FEATURE ARTICLE: CYBERSECURITY

A Data-Driven Classification Framework for Cybersecurity Breaches

Priyanka Rani, *Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh*

Abhijit Kumar Nag, *Texas A&M University–Central Texas, Killeen, TX, 76549, USA*

Rifat Shahriyar, *Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh*

Unauthorized access to sensitive or confidential data results in a data breach, which can cause significant harm to an organization. Reporting breaches and reviewing prior records can help reduce damages. To aid in preparation, antivirus and security companies have published data breach reports, but they can be difficult to comprehend and require substantial effort to study. This article proposes a data breach incident classification framework using machine learning algorithms (naive Bayes, logistic regression, support vector machine, and random forest) on a dataset from the Privacy Rights Clearinghouse. The framework's performance is evaluated using various metrics, including accuracy, F1 score, and confusion matrix. The article also employs topic modeling with latent Dirichlet allocation to enhance the classification's accuracy.

A cyberattack often leads to a data breach where cybercriminals gain unauthorized access to a company's network or workstation, stealing sensitive and confidential personal and financial data from customers or users. Such incidents cause significant losses to organizations. *Cost of a Data Breach Report*[5] reveals that the average cost of a data breach is $3.86 million. The report also highlights that data breaches can go unnoticed for around 197 days and that it takes another 69 days to fix the issue. By the time the breach is detected, the harm has already been done.

Knowledge is a crucial defense in any crisis, including data breaches. To mitigate the risks of such incidents, organizations must gain perspective on, insight about, and knowledge of the threats they are likely to face. By understanding the big picture and how cyberattacks typically unfold in various industries, they can efficiently deploy defenses and make the most of their security budget. To facilitate this, numerous antivirus and software companies, such as Verizon, Symantec, PwC, McAfee, and Visa, have been publishing data breach reports.[15,14]

Today, many companies have established protocols for reporting data breach incidents. However, data breach reports rely on textual data, which can be challenging to classify and often require manual intervention, making the process time-consuming and error prone. These challenges hinder the publication of such reports by antivirus and software companies. To address this issue, this article proposes a classification framework for data breach incidents to assist companies in publishing useful reports.

We employ four different text classification algorithms, namely, naive Bayes, logistic regression, support vector machine (SVM), and random forest classifier, to classify data breach incidents. We also compare the performance of these classifiers using various performance metrics, such as accuracy, F1 score, and confusion matrix. Additionally, we develop a method to classify unknown types of data breaches into known data breach incidents. To evaluate our proposed framework, we use popular datasets from Privacy Rights Clearinghouse (PRC)[1] and train and evaluate our model using both the train–test and k-fold cross-validation approaches. Finally, we utilize topics derived from latent Dirichlet allocation (LDA)-based topic modeling to augment classification performance.

The rest of this article is organized as follows. The "Dataset" section details the source of our dataset. The "Literature Review" section provides a literature review of related research. Then, the "Proposed Classification Framework" section presents our proposed classification framework, including improvements with

1520-9202 © 2024 IEEE
Digital Object Identifier 10.1109/MITP.2024.3374096
Date of current version 8 May 2024.

topic modeling. In the section "Improve Performance With LDA," we aim to improve performance by employing the topic modeling technique known as LDA. The "Experiment Result" section discusses the experimental results and their analyses as well as the deployment of the model. Finally, the "Conclusion" section concludes our article and outlines some future research directions.

DATASET

We have a dataset from PRC[1] that we aim to use for classifying the column "Description of Incidents" into the column "Type of Breach." Our classification framework comprises the seven known types of data breaches found in PRC, namely, CARD, HACK, INSD, PHYS, PORT, STAT, and DISC. We will not include the UNKWN type in our training data, as we intend to classify unknown data breach records into one of the known types. The CARD type pertains to breaches involving debit and credit cards not obtained via hacking, while HACK refers to those caused by outside parties or malware infection. INSD refers to data breaches caused by insiders, such as employees, contractors, or customers. PHYS involves lost, discarded, or stolen paper documents, while PORT involves portable devices, such as laptops, personal digital assistants, smartphones, memory sticks, CDs, hard drives, and data tapes, that have been lost, discarded, or stolen. STAT refers to stationary computers that are lost, inappropriately accessed, discarded, or stolen, including computers or servers not designed for mobility. DISC involves the unintended disclosure of sensitive data, such as through public social media posts, mishandled or erroneously sent mailings or faxes, and the like.

