IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 March 2024, accepted 4 April 2024, date of publication 10 April 2024, date of current version 18 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387027

RESEARCH ARTICLE

Korean Voice Phishing Detection Applying NER With Key Tags and Sentence-Level N-Gram

SEUNGUK YU[1], (Graduate Student Member, IEEE), YEJIN KWON[2], MINJU KIM[3], AND KISEONG LEE[4]

[1]Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, South Korea
[2]School of Applied Statistics, Chung-Ang University, Seoul 06974, South Korea
[3]School of Business and Economics, Chung-Ang University, Seoul 06974, South Korea
[4]Humanities Research Institute, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Kiseong Lee (goory@cau.ac.kr)

This work was supported in part by the Ministry of Education of Republic of Korea, and in part by the National Research Foundation of Korea under Grant NRF-2017S1A6A3A01078538.

ABSTRACT Voice phishing is the criminal act of tricking others to transfer funds or to seek financial gain based on personal information obtained illegally. The importance of this crime is recognized worldwide, and technical solutions have been proposed to reduce the increasing damage. In this paper, we propose a process for voice phishing detection in Korean by applying named entity recognition (NER) with *Key Tags* and Sentence-level N-gram. From the perspective of humans, we collect financial counseling texts as non-phishing dataset since the victim confuses voice phishing with them. We carefully select *Key Tags* that can be meaningful for distinguishing voice phishing and financial counseling texts and combine sentence bundles to effectively detect voice phishing. The experimental results, using ten types of machine learning models, showed that maintained results when generalizing information by *Key Tags* and improved results when combining text bundles. We hope that the proposed process can be effectively applied to other criminal scenarios in the future.

INDEX TERMS Voice phishing detection, named entity recognition, N-gram, machine learning.

I. INTRODUCTION

Techniques for voice phishing, also called telephone financial fraud, are increasing in number over time, and the economic and social consequences of phishing are also increasing due to the rapid development of information and communication technology, including mobile networks [1]. Voice phishing is considered to be the process of generating illegal benefits by inducing call respondents to provide personal information [2].

Phishing refers to the crime of attracting users to fake sites that imitate the homepages of financial companies, and extracting information from users [3]. At this time, the users expose their main transaction site password, credit card account, or other sensitive information while the attacker approaches the victim in a subtle and elusive way [4]. According to the FBI, phishing is the only crime among the top five reported crime types, including *extortion* and *identity theft*, for which the number of victims has increased rapidly over the past five years. We confirmed that phishing resulted in four times more victims than *non-payment/non-delivery*, which showed the highest number of victims among other crime types except phishing [5].

Phishing is divided into three types: *smishing* using text messages, *pharming* using mail, and *vishing* using phone calls [6]. Vishing means a combination of voice and phishing, which is different from smishing and pharming, where the suspect's attack ends with a single letter or piece of mail [7]. During the voice phishing call, the suspect steals the victim's personal information or seeks financial gain through a chain of smoke and mirrors. In the case of text or mail, when a suspicious situation is discovered, the victim can share it with others or search for it. In the case of a phone call, however, real-time communication between the suspect and the victim makes it difficult for the victim to know whether the situation is true or not immediately.

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

Indeed, when someone receives a voice phishing call, he or she generally recognizes the situation as a loan-related consultation, not a voice phishing. This is because the conversation proceeds with content related to the victim's economic situation. Therefore, we consider financial counseling texts as non-phishing dataset to classify with voice phishing. It is an attempt to distinguish between voice phishing and financial counseling by using situational similarity from the perspective of humans.

In Korea especially, financial authorities count voice phishing incidents according to the two types; *loan fraud* type which induces the victim to offer a loan by providing favorable conditions, and *non-loan fraud* type which impersonates specific institutions by delivering financial information [8]. We found that some words such as names of financial institutions appear in both types. When suspects want to impersonate a specific bank, for example, they can use various bank names, such as *Nonghyup* or *Kookmin* in Korea. In this situation, we employ named entity recognition (NER) to minimize the impact of specific words on voice phishing detection. NER refers to the process of recognizing names and converting them to predefined entities [9]. We define some entities associated with voice phishing into *Key Tags*, and when some of them appear in the text, we change them to predefined terms. It is not just an attempt to detect voice phishing based on specific words, but an attempt to grasp the context of the conversation.

In the case of smishing or pharming, as we mentioned above, the full text of text or mail can be discovered immediately, while in the case of voice phishing, the full text can be obtained only when the call is terminated. It may be after the phishing damage when the call is over. We noted that it is important to determine whether the call is phishing or not during the conversation. To deal with this, we employ sentence-level N-gram to phone call texts, which combines texts into sentence-by-sentence, but not the whole content. It is an attempt to consider the voice phishing conversation between a suspect and a victim by extending the existing N-gram [10] to a sentence-level.

Our research questions are as follows.

RQ1. Is it helpful to apply NER with *Key Tags* to phone call texts to identify whether a conversation is voice phishing?

RQ2. Is it helpful to apply sentence-level N-gram to phone call texts to identify whether a conversation is voice phishing?

The main contributions of this study are as follows.

- We propose a novel voice phishing detection process by applying NER with *Key Tags* and sentence-level N-gram in the phone call texts.
- We use financial counseling as non-phishing dataset to detect voice phishing from the perspective of humans.
- We present experimental results to answer the above research questions, and we confirmed that the proposed process helps to detect voice phishing.

Our paper consists of the following parts. Section II introduces previous studies for NER and machine learning (ML) methods, especially for detecting deceptions. Section III describes the overall flow of the proposed process including the process of collecting datasets, applying NER with *Key Tags* and sentence-level N-gram, and training ML models to detect voice phishing. In Section IV, the analysis of experimental results is presented and research questions are discussed in Section V. Section VI presents threats to validity. Section VII presents future work for the proposed process, and finally, Section VIII concludes our study.

II. RELATED WORK

A. ML APPROACH FOR DETECTING DECEPTIONS

Prior research have been conducted using ML methods to detect crimes. Initially, Support Vector Machine (SVM) and Boosting algorithms were introduced as a solution for the procurement of spam [11]. Bayesian filtering was also used in mobile phones to detect spam [12]. Because phishing detection has steadily progressed, models using Logistic Regression, Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Random Forests, Neural Networks [13], or k-Nearest-Neighbors (kNN) have been compared in various ways [14].

