users, such as on social media. It can be applied to building a practical system for detecting fake identities by using server-side clickstream models to group users with similar clickstreams into clusters or analyze user browsing behavior on specific websites [35], including e-commerce, education, and healthcare. The aim is the personalization or targeting of users with advertisements based on their browsing behavior. Thus, the application of machine-learning techniques helps classify users with a high degree of accuracy. In the security domain, its value has been proven in the fight against fraud and other applications [37]. Moreover, ML is used in the detection of phishing emails using algorithms. This can automate the detection of phishing emails using a variety of techniques, including deep-learning detectors that automate the process [38], where deep-learning algorithms have produced impressive results with unstructured data such as email data [39].

This proposed model can aid organizations in maintaining the security of their assets and data, as we include the human factor by enhancing the awareness levels of their employees regarding cybersecurity threats by providing appropriate training and awareness based on the analysis of their online behavior that may help the organization in classifying users based on the analysis results.

Ryu et al. [18] and many others demonstrated the importance of personal security factors in this area. They showed the significance of raising awareness of the importance of security in industries. As a result, regardless of the type of security system in place, considering the importance of employees' online awareness and behaviors is critical.

Many other researchers [34–36] have shown that a strong awareness-raising program is required to ensure that employees understand their respective IT security duties and roles to protect the IT resources delegated to them. However, these studies achieved low accuracy in measuring users' online awareness; for example, questionnaires or surveys were published to a general audience, and the analysis was performed based on their answers [33]. This approach fails to analyze employees' actual online behavior that reflects their cybersecurity knowledge. As a result, the awareness content that is subsequently provided is not suitable for each individual.

In this study, we applied several machine-learning classifications to the same dataset with the same percentage split: 70% for training the model and 30% for testing the model. Thereafter, we compared the final results of the performance measures among all classifiers to determine the best one. The results demonstrated that the random forest classifier was the best option to choose with the best results, and it could be applied for analyzing user behavior inside the organization. Random forest achieved the highest accuracy rate in both training and testing sets of the whole dataset with different methods of testing and different measures that have been used, which are the accuracy, MCC, and F1 measures.

For the AdaBoost, decision tree, and random forest classifiers, the accuracy rates were similar. Therefore, we included the MCC and F1 measurements to ensure a more accurate comparison, rather than just taking into consideration the FPR and which classifier had the lowest FPR. PCA was also applied to the concept of reducing the dimensionality of the dataset used in the model, and cross-validation was used to validate each classifier.

Theoretically, when considering the computational costs of the random forest classifier, the complexity of the test time of a random forest of size T, which is the number of trees to build, and the maximum depth D is O ($T.D$), which is 0 by default and is the unlimited depth of the tree. Another important disadvantage is the memory space required for random forest classification, which is calculated by $O(2^D)$ [33]. This experiment showed that the running time to build the model is 0.23 s, on average, and the time required to test the model on 5,683 instances of training data is 0.11 s. Additionally, the time required to build the model is 0.19 s, and the time required to test the model on the supplied test set is 0.09 s for 2,435 instances.

Random forest showed its effectiveness in the classification process, as it did in many previous works, such as in Android malware classification [40], where it performed very well with an accuracy of over 99%. In general, the samples were correctly classified, and the highest number of misclassified cases resulted from samples from the malicious class being mistakenly assigned to the benign class.

Moreover, Farnaaz and Jaber [41] used random forest classification to detect intrusions on a system, where the random forest classifier was used to classify four types of attacks. According to empirical findings, the proposed model was effective, with a low false alarm rate and a high detection rate.

Thus, the experimental results conclude that users can be successfully classified based on their online behavior to target them with the correct awareness materials using a machine-learning-based model.

VI. Conclusion

The causes of and methods for preventing security issues and risks to any organization are continually changing as a direct result of the ongoing evolution of cybersecurity threats. In addition, individuals' knowledge levels, technical skills, and levels of awareness regarding cybersecurity vary, which is one of the reasons for the difficulty in controlling their online behavior and the associated risks. Because of this, the measurement and analysis of online behavior are now absolutely necessary for any organization that wants to protect its assets from both internal and external breaches of security. A substantial number of earlier studies have established a clear connection between online users' actions and various problems and dangers related to cybersecurity. Regardless of the security technology in place, the most reliable indicator of potential vulnerabilities in an organization or network is users' actions when they are online. Providing directed and dedicated awareness sessions and training regarding cybersecurity is essential in any organization, and this must be managed appropriately.