LITERATURE REVIEW

We examine various research studies that investigate data breach incidents. Some researchers used the same dataset, PRC, as we do. For instance, Maochao et al.[17] analyzed 12 years of cyberhacking incidents, such as malware attacks, using the PRC dataset. Similarly, Edwards et al.[3] used a dataset of 2253 data breach events from PRC that span over a decade (2005–2015) as well as model breach size and frequency using a log-normal or log-skew-normal distribution and a negative binomial distribution, respectively. They distinguish data breaches into two types: negligent breaches (e.g., incidents caused by lost, discarded, or stolen devices) and malicious breaches (e.g., hacking).

Wheatley et al.[16] analyzed a dataset that combines the electronic data loss database[13] and PRC records to calculate the maximum breach size using extreme value theory[4] and a doubly truncated Pareto distribution.

Maillart and Sornette[7] examined 956 instances of personal identity loss in the United States from 2000–2008 and found that a heavy-tailed distribution could model the per-incident personal identity loss.

Juma'h and Ainsour[6] conducted an analysis of data breaches and highlight the legal, social, and economic concerns that arise from business performance studies. They argue that affected companies are liable to their employees, customers, and investors due to the internal inadequacies suggested by data breach notifications.

Böhme and Kataria[2] proposed a new classification of cyberrisk correlation properties based on a two-tier approach that considers both internal and global dependence. Mishra and Thakur[8] used various machine learning algorithms to classify spam e-mails. Ramage et al.[9] extended the LDA model to a supervised form and analyzed its applications in a microblogging environment.

PROPOSED CLASSIFICATION FRAMEWORK

In our study, we aim to make predictions on a set of samples using supervised machine learning algorithms. Our research employs four different algorithms, namely, linear SVM, logistic regression, naive Bayes, and random forest classifier. The sample datasets we have belong to one of the seven types of data breaches, and the prediction result should also be one of these seven types. Thus, our problem is categorized as a multiclass supervised machine learning problem.

Data Explorations

In the initial stage of training our dataset, we perform an analysis of the dataset to determine the number of data breaches that fall into each type, as presented in Table 1.

This study requires only two columns from the dataset, namely, "Type of Breach" and "Description of Incident." As an illustration, the input and output of the classifiers are provided below:

- *Input*: "A data breach at the University of Alaska has impacted dozens of current and former employees and students, officials said. The university said the accounts of 50 people were impacted."
- *Output*: HACK.

To ensure accurate predictions, it is necessary to address the missing values in the "Description of Incident" column by removing them. Table 2 displays some rows of the dataset after cleaning up.

TABLE 1. Example dataset from the Privacy Rights Clearinghouse.[1]

Type of breach	Description of incident	Information source . . .
HACK	We recently became aware that your information . . .	California Attorney General . . .
INSD	On September 20, 2016, DFCU learned that . . .	California Attorney General . . .
DISC	HMSA notified 10,800 members of a data breach . . .	Databreaches.net . . .
STAT	On or around October 31, 2014, a paper . . .	Government agency . . .
HACK	As reported by Health and Human Services . . .	Government agency . . .

Text Preprocessing

Data preprocessing is an essential process that converts raw data into a format that is both valuable and efficient. Prior to preprocessing, the dataset contains 8222 rows (refer to Figure 1). Among the various types of breaches, the "HACK" category exhibits the highest number of records, with a total of 2533 instances.

In this research study, our text-cleaning process encompasses several steps, which involve decoding HTML, eliminating stop words, converting text to lowercase, removing punctuation, and filtering out any undesirable characters.

We present a comparison of the dataset before and after undergoing preprocessing in Table 3. Initially, the dataset contained *8222* rows, but, after preprocessing, the number of rows was reduced to *5749* (refer to Figure 1). This preprocessing step has also resulted in a slight improvement in the balance of the previously highly imbalanced dataset. Specifically, the count of data breaches classified as "HACK" has decreased from 2533 to 1942 (as shown in Figure 1). The resulting balanced dataset contributes to faster and more accurate classification processes.

Text Representation

Typically, text classification algorithms face challenges when directly processing text documents. Additionally, the variable lengths of text documents pose another hurdle for classification tasks. Most algorithms require numerical feature vectors of fixed sizes.

To overcome these challenges, one of the widely used approaches for feature extraction from text is the bag of words technique. This method considers the

TABLE 2. Dataset after removing noise.[1]

Index	Type of breach	Description of incident
1	HACK	We recently became aware that your information . . .
2	INSD	On September 20, 2016, DFCU learned that . . .
3	DISC	HMSA notified 10,800 members of a data breach . . .
4	STAT	On or around October 31, 2014, a paper . . .