Especially, Abdelhamid et al. [15] compared various models to determine which model was most effective in preventing the risk of phishing. Based on the feature entity which means whether a website is related to crime, they used the PhishTank dataset which consists of features extracted from phishing and legitimate webpages. They used various models such as C4.5, One Rule, Conjunction Rule, European Digital Rights (EDRi), RIpple-DOwn Rule learner (RIDOR), and so on. They confirmed that the best performance in EDRi and RIDOR which are simply available, showed robustness to texts that have already been fixed and cannot be modified on the webpage. In addition, Lee and Park [1] conducted a real-time detection of voice phishing by distinguishing both voice phishing and daily conversation. They used various models such as SVM, Linear Regression, Random Forest (RF), Decision Tree (DT), and so on. They confirmed that the best performance was observed only for classifiers such as SVM. Although the significance of voice phishing detection was conducted using the traditional ML models in their study, there was a possibility of overfitting in the training, because they only used daily conversation texts as a non-phishing dataset that was not significantly related to voice phishing.

In the perspective of voice spoofing, Javed et al. [16] suggested a unified voice anti-spoofing framework. They used ML models containing DT, and Naïve Bayes to identify the input voice as either authentic or not. Ali et al. [17] also conducted a voice spoofing detection system. They measured the Gammatone cepstral coefficient which is a feature of voice form. On the other hand, our study converts voice phishing phone calls into texts, so is close to a NLP study that doesn't use any features of voice form.

TABLE 1. Comparing our study with previous researches on phishing or crime detection.

Sources	Type	Methods	Datasets
Drucker et al. (1999)	Classifying e-mail as spam or non-spam	ML methods: Boosting (using Decision Trees), Ripper, Rocchio, and Linear SVM's	850 / 2,150 for spam / nonspam messages from AT&T
Gómez et al. (2006)	Classifying SMS as spam or non-spam	ML methods: Naïve Bayes, C4.5, Partial Decision Trees (PART), and SVM	199 / 1,157 for spam / nonspam messages in Spanish and 82 / 1,119 for spam / nonspam messages in English
Abu-Nimeh et al. (2007)	Classifying e-mail as spam or non-spam	ML methods: Logistic Regression, Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), SVM, Random Forest, and Neural Networks	1,171 / 1,718 for spam / nonspam emails
Mccord and Chuah (2011)	Classifying Twitter as spam or non-spam	ML methods: Random Forest, SVM, Naïve Bayes and K-Nearest Neighbors (KNN)	4,435 / 7,293 followers for spammers / legitimate users and 3,535 / 1107 following for spammers / legitimate users
Abdelhamid et al.(2017)	Classifying websites as phishy or legitiamate	ML methods: Bayes Net, C4.5, SVM, AdaBoost, EDRi, OneRule, Conjunctive Rule, and RIDOR	11,000 websites collected from Phishtank and Millersmiles
Lee et al. (2021)	Classifying voice messages as vishing and non-vishing	ML methods: Linear Regression, Random Forest, SVM, Decision Trees, XGB	45 / 500 hours for vishing / nonvishing speech
Ali et al. (2022)	Classifying voice as a live source or a prerecorded	ML methods: Gaussian shallow learning Mixture Model (GMM)	3,014 / 1,710 / 13,306 for training/development/evaluation signals from ASV spoof 2017 database
Our Study	Classifying phone call text as a voice phishing or not	Preprocessing with NER tagging and sentence-level N-gram ML methods: Logistic Regression, SVM, XGBoost, AdaBoost, Decision Tree, Naïve Bayes, Random Forest, Extra Tree, GBM, Light GBM	10,793 sentences from 141 voice phishing phone call, 10,725 sentences from 200 financial counseling phone call

B. NER APPROACH FOR DETECTING DECEPTIONS

Prior research have been conducted recognizing meaningful names to detect crimes, such as the name of a person or an institution. Chau et al. [18] conducted NER tagging on the Police Narrative Reports to identify names that could be used for crime identification. They recognized names of people, addresses, vehicles, drug names, and personal assets. They predicted the entity type most likely to match with the binary labels from the neural network. It was an attempt to detect only some named entities that were determined to be useful for crime identification.

In addition, Arulanandam et al. [19] extracted theft-related information from newspapers. They conducted NER tagging to gather information about certain places and locations. Through this, they classified whether the sentence was crime-related or not by using Conditional Random Fields. They compared the results by using various NER tagging methods such as Stanford NER and LBJ Tagger. It was an attempt to find information related to crime identification using various tagging methods. From the perspective of verbal detection, Kleinberg et al. [20] judged the truthness of the review texts by presenting the following theoretical basis for the use of NER tagging: Those who tell facts include more informational and contextual facts, and those who deceive tend to hide this potential information. They confirmed better performance than the lexicon approach and sentence specificity methods and proved the superiority of NER tagging in determining human speech.

Including related studies on phishing or crime detection mentioned above, especially using ML models to detect, we summarized them in Table 1. This provides and compares what was the purpose of each study, which methods used, and what datasets they collected for experiments on phishing or crime detection.

III. METHOD

In the process of our approach, the caller who made the call is referred to as a *suspect*, and the receiver who answered the call is referred to as a *victim*. Our proposed process to detect voice phishing is shown in Figure 1. The datasets we collected are in Korean, but the following examples are shown in English to help readers understand.

We collect financial counseling texts as non-phishing dataset to classify with voice phishing, which have similar contexts to voice phishing from the perspective of a victim. Then, we employ NER tagging with *Key Tags* to minimize the impact of specific names to detect voice phishing and sentence-level N-gram to consider each sentence bundle from voice phishing texts separately. Finally, we train various ML models to classify voice phishing and financial counseling texts.

A. DATASET

First, we collect voice phishing datasets by crawling the Korean Financial Supervisory Service's online website.[1] When abusive language appeared in a phone call, the victim would immediately realize that the call was voice phishing easily, so we remove the sentences that appeared after the abusive texts. When the name of a person was labeled as unknown for privacy reasons, we replace it with the arbitrary name of a person so that it can be identified during the NER tagging. Above this process, we collect 10,793 sentences from 141 voice phishing cases. The following conversation from the suspect can be found in our voice phishing datasets. The suspect used the specific name of the bank such as *Hyundai* and economic terms to deceive the victim.