In this study, we proposed a machine-learning-based model that can assist organizations in providing targeted awareness sessions to their employees based on an analysis of the employees' behaviors. The model will classify the users into three classes: malicious, suspicious, and normal behavior. This classification will ultimately increase awareness of particular behaviors. It may enable organizations to target each employee segment with appropriate sessions and training, increasing the effectiveness of resources.

To achieve this objective, a machine-learning model can be applied to identify patterns in users' web activities and, as a result, classify users according to their activities in virtual spaces. The primary goal of the proposed model is to help organizations target users with sessions of security awareness that are specific and tailored to their needs. Raising awareness can be automated based on specific behaviors, which may result in an effective process that saves organizations time and money. Six well-known machine-learning algorithms, namely KNN, LR, SVM, AdaBoost, decision tree, and random forest classifiers, were trained and tested independently on a user behavior records dataset by splitting the dataset into a 70% training dataset and a 30% testing dataset. The random forest classifier showed superior performance among all the classifiers regarding the accuracy, F-measure, and the MCC measure. While applying PCA, the model also demonstrates a high accuracy rate, low FPR, high recall, and precision, as well as high F-measures.

Furthermore, as this model is based on machine learning, Machine learning methods at some point also have limitations, as when applied to security that can result in amplified nuances. They can give false positives and false negatives, causing them to miss detection, or insiders can corrupt the dataset, which will lead to wrong outcomes or corruption of the model itself. Furthermore, hackers are also learning machine learning and applying them to their hacking procedures and fishing for loopholes to exploit.

This model has the potential to undergo further development by automatically learning user classes to set up appropriate awareness sessions and training without human intervention. In subsequent research, an improved feature analysis might be included with the goal of making the model more precise. Another potential development would be the incorporation of additional user behavior categories. In addition, a monitoring strategy can be used to observe user behavior. Management can be notified if there is no change in the manner in which users conduct themselves while online. In the future, we plan to increase the number of classes for classifying users and the amount of automated content to be sent to each class to enhance the model's value to organizations.

References

[1] Chen, C. C., Shaw, R. S., & Yang, S. C. (2006). Mitigating information security risks by increasing user security awareness: A case study of an information security awareness system. Information Technology, Learning & Performance Journal, 24(1).

[2] Johnston, A. C., Warkentin, M., McBride, M., & Carter, L. (2016). Dispositional and situational factors: influences on information security policy violations. European Journal of Information Systems, 25(3), 231-251.

[3] Donalds, C., & Osei-Bryson, K. M. (2020). Cybersecurity compliance behavior: Exploring the influences of individual decision style and other antecedents. International Journal of Information Management, 51, 102056.

[4] Bishop, M. (2005). Authentication. In Introduction to Computer Security (pp. 171-96). Addison-Wesley.

[5] de Zafra, D. E., Pitcher, S. I., Tressler, J. D., & Ippolito, J. B. (1998). Information technology security training requirements: A role-and performance-based model. NIST Special publication, 800(16), 800-16.

[6] Kruger, H. A., & Kearney, W. D. (2006). A prototype for assessing information security awareness. Computers & security, 25(4), 289-296.

[7] Carblanc, A., & Moers, S. (2003). Towards a culture of online security: making information systems trustworthy is a job that concerns everyone. What can be done?. OECD Observer, (240-241), 30-32.

[8] Ryu, S., Kang, Y. J., & Lee, H. (2018, February). A study on detection of anomaly behavior in automation industry. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 377-380). IEEE.

[9] Curme, C., Preis, T., Stanley, H. E., & Moat, H. S. (2014). Quantifying the semantics of search behavior before stock market moves. Proceedings of the National Academy of Sciences, 111(32), 11600-11605.

[10] Bernaschina, C., Brambilla, M., Mauri, A., & Umuhoza, E. (2017). A big data analysis framework for model-based web user behavior analytics. In Web Engineering: 17th International Conference, ICWE

2017, Rome, Italy, June 5-8, 2017, Proceedings 17 (pp. 98-114). Springer International Publishing.

[11] Niranjan, A., Nitish, A., Deepa Shenoy, P., & Venugopal, K. R. (2016). Security in data mining-a comprehensive survey. Global Journal of Computer Science and Technology, 16(5).