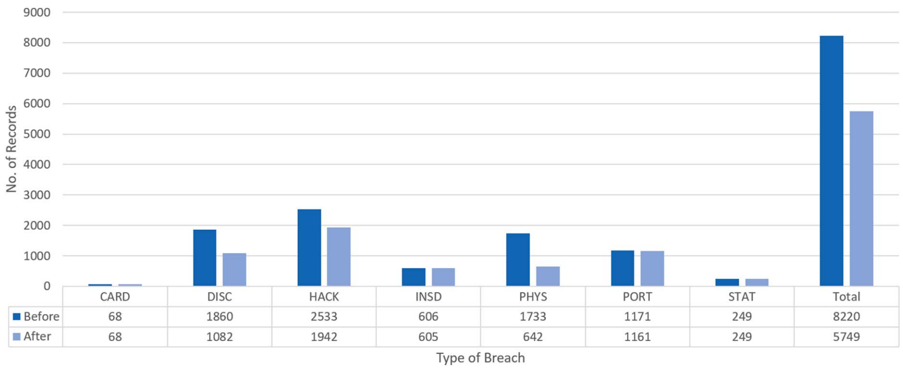


FIGURE 1. Before and after data preprocessing.

TABLE 3. Top five rows of the dataset before and after preprocessing.

Type of breach	Description of incident
Top five rows of dataset before preprocessing	
DISC	Location of breached information: unauthorized . . .
DISC	Location of breached information: unauthorized . . .
HACK	Location of breached information: hacking/IT I . . .
HACK	Location of breached information: hacking/IT I . . .
DISC	Location of breached information: unauthorized . . .
Top five rows of dataset after preprocessing	
HACK	Ticketfly was the target of a malicious cyber . . .
DISC	*New Scientist* reports: Data from millions of . . .
HACK	*TechRadar* reports: PageUp, an Australia-based software . . .
INSD	*Bank Info Security* reports: Nuance Communications . . .
HACK	WIVB4 is reporting: University at Buffalo leaders . . .

IT: Information Technology.

occurrence and frequency of words within each text document. However, it does not take into account the order of word occurrences.

N-Grams Model

An approach to improve the bag of words model involves constructing a vocabulary that incorporates grouped words. This method involves labeling each individual word or token as a “gram.” When we specifically construct a vocabulary for two-word pairs, it is referred to as a bigram model. Similarly, an N-gram denotes a sequence of N words.

Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF is a numerical metric that gauges the significance of a word to a document within a corpus or collection. The value of TF-IDF rises in proportion to the frequency with which a word appears in a particular document. We calculate the TF-IDF score for both unigram and bigram models. The feature vectors are then extracted, resulting in 8903 features for each of the 5749 data breach incidents.

Model Training

After transforming the dataset, the subsequent step involves training it using various algorithms for text classification. Some commonly employed algorithms include naive Bayes, decision trees, SVM, random forest classifier, k-nearest neighbors, and logistic regression. In our research, we perform a benchmark study using four supervised machine learning algorithms, which are as follows:

- *Naive Bayes*: Naive Bayes is a classification model that leverages conditional probability and applies Bayes’ theorem to predict the class of unknown datasets.
- *Logistic regression*: Logistic regression is a widely employed supervised machine learning algorithm utilized primarily for classification purposes, specifically in predicting discrete-valued outcomes.
- *SVM*: SVM is a versatile linear model employed for solving classification and regression problems. It can handle both linear and nonlinear problems effectively, making it suitable for various practical scenarios.
- *Random forest*: Random forest is a robust ensemble tree-based learning technique, known for its effectiveness in supervised machine learning. It finds wide application in tasks involving both classification and regression.

Imbalanced Classes Problem

The dataset exhibits a bias toward the HACK, DISC, and PHYS types of data breaches (refer to Figure 1). This bias can potentially lead to imbalanced results favoring the majority class. Therefore, it is crucial to address this issue before training the dataset by carefully configuring the model or employing artificial techniques to balance the dataset.

The synthetic minority oversampling technique (SMOTE) is a popular method used to address imbalanced datasets. It involves oversampling the minority classes by creating synthetic new records based on instances from the minority class. By employing SMOTE, we can attain a more balanced dataset.