And then, we collect financial counseling datasets from AI Hub's open data.[2] We modify the form of texts to suit our needs, e.g. if the conversation contents of the counselor and the recipient were separated in different files, we combine them. Because of the total number of sentences from financial counseling was much higher than in voice phishing, we conduct random under-sampling so that the total number of sentences in the financial counseling fell into the $\pm 1\%$ range of the number of sentences from voice phishing. Above this process, we collect 10,725 sentences from 200 financial counseling cases. The following conversation from the counselor can be found in our financial counseling datasets. The counselor uses the specific name of the bank such as *Hyundai* and economic terms to notify the recipient.

B. NER WITH KEY TAGS

In voice phishing, the suspect deceives the victim to extort money and valuables, and the victim is fascinated by impersonating financial institutions such as banks, or by using difficult financial or economic terms. As shown in Table 2, the suspect attempted to trick the victim skillfully by using the names of financial institutions such as *Hyundai Savings Bank* and economic terms such as *repay*. The use of specific words that represent factual relationships can give the victim the feeling that the conversation is real.

TABLE 2. Texts from our voice phishing datasets.

*" This is Jason who is team leader of **Hyundai Savings Bank**'s fund recovery team. "*
" According to the information, you received the funds on Friday, May 19th, and you were informed that you would repay the full amount of 23 million won. "

The counselors also used the names of financial institutions such as *Hyundai Card* and economic terms such as *withdrawal limit*, as shown in Table 3. We confirmed that some words frequently used in both voice phishing and financial counseling are similar, e.g. words from situations such as asking for personal information. In other words, the

[1] https://www.fss.or.kr

[2] https://aihub.or.kr

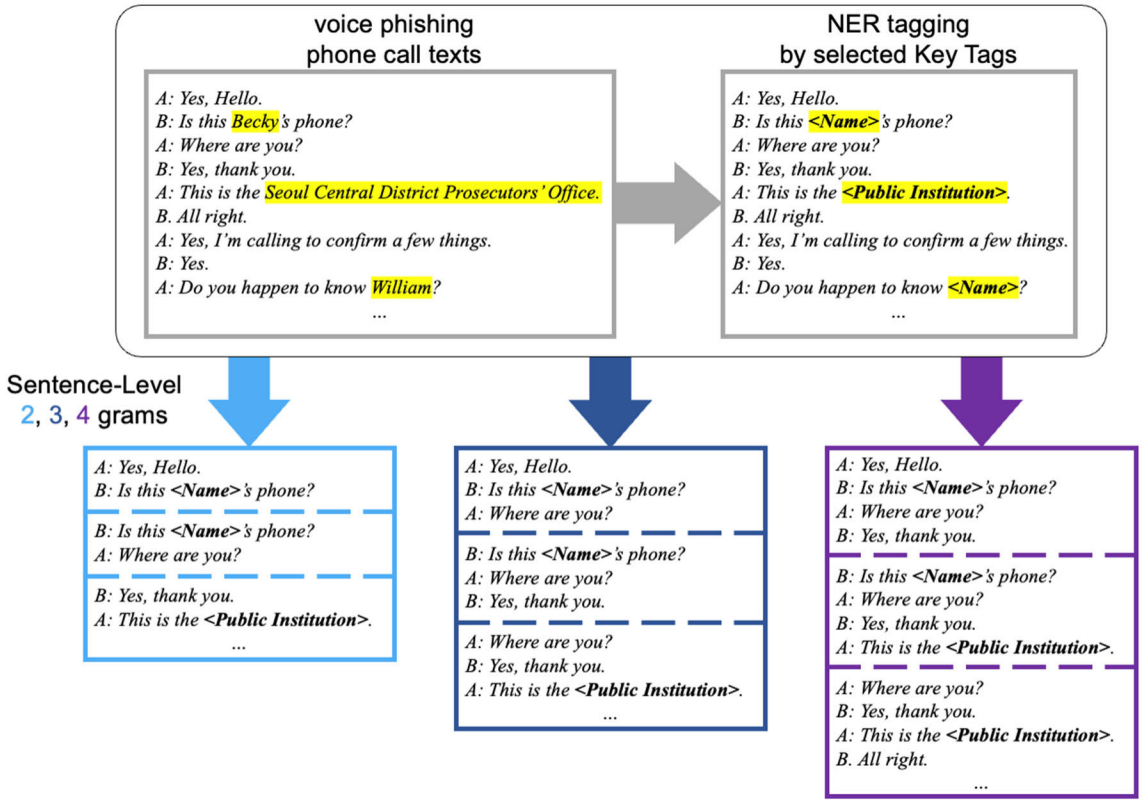

FIGURE 1. Proposed voice phishing detection process applying NER with key tags and sentence-level N-gram.

TABLE 3. Texts from our financial counseling datasets.

*" This is **Hyundai Card** counselor Michael. "*
" In order to prevent financial accidents,
the card withdrawal limit,
which had no recent transactions, will be reduced. "

conversation flows of voice phishing and financial counseling are similar, but the purpose is completely different. It is noteworthy that the name of a specific institution can be changed at any time depending on the context of the suspect's conversation. To allow the voice phishing detection model to flexibly cope with these changes, we specify some names, such as financial institutions or words related to economics, into *Key Tags*.

We use the NER tagging open API system,[3] which follows the tagging standard of the Korea Information and Communications Technology Association. Instead of changing all names that could be converted, we chose only some names useful for voice phishing detection. This was done to prevent the excessive generalization from the NER tagging. First, we listed all tags that appeared in the voice phishing and financial counseling datasets in order of frequency. The results are as follows in Figure 2 and 3.

[3]https://aiopen.etri.re.kr/serviceList

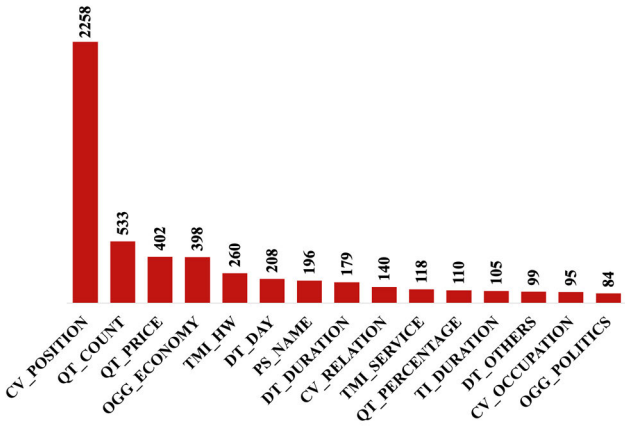

FIGURE 2. NER tags in our voice phishing dataset.