[12] [12] Kumar, S., & Singh, M. (2018). Big data analytics for healthcare industry: impact, applications, and tools. Big data mining and analytics, 2(1), 48-57.

[13] Robila, S. A., & Ragucci, J. W. (2006). Don't be a phish: steps in user education. Acm sigcse bulletin, 38(3), 237-241.

[14] Nunes, P., Antunes, M., & Silva, C. (2021). Evaluating cybersecurity attitudes and behaviors in Portuguese healthcare institutions. Procedia Computer Science, 181, 173-181.

[15] Callara, M., & Wira, P. (2018, November). User behavior analysis with machine learning techniques in cloud computing architectures. In 2018 International Conference on Applied Smart Systems (ICASS) (pp. 1–6). IEEE.

[16] Tsai, F. S. (2010). Comparative study of dimensionality reduction techniques for data visualization. Journal of artificial intelligence, 3(3), 119-134.

[17] Jiang, H., He, M., Xi, Y., & Zeng, J. (2021). Machine-learning-based user position prediction and behavior analysis for location services. Information, 12(5), 180.

[18] Ryu, S., Kang, Y.J., & Lee, H. (2018, February). A study on detection of anomaly behavior in automation industry. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 377–380). IEEE.

[19] https://archive.ics.uci.edu/ml/datasets/website+phishing

[20] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28.

[21] Alhogail, A., Al-Turaiki, I.: Improved detection of malicious domain names using gradient boosted machines and feature engineering. Inf. Technol. Control 51, 313–331 (2022)

[22] https://www.javatpoint.com/entropy-in-machine-learning

[23] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[24] Ranganathan, P., Pramesh, C. S., & Aggarwal, R. (2017). Common pitfalls in statistical analysis: Measures of agreement. Perspectives in clinical research, 8(4), 187.

[25] Tallón-Ballesteros, A. J., & Riquelme, J. C. (2014). Data mining methods applied to a digital forensics task for supervised machine learning. Computational intelligence in digital forensics: forensic investigation and applications, 413-428.

[26] Al-Turaiki, I., & Altwaijry, N. (2021). A convolutional neural network for improved anomaly-based network intrusion detection. Big Data, 9(3), 233-252.

[27] Kanter, J. M., & Veeramachaneni, K. (2015, October). Deep feature synthesis: Towards automating data science endeavors. In 2015 IEEE international conference on data science and advanced analytics (DSAA) (pp. 1-10). IEEE.

[28] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

[29] Robila, S.A., & Ragucci, J. W. (2006). Don’t be a phish: steps in user education. Acm sigcse bulletin, 38(3), 237–241.

[30] Patel, B. R., & Rana, K. K. (2014). A survey on decision tree algorithm for classification. International Journal of Engineering Development and Research, 2(1), 1-5.

[31] Agrawal, R., & Srikant, R. (1995, March). Mining sequential patterns. In Proceedings of the eleventh international conference on data engineering (pp. 3-14). IEEE.

[32] Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng,H., & Zhao, B. Y. (2013). You are how you click: Clickstream analysis for sybil detection. In 22nd USENIX Security Symposium (USENIX Security 13) (pp. 241–256).

[33] Solé, X., Ramisa, A., & Torras, C. (2014). Evaluation of random forests on large-scale classification problems using a bag-of-visual-words representation. In Artificial Intelligence Research and Development (pp. 273-276). IOS Press.

[34] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure, 405(2), 442–451.

[35] Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Random k-labelsets for multilabel classification. IEEE transactions on knowledge and data engineering, 23(7), 1079-1089.

[36] Foxall, G. R. (1994). Behavior analysis and consumer psychology. Journal of Economic Psychology, 15(1), 5-91.

[37] Baig, A. R., & Jabeen, H. (2016). Big data analytics for behavior monitoring of students. Procedia Computer Science, 82, 43-48.

[38] Alhogail,A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email.Computers & Security, 110, 102414.

[39] Halgaš, L., Agrafiotis,I., & Nurse, J. R. (2020). Catching the phish: Detecting phishing attacks using recurrent neural networks (rnns). In Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, South Korea, August 21–24, 2019, Revised Selected Papers 20 (pp. 219-233). Springer International Publishing.

[40] Alam, M.S., & Vuong, S. T. (2013, August). Random forest classification for detecting android malware. In 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing (pp. 663–669). IEEE.

[41] Farnaaz,N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213–217.

[42] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[43] Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.