Moreover, it is essential to utilize a performance metric that takes into account both false positives and false negatives, such as the F1 score. The F1 score is a balanced measure that combines precision and recall scores. Precision indicates the ratio of accurately predicted positive outcomes to the total predicted positive outcomes, while recall measures the proportion of correctly predicted positive outcomes among all actual positive instances. The F1 score can be calculated using the following formula:

$$\mathrm{F1\quad score} = 2 * \frac{(\mathrm{recall} * \mathrm{precision})}{(\mathrm{recall} + \mathrm{precision})}.$$

Train Dataset

To train the dataset, we employ two common approaches: the train–test split and K-fold cross validation. In both approaches, we begin by applying SMOTE on the training dataset to address the issue of class imbalance. This results in a balanced dataset that we can then utilize for training.

In the train–test split approach, we divide the dataset into two portions: a training set and a testing set. The training set is used to train the classifiers, while the testing set is kept separate and serves as an unseen dataset for evaluating the model's performance.

In the K-fold cross-validation approach, the dataset is partitioned into K equally sized subsets or folds. Each fold takes turns being the testing set, while the remaining K – 1 folds are used for training the classifiers. This process is repeated K times, with each fold serving as the testing set exactly once. The performance of the classifiers is then evaluated based on the average results obtained across all K iterations.

In both approaches, after applying SMOTE to address class imbalance, we pass the resulting balanced dataset to the classifiers for training and evaluation.

Train–Test Split

This process generates two distinct splits: one portion is designated as the train dataset, while the other is allocated as the test dataset. In our research, we adopt a 70:30 split, where 30% of the dataset is utilized for testing, while the remaining 70% is employed for training. Importantly, both the train and test splits maintain a similar distribution of the "target" variable, ensuring that the class distribution is preserved in both subsets. This approach helps us evaluate the performance of the model effectively and ensure that the results are reliable.

Train Dataset Using K-fold Cross Validation

In the K-fold cross-validation approach, the dataset is divided into k subsets or folds. Each fold, one at a time, is used as the test set, while the remaining k – 1 folds are combined and used as the training set. This process is repeated for each fold, ensuring that each fold serves as the test set once. The models are trained and evaluated on each fold, and their average performance is calculated. This helps in obtaining a more robust and reliable estimation of the model's performance.

When a specific value is chosen for k, such as $k = 10$, it signifies 10-fold cross validation. In this case, the dataset is divided into 10 subsets, and the model is trained and evaluated 10 times, with each fold serving as the test set once. The average performance across these 10 iterations is then calculated to finalize the model.

For our research, we employ a 10-fold cross validation, where the dataset is divided into 10 subsets, and the model is trained and evaluated accordingly. This approach allows for a comprehensive assessment of the model's performance and ensures a more robust evaluation.

IMPROVE PERFORMANCE WITH LDA

LDA is a widely used topic-modeling algorithm that can identify topics within a given dataset. Each document in the dataset is composed of various words, and each topic is represented by a set of words. LDA finds the topics that are most relevant to each document, making it a powerful tool for unsupervised classification and feature extraction. Our research seeks to improve the accuracy of our classification framework by incorporating additional features extracted through LDA.

To accomplish this, we first apply LDA to our dataset to generate feature vectors based on the identified topics. We then add these feature vectors to our existing TF-IDF features and train our dataset using both the train–test and K-fold cross-validation approaches. We evaluate the accuracy of our predictions using performance metrics, such as precision, recall, F1 score, and accuracy.

As part of the LDA preprocessing step, we perform various text-cleaning steps, such as tokenization, removing punctuation and stop words, lemmatization, and stemming. This transforms each breach description into a list of words that can be used for topic modeling.

- *Original document*: "LendKey Technologies, Inc. suffered a breach affecting 6403 records,