In the voice phishing dataset, the total number of converted tags was 6052, and the total number of categories was 89. CV_POSITION, which means the name of a position, appeared the most frequently at 2258 times, followed by QT_COUNT, which indicates a number or frequency at 583 times. In addition, it can be seen that various tags, such as TMI_HW which represents hardware, and CV_RELATION which represents human relationships.

Next, in the financial counseling dataset, the total number of converted tags was 6848, and the total number of categories

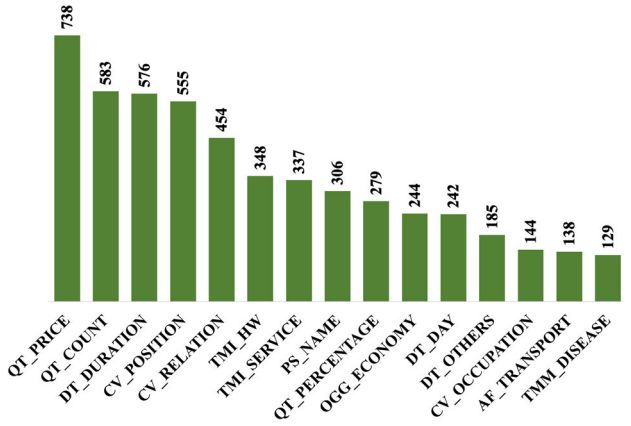

FIGURE 3. NER tags in our financial counseling dataset.

was 103. QT_PRICE which indicates an amount, appeared the most frequently at 738 times, followed by QT_COUNT, which represents a number or frequency at 583 times, similar to voice phishing. In addition, it can be seen that various tags, such as DT_DAY, which means a date, and CV_TAX, which means a tax name.

Through this analysis, we select *Key Tags* that were considered highly relevant to detecting voice phishing. We compare the frequencies of NER tags and exclude some tags that are not important to detect voice phishing. For example, PS_NAME, which indicates the name of a person, was excluded because it rarely appears except at the beginning of a conversation. We include some tags that are determined to be important to detect voice phishing to the *Key Tags* even if the frequency is low. For example, OGG_POLITICS, which indicates a public or political institution, was included in the *Key Tags* because it can convey a sense of reality to the victim simply by the appearance of the word itself. There are *Key Tags* we selected as shown in Table 4.

We divide the group of *Key Tags* to the more important degree: tags containing CV_POSITION to OGG_POLITICS from the top of Table 4 are called NER *Key Tags* 5%, and all the tags in Table 4 are called NER *Key Tags*10%. This was done to see how the voice phishing detection performance varied depending on the degree of NER tagging applied.

C. SENTENCE-LEVEL N-GRAM

Texts from the phone calls of two people are in a form that cannot be completed by one person's speech, and this form contains the following characteristics: its structure varies according to the conversation exchanged by the two people, and its context can be changed depending on who leads the conversation. To reflect these conversational features, we apply sentence-level N-gram, which forms an N-gram at the sentence level rather than at the character- or word-level.

The existing N-gram concept is divided into two types: *Character-based* N-gram, which is a set of consecutive characters extracted from words, and *word-based* N-gram, which is a set of consecutive words extracted from texts [10].

TABLE 4. Selected NER key tags 5% / 10%.

name	description
CV_POSITION	position or one's duty
QT_PRICE	amount of money
OGG_ECONOMY	economic institutions, organizations and businesses
CV_RELATION	human relationship designation
CV_OCCUPATION	occupation name
OGG_POLITICS	government or administrative (public or political) agencies
TMI_HW	IT hardware
DT_DURATION	date duration
TMI_SERVICE	IT service term
TI_DURATION	time duration
CV_LAW	name of a law

Just as it is natural for '*am*' to follow '*I*' (finally '*I am*') and for '*are*' to follow '*you*' (finally '*you are*'), N-gram infer the next letter or word from existing information. This method originated from estimating the present representation using only a few existing records rather than using all.

We extend the concept of N-gram to the *sentence-level* in order to represent the phone call texts. The entire context without sentence-level N-gram could be used, but we consider that voice phishing detection could be applied in a real-time during a call. If only the entire context was used, it is not known whether this situation is voice phishing or not until the phone call is over. Therefore, rather than using the entire context as a single sequence, we attempt to use a combination of sentences according to their temporal order.

We regard that N is the number of some sentences as a bundle. When N was set to 1, it means a bundle consists of only one sentence so the meaning of using sentence-level N-gram is tarnished. When N was set to a number close to the total number of all sentences, on the other side, it means a bundle can be a whole conversation so the meaning of using sentence-level N-gram is also tarnished. Therefore, we set N to an appropriate range of numbers 2, 3, and 4 that do not compromise the above restrictions.

These sentence-level N-gram bundles were embedded using KoSentence-BERT [21] which makes semantically meaningful sentence embeddings to closely map sentences into vector space. We set the same fixed length vector of 768 dimensions, regardless of the length from the input sentence-level N-gram bundles. This model uses KoBERT [22], which reflects the characteristics of Korean, an agglutinative language with more diverse forms than English.

D. MACHINE LEARNING MODELS

We train machine learning models to identify whether a text exhibits characteristics of voice phishing or not.

TABLE 5. Detection f1-scores of voice phishing texts applied NER with Key Tags and sentence-level N-gram.

NER	N-gram	Logistic Regression	SVM	XGBoost	AdaBoost	Decision Tree	Naïve Bayes	Random Forest	Extra Trees	GBM	Light GBM
None	1-gram	0.861	0.861	0.857	0.791	0.742	0.654	0.829	0.834	0.811	0.849
	2-gram	0.927	0.924	0.910	0.844	0.770	0.738	0.870	0.879	0.860	0.896
	3-gram	0.956	0.955	0.940	0.880	0.809	0.774	0.902	0.914	0.896	0.927
	4-gram	**0.974**	**0.973**	**0.960**	**0.909**	**0.833**	**0.808**	**0.928**	**0.939**	**0.919**	**0.948**
Key Tags 5%	1-gram	0.858	0.856	0.849	0.795	0.750	0.666	0.826	0.826	0.809	0.844
	2-gram	0.925	0.923	0.906	0.845	0.776	0.752	0.867	0.877	0.860	0.895
	3-gram	0.954	0.952	0.940	0.885	0.819	0.791	0.901	0.908	0.897	0.925
	4-gram	**0.973**	**0.972**	**0.959**	**0.910**	**0.847**	**0.820**	**0.925**	**0.937**	**0.922**	**0.948**
Key Tags 10%	1-gram	0.857	0.855	0.850	0.789	0.748	0.667	0.823	0.826	0.809	0.842
	2-gram	0.923	0.921	0.909	0.848	0.779	0.753	0.868	0.875	0.862	0.892
	3-gram	0.953	0.953	0.939	0.886	0.817	0.791	0.900	0.908	0.896	0.923
	4-gram	**0.973**	**0.971**	**0.959**	**0.907**	**0.849**	**0.821**	**0.923**	**0.934**	**0.919**	**0.946**

The classification is performed at the level of sentence bundles rather than at the level of individual sentences or cases.

We divide the datasets into train and test sets, and split them into the ratio of 8:2. Due to the limited amount of voice phishing data, we conducted a 5-fold cross-validation, thereby utilizing the average performance across the test set. We used macro-averaged metrics for the experiment.

The description of each model is as follows.

- Naive Bayes [23] is a statistical classification method based on Bayes theorem, which greatly simplifies learning by assuming that each feature is independent in a given class.
- Logistic Regression [24] predicts categorical variables with a linear regression method and is mainly used for classification where dependent variables are binary.
- Support Vector Machine (SVM) [25] uses perceptron-based classification and finds the most stable discriminant boundaries between samples.
- Adaptive Boosting (AdaBoost) [26] increases the weight of samples that were underfitted in the previous model during the concurrent training process of several models and retrains those samples in the next model.
- Decision Tree [27] is a classification model that sequentially applies several attributes and divides the independent variable space and is applicable to both classification and regression.
- Random forest [28] is an ensemble model that combines several independent decision trees.
- ExtraTrees [29] is a form of random forest that uses the entire dataset for training but bootstrapping is not applied.
- Gradient Boost Machine (GBM) [30] is an ensemble model of a decision tree that performs weight updates by applying gradient descent.
- eXtreme Gradient Boosting (XGBoost) [31] is an ensemble model of tree-based algorithms that is based on GBM but addresses its drawbacks: slow performance time and overfit regulation.
- LightGBM [32] is an ensemble model of tree-based algorithms, and unlike GBM, it uses a leaf-wise method rather than a level-wise method and enables faster learning than XGBoost.

IV. EXPERIMENTAL RESULTS

We present the results of the proposed process as shown in Table 5. This only shows f1-scores due to space limitation, and detailed experimental results including accuracy, precision, and recall are in Table 8 at the end of this paper. Looking at the underlined values, applied none-NER tagging with sentence-level 1-gram, they have relatively low f1-scores compared to other cases. However, when NER with *Key Tags* or sentence-level N-grams were applied, the performance was maintained or even improved, respectively.

In terms of classification performance, Logistic Regression and SVM demonstrated outstanding performance among various models. Particularly within the domain of crime situation identification, the concern lies more in failing to detect situations rather than incorrectly detecting them. Therefore, it is more crucial to reduce false negatives than to minimize false positives. Thus, it is important to consider recall alongside the F1 score in model evaluation.

In the aspects of sentence-level N-gram, we confirmed that the f1-scores increased according to the degree of its application. We describe the variation of them only when

using sentence-level N-gram while maintaining none-NER tagging, in order to fairly compare the results of using only sentence-level N-gram. We also present the performance improvement as we scale from 1-gram to 4-gram, as shown in Figure 4.

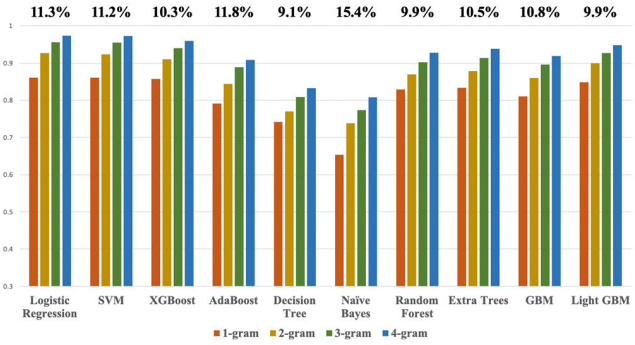

FIGURE 4. Variation of f1-score on sentence-level N-gram.

Naïve Bayes model exhibited the largest improvement at 15.4%, and the other models also showed improvement around 10% with larger n-grams. Therefore, we found that increasing the number of sentences added to the sentence bundle assisted in improving detection performance.

In the aspects of NER with *Key Tags*, we confirmed that the f1-scores remained according to the degree of its application. We describe the variation of them only when using NER with *Key Tags* while maintaining sentence-level 1-gram, in order to fairly compare the results of using only NER tagging, as shown in Table 6. We highlighted the largest and smallest performance changes for each ratio.

TABLE 6. Variation of f1-score on NER with key tags.

Key Tags	Logistic Regression	SVM	XGBoost	AdaBoost	Decision Tree
5%	-0.3%	-0.5%	**-0.8%**	+0.4%	+0.8%
10%	-0.4%	-0.6%	-0.7%	-0.2%	+0.6%
Key Tags	Naïve Bayes	Random Forest	Extra tree	GBM	Light GBM
5%	**+1.2%**	-0.3%	**-0.8%**	-0.2%	-0.5%
10%	**+1.3%**	-0.6%	**-0.8%**	-0.2%	-0.7%

Naïve Bayes model exhibited the largest improvement around 1%, while changes in the other models had little impact. The NER tagging process we performed deliberately transforms the names of texts into predefined entities, so it can be thought of as a loss of information. However, we selected meaningful words into *Key Tags* to detect voice phishing texts against financial counseling texts, and the impact on f1-scores was found to be insignificant.

V. DISCUSSION

RQ1. Is it helpful to apply NER with *Key Tags* to phone call texts to identify whether a conversation is voice phishing?

In response to RQ1, we observed that NER with *Key Tags* did not generally improve performance across all models, but it had only little impact on the experimental results. We trained the meaning of the sentences using entity names like OGG_ECONOMY without relying on specific proper nouns like *Hyundai Savings Bank*. In other words, we found that NER with *Key Tags* has the advantage of responding to changes in various voice phishing scenarios, depending on the context of the suspect's conversation.

To clearly confirm this, we performed a test with voice phishing texts that were not used for the training. There are four cases, that have a time difference of more than two and a half years, so the voice phishing scenario is totally different. We convert voice form into text using a speech-to-text (STT) tool.[4] As in the proposed process, NER with *Key Tags* and sentence-level N-gram were applied, and unlike the original experiment, we conducted voice phishing detection using only Logistic Regression and SVM which performed the best, as shown in Table 7.