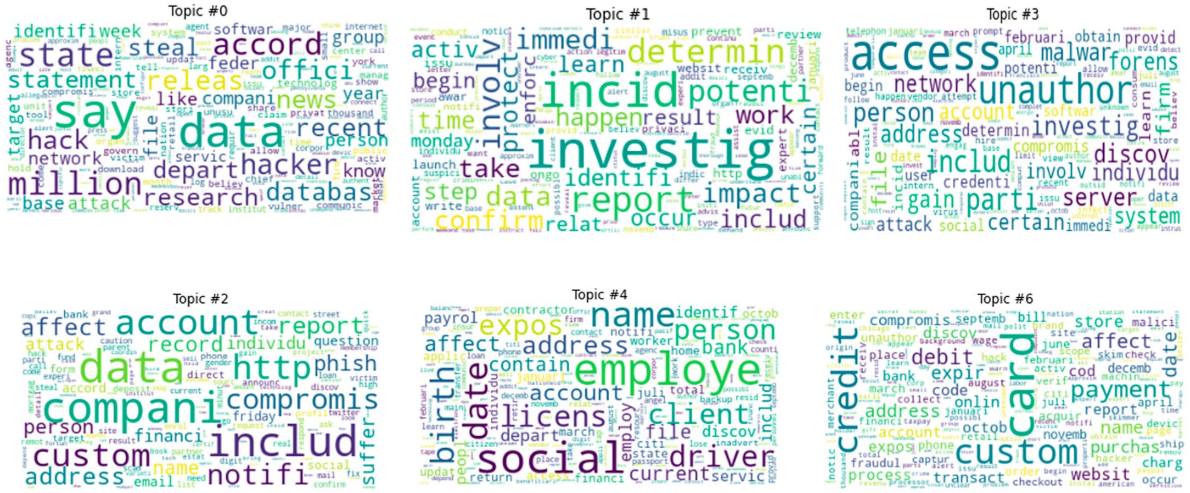


FIGURE 2. Some of the latent Dirichlet allocation topics as a word cloud. (a) Dataset before and after noise removal. (b) Performance of the framework before and after noise removal.

including account numbers, driver's licenses, and SSN [social security numbers]."

- *Tokenized and lemmatized document*: "'lendkey,' 'technolog,' 'suffer,' 'breach,' 'affect,' 'record,' 'include,' 'account,' 'number,' 'driver,' 'licens.'"

Text Classification Using LDA

For each "Description of Incident," we utilize LDA to extract 20 topics. These topics represent key themes or concepts present in the data. To provide a visual representation of these topics, Figure 2 showcases a word cloud that highlights some of the important words associated with each topic. The word cloud helps to visualize the prominent terms within each topic, offering insights into the underlying patterns and themes identified by the LDA algorithm.

After combining the TF-IDF features (which yield 8902 feature vectors) with the 20 topics extracted from LDA, we obtain a total of 8922 feature vectors for each of the 5749 "Description of Incident" instances. These combined feature vectors capture both the textual information represented by TF-IDF and the thematic information represented by LDA topics.

Subsequently, we proceed with the classification task by employing the train–test and k-fold approaches on these feature vectors. The dataset is divided into training and testing sets or into k subsets for k-fold cross validation. The classifiers are trained on the training data, and their performance is evaluated on the test data or through cross validation. This allows us to assess the effectiveness of the combined feature vectors in improving the classification accuracy.