For comparison with the original experiment, the underlined values in Table 7 are the same as in Table 5. We observed that when applying NER with *Key Tags* under the same sentence-level N-gram conditions, f1-scores were maintained or slightly increased. In other words, although there are any changes in voice phishing scenarios, NER with *Key Tags* can cope well with various conversations.

TABLE 7. F1-scores of test data with a time difference.

NER	N-gram	Logistic Regression	SVM
Baseline	1-gram	0.861	0.861
None	1-gram	0.888	0.888
	2-gram	0.914	0.972
	3-gram	1.000	1.000
	4-gram	1.000	1.000
Key Tags 5%	1-gram	0.947	0.947
	2-gram	1.000	1.000
	3-gram	1.000	1.000
	4-gram	1.000	1.000
Key Tags 10%	1-gram	0.947	0.947
	2-gram	1.000	1.000
	3-gram	1.000	1.000
	4-gram	1.000	1.000

Unlike the previous study [1] that chose everyday conversations that are not significantly related to voice phishing, we collected financial counseling texts as a non-phishing dataset. So we were able to convert names that frequently appear both in voice phishing and financial counseling. Keeping the model up to date whenever additional voice

[4]https://clova.ai/speech/

phishing data comes out can be a burden. In this respect, we believe that the proposed process can help the model and reduce cost because it can robustly respond to changes in voice phishing scenarios by generalizing the context with the *Key Tags*.

RQ2. Is it helpful to apply sentence-level N-gram to phone call texts to identify whether a conversation is voice phishing?

In response to RQ2, we observed that texts with sentence-level N-gram do improve f1-scores against a single sentence. As the N increases, it means the entire text can be a sentence bundle. However, we intended to detect in real-time conversations where the full text of voice phishing is not yet available. It is necessary to monitor the conversation from time to time, detect abnormalities, and immediately notify the victim. We assumed that the sentences in a conversation should be detected every 10 or 20 seconds, so we set N from 2 to 4 in the proposed process, which needs to be tested more precisely in the practical environment.

While our experiments and two research questions were focused on detecting voice phishing, the practical significance of the proposed process can be adopted in other domains that perform text classification tasks. If sentences share similar contexts while containing common frequent proper nouns, it is possible to train solely on the context through named entity substitution. Proper nouns are generalized, but the context is maintained in the sentences; therefore, models trained on such sentences may be more robust to changes in circumstances. This means that even if it's not a case of voice phishing, it could continue to expand into other domains. In this scenario, our study can be applied in various research utilizing text classification. Additionally, with further advancements in named entity recognition technology, our methodology will be further improved.

VI. THREATS TO VALIDITY

For the effective detection of voice phishing, we were able to choose some different cases in the proposed process including NER with *Key Tags*, sentence-level N-gram, and so on. These are treated as threats to validity which can play an important role in voice phishing detection.

A. INTERNAL VALIDITY

In the aspects of NER with *Key Tags*, we did not convert all names, but only *Key Tags* considering the frequency and how they affect voice phishing. Therefore, there is a concern that the model performance may depend on the selection of them. In the experimental results, there was no significant difference between *Key Tags* 5% and 10%, and neither of them had a significant difference from none-NER tagging. For this reason, it is necessary to further increase the ratio of *Key Tags* to minimize the characteristics of individual names that vary depending on the voice phishing scenarios.

In the aspects of sentence-level N-gram, we conducted with the N from 2 to 4 to directly reflect sentence bundles where calls are made in a real-time. To utilize sentence-level N-gram in a more diverse way, it can be considered using sentence-level 2-gram and 3-gram simultaneously in the manner of utilizing multiple N's instead of setting N as a single value.

B. EXTERNAL VALIDITY

In the aspects of collecting datasets, there were restrictions on gathering voice phishing datasets especially formed in Korean because there are not a lot of voice phishing datasets available. Even if the size of the dataset increases, some criteria for selecting *Key Tags* may also change. Although NER with *Key Tags* and sentence-level N-gram were applied to minimize the impact of specific situations in voice phishing, there may be some limitations in generalizing to all real-world voice phishing situations because the voice phishing datasets we collected only target some kind of financial crimes.

In the aspects of NER tagging, we used the well-known NER tagging tool, so its performance was beyond our study. If more datasets are collected and NER tagging models are built specifically for voice phishing texts, it will be possible to make a meaningful contribution to detecting voice phishing.

VII. FUTURE WORK

We propose the following three situations for future work related to our research. All of them are related to NER with *Key Tags* and sentence-level N-gram to detect each of the text's labels (e.g. whether the texts are voice phishing).

- Applying the proposed process to a real-time voice phishing situation
- Experimenting with the ability to cope with criminal-related textual adversarial attacks
- Extending and exploring meaningful areas for generalizing texts using NER *Key Tags*

First, we aim to detect voice phishing during a real-time phone call as quickly as possible using the proposed process. Since we confirmed the robustness of NER with *Key Tags*, it can be applied to various voice phishing scenarios. In addition, it is possible to further explore ways to predefine *Key Tags* in a more diverse and to utilize them depending on the type of crime and specific scenarios.

When we apply sentence-level N-gram to a real-time, furthermore, the phone call can proceed fast with few words. It is necessary to consider the trade-off in the performance between the size of N and the actual time to obtain sentence bundles. In summary, when we utilize the proposed process, voice phishing can be detected with *Key Tags* that generalize the texts, and quick judgment will be possible through sentence-level N-gram to detect voice phishing.

Second, we plan to apply the proposed process to deal with textual adversarial attacks. Adversarial attacks are noises that are not included in the training stage and are later used as input data. By generalizing given text with

TABLE 8. Detection accuracy, f1-score, precision, and recall of voice phishing texts applied NER with Key Tags and sentence-level N-gram.