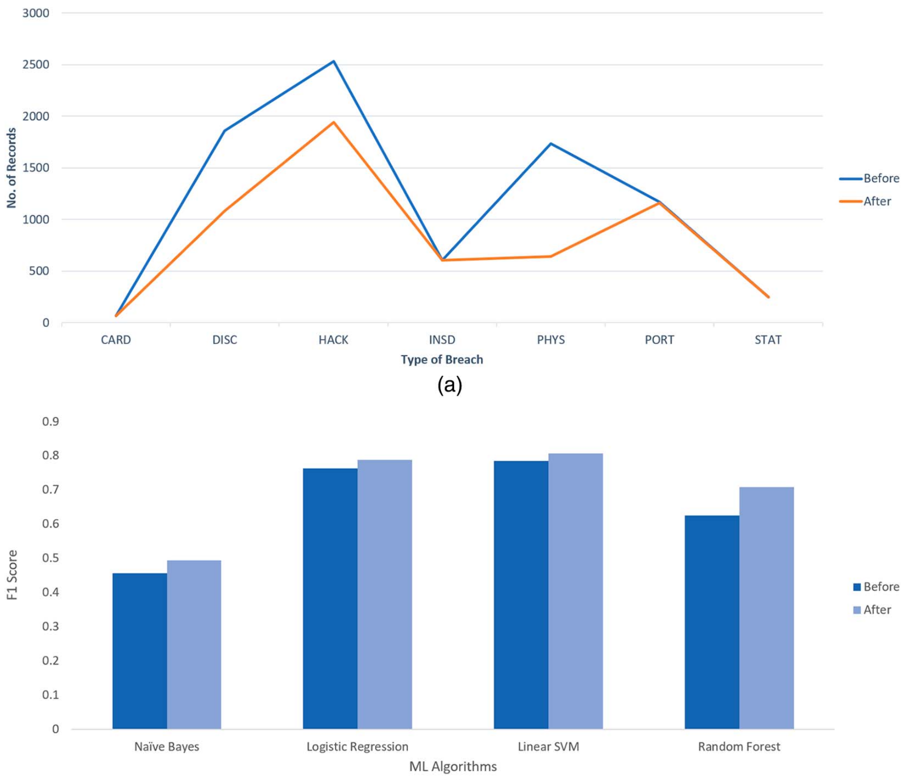
EXPERIMENT RESULT

The original dataset obtained from PRC[1] consists of 8220 records. However, after the removal of redundancies and noise from the dataset, the total number of records is reduced to 5749 [as depicted in Figure 3(a)]. This process aims to improve the quality and reliability of the dataset by eliminating duplicate or irrelevant entries. By eliminating noisy data from the dataset, several benefits are achieved. First, the training time for the framework is significantly reduced. Removing noise reduces the complexity of the data and allows machine learning algorithms to focus on relevant patterns and relationships.

Furthermore, the performance of the framework is improved as a result of noise removal. The F1 score, which is a metric that considers both precision and recall, is commonly used to evaluate the effectiveness of classification algorithms. In Figure 3(b), it can be observed that the F1 score for each of the four machine learning algorithms employed in the research shows improvement. This indicates that the removal of noisy data enhances the accuracy and reliability of the framework, leading to more accurate predictions and classifications.

Data Balancing

The initial dataset exhibits an imbalance between the CARD data type, which has 68 examples, and the HACK data type, which has 1942 examples. This imbalance can potentially lead to misleading results. To address this issue, we employ SMOTE on the minority



(a)

(b)

FIGURE 3. Effects of noise removal on the dataset. (a) Dataset before and after SMOTE. (b) Performance of framework before and after SMOTE. ML: machine learning; SMOTE: synthetic minority oversampling technique.

classes. As a result, each data breach class has an equal number of examples in the dataset [see Figure 4(a)].

The application of SMOTE has a positive impact on all four machine learning algorithms. Particularly, the random forest classifier shows significant improvement. Typically, the random forest classifier struggles when dealing with rare outcomes, which is why data balancing has a considerable influence on its performance [refer to Figure 4(b)].

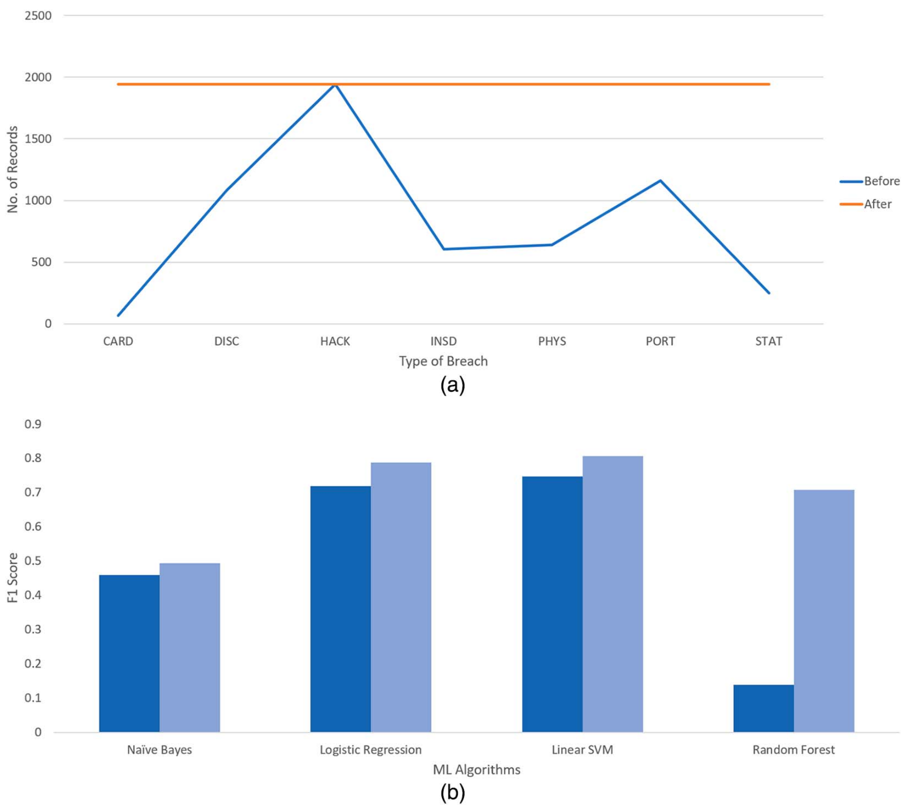
Classification Result

Table 4 presents the F1 scores of various machine learning algorithms obtained through the train–test approach. The findings suggest that when TF-IDF vectors and LDA vectors are used separately, the F1 scores are not as high compared to when the vectors are combined. The performance of LDA vectors alone is relatively unsatisfactory; however, it improves when augmented with TF-IDF vectors. Surprisingly, TF-IDF vectors alone produce good results, but the combination of vectors leads to even better performance.

Table 4 displays the F1 scores obtained from the K-fold cross-validation approach, indicating a slight decrease compared to the train–test approach. In the train–test approach, the F1 score is computed on a single set of trained data, while in the K-fold approach, it represents the average performance across all K folds. Although certain algorithms may exhibit a minor performance decrease with the K-fold approach, the results are deemed more accurate as the model is trained K times more frequently compared to the train–test approach.

Linear SVM emerges as the top performing algorithm among those tested using the train–test approach, employing feature vectors derived from both TF-IDF and LDA. It achieves an impressive F1 score of 80.6% (refer to Table 4). The logistic regression and random forest classifiers also demonstrate excellent F1 scores of 78.7% and 70.7%, respectively, when the training process incorporates combined feature vectors of TF-IDF and LDA. It is worth noting that naive Bayes exhibits slightly lower performance, which is expected due to its simplistic nature, making it less suited for handling correlated data.

Lastly, an effort is made to predict the occurrence of a data breach incident labeled as “UNKN” in the original dataset acquired from PRC.



(a)

(b)

FIGURE 4. Effects of data balancing (SMOTE) on the dataset.

- *Unknown incident*: Customer information was sent to the wrong agent. The format of the information is unknown. The information included names and Medicare numbers.
- *Type of breach found*: `DISC`.

The prediction aligns with the presence of terms related to disclosing information in the dataset. However, it is essential to acknowledge that the actual reason for the data breach incident labeled as “UNKN” (unknown) may differ from disclosure,

TABLE 4. Classification results for different algorithms using the train–test and K-fold approaches.*

Algorithms	TF-IDF vectors	LDA vectors	TF-IDF + LDA Vectors
Classification result for different algorithm using train–test split			
Random forest	0.675941	0.55931	0.707733
Linear SVM	0.780271	0.496663	0.80617
Naive Bayes	0.510096	0.362596	0.493996
Logistic regression	0.769142	0.532612	0.787807
Classification result for different algorithm using K-fold ($K = 10$) cross-validation training approach			
Random forest	0.642876	0.52257	0.663213
Linear SVM	0.747578	0.477582	0.764079
Naive Bayes	0.504978	0.354186	0.499099
Logistic regression	0.720767	0.509295	0.730492

*LDA: latent Dirichlet allocation; SVM: support vector machine; TF-IDF: term frequency–inverse document frequency.

and the prediction represents the closest possible outcome based on the available incident information. It is important to note that machine learning models make predictions based on patterns and correlations in the data, and, while they can provide valuable insights, they are not infallible and should be interpreted with caution. Additional investigation and analysis may be necessary to determine the precise cause of the incident.

CONCLUSION

This article introduces a classification framework designed to categorize data breaches into seven distinct types. Two approaches are proposed: one utilizes text classification with TF-IDF feature vectors, while the other incorporates a topic-modeling technique using LDA to extract feature vectors combined with TF-IDF vectors. The framework is trained and evaluated using both the train–test split (70/30) and k-fold (k = 10) cross-validation methods. The results highlight the improved performance achieved by incorporating LDA features, with the linear SVM classifier from scikit-learn demonstrating the best performance.

Additionally, the framework proves effective in classifying data breach incidents that were previously labeled as unknown, lacking sufficient information about their occurrence and origin. The framework successfully assigns these incidents to one of the seven known data breach types identified in the PRC dataset.

Future research opportunities include exploring and evaluating alternative deep learning algorithms to further enhance classification performance. Furthermore, the researchers plan to investigate datasets from other public platforms, like Stack Overflow and Quora, to expand the model's training and evaluation capabilities.