		Logistic Regression				SVM				XGBoost				Naïve Bayes			
NER	N-gram	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.
None	1-gram	0.86	0.861	0.856	0.865	0.861	0.861	0.854	0.867	0.856	0.857	0.85	0.863	0.689	0.654	0.582	0.746
	2-gram	0.926	0.927	0.925	0.929	0.923	0.924	0.924	0.925	0.909	0.91	0.913	0.907	0.745	0.738	0.708	0.770
	3-gram	0.956	0.956	0.956	0.957	0.954	0.955	0.955	0.954	0.938	0.94	0.943	0.936	0.779	0.774	0.749	0.802
	4-gram	0.973	0.974	0.973	0.975	0.973	0.973	0.973	0.974	0.96	0.96	0.964	0.957	0.810	0.808	0.787	0.830
Key Tags 5%	1-gram	0.858	0.858	0.853	0.863	0.855	0.856	0.85	0.862	0.85	0.849	0.841	0.858	0.696	0.666	0.600	0.749
	2-gram	0.924	0.925	0.924	0.926	0.922	0.923	0.922	0.925	0.905	0.906	0.907	0.905	0.757	0.752	0.725	0.781
	3-gram	0.954	0.954	0.954	0.955	0.952	0.952	0.952	0.953	0.939	0.94	0.941	0.939	0.794	0.791	0.770	0.814
	4-gram	0.973	0.973	0.972	0.974	0.972	0.972	0.973	0.971	0.958	0.959	0.962	0.956	0.821	0.820	0.803	0.839
Key Tags 10%	1-gram	0.856	0.857	0.852	0.861	0.854	0.855	0.849	0.86	0.85	0.85	0.843	0.857	0.697	0.667	0.602	0.749
	2-gram	0.922	0.923	0.923	0.923	0.92	0.921	0.922	0.921	0.907	0.909	0.909	0.908	0.759	0.753	0.726	0.782
	3-gram	0.953	0.953	0.952	0.954	0.952	0.953	0.954	0.952	0.938	0.939	0.942	0.937	0.793	0.791	0.770	0.813
	4-gram	0.973	0.973	0.972	0.975	0.97	0.971	0.97	0.971	0.958	0.959	0.96	0.958	0.821	0.821	0.803	0.839
		AdaBoost				Decision Tree				Random Forest				Extra Trees			
NER	N-gram	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.
None	1-gram	0.790	0.791	0.788	0.794	0.743	0.742	0.730	0.754	0.831	0.829	0.847	0.812	0.835	0.834	0.846	0.822
	2-gram	0.843	0.844	0.843	0.845	0.770	0.770	0.760	0.779	0.868	0.870	0.863	0.877	0.876	0.879	0.873	0.885
	3-gram	0.878	0.880	0.880	0.880	0.807	0.809	0.807	0.812	0.900	0.902	0.892	0.913	0.912	0.914	0.904	0.924
	4-gram	0.907	0.909	0.909	0.909	0.831	0.833	0.829	0.838	0.926	0.928	0.916	0.941	0.937	0.939	0.928	0.950
Key Tags 5%	1-gram	0.794	0.795	0.793	0.798	0.751	0.750	0.742	0.760	0.827	0.826	0.842	0.810	0.826	0.826	0.838	0.814
	2-gram	0.843	0.845	0.845	0.845	0.776	0.776	0.768	0.785	0.865	0.867	0.861	0.874	0.875	0.877	0.870	0.885
	3-gram	0.883	0.885	0.884	0.885	0.817	0.819	0.820	0.819	0.899	0.901	0.894	0.909	0.905	0.908	0.899	0.917
	4-gram	0.908	0.910	0.909	0.911	0.845	0.847	0.842	0.852	0.923	0.925	0.917	0.934	0.936	0.937	0.929	0.946
Key Tags 10%	1-gram	0.788	0.789	0.786	0.792	0.748	0.748	0.741	0.756	0.825	0.823	0.840	0.806	0.827	0.826	0.839	0.813
	2-gram	0.846	0.848	0.850	0.847	0.777	0.779	0.774	0.784	0.865	0.868	0.860	0.877	0.872	0.875	0.867	0.883
	3-gram	0.884	0.886	0.887	0.885	0.815	0.817	0.813	0.820	0.897	0.900	0.892	0.908	0.905	0.908	0.899	0.917
	4-gram	0.906	0.907	0.904	0.911	0.847	0.849	0.847	0.852	0.921	0.923	0.915	0.933	0.933	0.934	0.927	0.942
		GBM				LightGBM											
NER	N-gram	Acc.	F1.	Pre.	Rec.	Acc.	F1.	Pre.	Rec.								
None	1-gram	0.813	0.811	0.825	0.798	0.850	0.849	0.859	0.840								
	2-gram	0.858	0.860	0.856	0.865	0.895	0.896	0.895	0.898								
	3-gram	0.894	0.896	0.893	0.899	0.926	0.927	0.927	0.927								
	4-gram	0.917	0.919	0.915	0.923	0.946	0.948	0.945	0.950								
Key Tags 5%	1-gram	0.810	0.809	0.823	0.795	0.844	0.844	0.855	0.833								
	2-gram	0.858	0.860	0.860	0.860	0.893	0.895	0.893	0.896								
	3-gram	0.895	0.897	0.895	0.898	0.924	0.925	0.925	0.926								
	4-gram	0.920	0.922	0.919	0.925	0.947	0.948	0.946	0.950								
Key Tags 10%	1-gram	0.810	0.809	0.820	0.799	0.842	0.842	0.852	0.832								
	2-gram	0.860	0.862	0.859	0.865	0.891	0.892	0.893	0.891								
	3-gram	0.894	0.896	0.892	0.901	0.922	0.923	0.923	0.924								
	4-gram	0.917	0.919	0.915	0.922	0.945	0.946	0.943	0.949								

selected *Key Tags* and combining sentences using sentence-level N-gram, we aim to experiment with whether there are significant performance changes when adversarial attacks are involved.

Finally, we will explore the impact of performing NER tagging with specific combinations of entities, such as the *Key Tags* we used. We have established a set of common *Key Tags* that help us distinguish between financial counseling and voice phishing, and we have seen that the performance is maintained even as we further generalize the text by increasing the ratios of them. Our goal is to extensively study whether this phenomenon can be applied to other tasks and domains for broader applications.

VIII. CONCLUSION

We propose applying NER with *Key Tags* and sentence-level N-gram to detect voice phishing. From the perspective of humans, we collected financial counseling as a non-phishing dataset and attempted to reduce the impact of common words that frequently appear, such as the specific institution's name. We applied NER with *Key Tags*, which were selected in consideration of their influence. We also applied sentence-level N-gram using sentence bundles, not the entire conversation. This has the advantage that the model trained from the proposed process can flexibly and robustly cope with various voice phishing scenarios.

However, our experiments were based on the texts written in Korean, so it is necessary to explore whether there are significant observations when applied to various languages such as English. Additionally, our research is constrained by the performance of named entity recognition systems or sentence embedding technologies such as Sentence Transformers. As the performance of these systems improves, our research can also be enhanced accordingly.

When the relationship between the two texts is similar as well as voice phishing and financial counseling texts, our proposed process can be a good way to generalize and classify them, even if they are from different domains and even they are involved in a crime. Therefore, we hope that our work will be significant to the process of crime detection in other languages, particularly when dealing with sensitive texts that may contain victims' personal information.

ADDITIONAL EXPERIMENTAL RESULTS

See Table 8.

REFERENCES

[1] M. Lee and E. Park, "Real-time Korean voice phishing detection based on machine learning approaches," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 7, pp. 8173–8184, Jul. 2023.

[2] E. Yeboah-Boateng and P. M. Amanor, "Phishing, SMiShing & vishing: An assessment of threats against mobile devices," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 5, no. 4, pp. 297–307, 2014.

[3] H. Yoon and D. Kwak, *Study on Prevention and Countermeasure of Voice Phishing*. Korean Inst. of Criminology, 2009.

[4] L. Chen-Wilson, A. M. Gravell, and D. Argles, "Giving you back control of your data digital signing practical issues and the eCert solution," in *Proc. World Congr. Internet Secur.*, Feb. 2011, pp. 93–99.

[5] *Internet Crime Report*, FBI, Internet Crime Complaint Center, Washington, DC, USA, 2021.

[6] M. H. Chai and C. H. Park, "Study on prevention and countermeasure of voice phishing," *Korean Assoc. Public Saf. Criminal Justice*, vol. 27, no. 3, pp. 417–448, Aug. 2018.

[7] G. Ollmann, "Understanding X-morphic exploitation," SC Media, 2007. [Online]. Available: https://www.scmagazine.com/perspective/understanding-x-morphic-exploitation

[8] S. Hwang, "A study on the eradication of telecommunication financial fraud, focused on the voice phishing," *J. Police Sci.*, vol. 21, no. 1, pp. 91–123, 2021.

[9] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 2020.

[10] M. Prasenjit, M. Mitra, and B. B. Chaudhuri, "N-Gram: A language independent approach to IR and NLP," in *Proc. Int. Conf. Universal Knowl. Lang.*, 2002.

[11] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.

[12] J. M. G. Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, "Content based SMS spam filtering," in *Proc. ACM Symp. Document Eng.*, Oct. 2006, pp. 107–114.

[13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Work. Groups 2nd Annu. eCrime Researchers Summit*, Oct. 2007, pp. 60–69.

[14] M. Mccord and M. Chuah, "Spam detection on Twitter using traditional classifiers," in *Proc. Int. Conf. Autonomic Trusted Comput.*, 2011, pp. 175–186.

[15] N. Abdelhamid, F. Thabtah, and H. Abdel-jaber, "Phishing detection: A recent intelligent machine learning comparison based on models content and features," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 72–77.

[16] A. Javed, K. M. Malik, H. Malik, and A. Irtaza, "Voice spoofing detector: A unified anti-spoofing framework," *Expert Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116770.

[17] D. Ali, S. Al-Shareeda, and N. Abdulrahman, "Low-key shallow learning voice spoofing detection system," in *Proc. 4th IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, Dec. 2022, pp. 77–82.

[18] M. Chau, J. J. Xu, and H. Chen, "Extracting meaningful entities from police narrative reports," in *Proc. Annu. Nat. Conf. Digital Government Res.*, 2002, pp. 1–5.

[19] R. Arulanandam, B. T. R. Savarimuthu, and M. A. Purvis, "Extracting crime information from online newspaper articles," in *Proc. 2nd Australas. Web Conf.*, 2014, pp. 31–38.

[20] B. Kleinberg, M. Mozes, A. Arntz, and B. Verschuere, "Using named entities for computer-automated verbal deception detection," *J. Forensic Sci.*, vol. 63, pp. 714–723, May 2018.

[21] *Sentence Transformer*. Accessed: Aug. 2023. [Online]. Available: https://github.com/BM-K/KoSentenceBERT-ETRI

[22] *KoBERT*. Accessed: Aug. 2023. [Online]. Available: https://github.com/SKTBrain/KoBERT

[23] I. Rish, "An empirical study of the naive Bayes classifier," in *Proc. IJCAI Workshop Empirical Methods Artif. Intell.*, 2001, pp. 41–46.

[24] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Informat.*, vol. 35, nos. 5–6, pp. 352–359, Oct. 2002.

[25] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, no. 1, pp. 3–24, 2007.

[26] T.-K. An and M.-H. Kim, "A new diverse AdaBoost classifier," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, vol. 1, Oct. 2010, pp. 359–363.

[27] Y. Y. Song and L. U. Ying, "Decision tree methods: Applications for classification and prediction," *Shanghai Arch. Psychiatry*, vol. 27, no. 2, p. 130, 2015.

[28] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[29] M. W. Ahmad, J. Reynolds, and Y. Rezgui, "Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees," *J. Cleaner Prod.*, vol. 203, pp. 810–821, Dec. 2018.

[30] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers Neurorobotics*, vol. 7, p. 21, 2013.

[31] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, and K. Chen, *XGBoost: EXtreme Gradient Boosting*, document Version 0.4-2, 2015.
[32] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electron. Commerce Res. Appl.*, vol. 31, pp. 24–39, Sep. 2018.

SEUNGUK YU (Graduate Student Member, IEEE) received the B.S. degree from the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea, in 2023. He is currently pursuing the M.S. degree with the Department of Artificial Intelligence, Chung-Ang University. His research interests include natural language processing, especially in Korean, data anlaytics, and machine learning.

YEJIN KWON received the B.S. degrees in applied statistics and computer science from Chung-Ang University, Seoul, South Korea, in 2024. Her current research interests include data analysis, natural language processing, financial analysis, and machine learning.

MINJU KIM received the B.S. degree in applied statistics from the School of Business Administration, Chung-Ang University, Seoul, South Korea, in 2024. Her current research interests include business analytics, natural language processing, and management science.

KISEONG LEE received the B.S. degree in Korean literature in classical Chinese from Sungkyunkwan University, Seoul, in 2005, and the M.S. and Ph.D. degrees in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2011 and 2015, respectively. Between 2006 and 2008, he was a Software Engineer with OnNet, a internet service company, Seoul. Between 2017 and 2022, he was an Assistant Professor with the Da Vinci College of General Education, Chung-Ang University. He is currently an Assistant Professor with the Humanities Research Institute, Chung-Ang University. His research interests include software architecture, machine learning, and natural language processing.