The classification script used in the framework is available on the GitHub repository.[11] Moreover, an Android application has been developed to showcase the performance of the classification framework.[10]

ACKNOWLEDGMENTS

The authors express their sincere gratitude to all the anonymous reviewers for their invaluable comments and feedback. This article is based on Priyanka Rani's master's degree thesis.[12]

REFERENCES

1. Privacy Rights Clearinghouse. Accessed: Sep. 22, 2020. [Online]. Available: https://privacyrights.org/data-breaches
2. R. Böhme and G. Kataria, "Models and measures for correlation in cyber-insurance," in *Proc. Workshop Econ. Inf. Secur. (WEIS)*, 2006, vol. 2, p. 3.
3. B. Edwards, S. Hofmeyr, and S. Forrest, "Hype and heavy tails: A closer look at data breaches," *J. Cybersecur.*, vol. 2, no. 1, pp. 3–14, 2016, doi: 10.1093/cybsec/tyw003.
4. P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling Extremal Events: For Insurance and Finance*, vol. 33. Berlin, Germany: Springer Science and Business Media, 2013.
5. "Cost of a data breach report 2023," Ponemon Institute, Traverse City, MI, USA, 2023. Accessed: Sep. 05, 2020. [Online]. Available: https://www.ibm.com/security/digital-assets/cost-data-breach-report/
6. A. H. Juma'h and Y. Alnsour, "The effect of data breaches on company performance," *Int. J. Accounting Inf. Manage.*, vol. 28, no. 2, pp. 275–301, 2020, doi: 10.1108/IJAIM-01-2019-0006.
7. T. Maillart and D. Sornette, "Heavy-tailed distribution of cyber-risks," *Eur. Physical J. B*, vol. 75, no. 3, pp. 357–364, 2010, doi: 10.1140/epjb/e2010-00120-8.
8. R. Mishra and R. Singh Thakur, "An efficient approach for supervised learning algorithms using different data mining tools for spam categorization," in *Proc. 4th Int. Conf. Commun. Syst. Netw. Technol.*, Piscataway, NJ, USA: IEEE Press, 2014, pp. 472–477, doi: 10.1109/CSNT.2014.100.
9. D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2009, pp. 248–256, doi: 10.3115/1699510.1699543.
10. P. Rani. "Data breach prediction android." GitHub. Accessed: Jan. 25, 2021. [Online]. Available: https://github.com/priyanka-rani/Data-Breach-Prediction-Android
11. P. Rani. "Data breach text classification." GitHub. Accessed: Sep. 12, 2020. [Online]. Available: https://github.com/priyanka-rani/Data-Breach-Text-Classification
12. P. Rani, "A data driven classification framework for cyber security breaches," Master's thesis, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2021.
13. "2015 reported data breaches surpasses all previous years." DataLossDB. Accessed: Aug. 19, 2020. [Online]. Available: https://blog.datalossdb.org
14. Symantec. *Internet Security Threat Report*, vol. 24. Mountain View, CA, USA: Symantec Corp., 2019. Accessed: Sep. 05, 2020. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-en
15. "2020 data breach investigations report." Verizon. Accessed: Sep. 21, 2020. [Online]. Available: https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf

16. S. Wheatley, T. Maillart, and D. Sornette, "The extreme risk of personal data breaches and the erosion of privacy," *Eur. Physical J. B*, vol. 89, no. 1, pp. 1–12, 2016, doi: 10.1140/epjb/e2015-60754-4.
17. M. Xu, K. M. Schweitzer, R. M. Bateman, and S. Xu, "Modeling and predicting cyber hacking breaches," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2856–2871, Nov. 2018, doi: 10.1109/TIFS.2018.2834227.

PRIYANKA RANI is a senior android engineer with the Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh. Her research interests include cybersecurity and artificial intelligence and topic-based breach analysis. Rani received her M.Sc. degree from Bangladesh University of Engineering and Technology. Contact her at priyanka.shill06@gmail.com.

ABHIJIT KUMAR NAG is a tenured associate professor with the Subhani Department of Computer Information Systems at Texas A&M University–Central Texas, Killeen, TX, 76549, USA. His research interest includes various authentication approaches, mainly continuous authentication, multifactor authentication systems, and evolutionary algorithms. Nag received his Ph.D. degree in computer science from the University of Memphis. He is the representative of Texas A&M University–Central Texas to the semiconductor research for the Texas A&M System collaboration. Contact him at aknag@tamuct.edu.

RIFAT SHAHRIYAR is a professor with the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. His research interests include memory management (garbage collection), programming language, software engineering, and natural language processing. Shahriyar received his Ph.D. degree from the Research School of Computer Science at the Australian National University. Contact him at rifat@cse.buet.ac.bd.