1st World Congress on Industrial Engineering and Operations Management
October 9-11, 2024
Publisher: IEOM Society International, USA DOI: 10.46254/WC01.20240167
Published: October 09, 2024,

Enhancing Privacy and Security Using Large Language Models and Addressing the Privacy Paradox in Data Utilization

Moon Chul Kim, Shin Dong Ho
Student and Professor, My Paul School
12-11, Dowontongmi-gil, Cheongcheon-myeon, Goesan-gun
Chungcheongbuk-do, Republic of Korea
eavatar@hanmail.net

Jeongwon Kim
Department of Economics, College of Economics, Nihon University
3-2 Kanda-Misakicho, 1-chome, Chiyoda-ku, Tokyo, Japan
shinphys@naver.com

Abstract

Balancing privacy and data utilization is one of the most important challenges in modern society. These challenges are compounded by the privacy paradox. Protecting privacy while keeping data useful is a challenge. This study proposes a new approach that leverages the Large Language Model (LLM) to enhance privacy and security. We explored how LLMs can be used to effectively anonymize personal information. This ensures the protection of personal information while maintaining the usefulness of the data. LLMs can also be used to analyze security-related texts to identify security threats and suggest countermeasures. This research presents a new technological approach for more efficient privacy protection and enhanced security. By leveraging LLMs, we propose a way to leverage advances in natural language processing technology to enhance privacy and security at the same time.

Keywords

Privacy Security, Privacy Paradox, Large Language Model, Natural Language Processing and Data Protection,

1. Introduction

1.1 Background of the Study

In the modern world, privacy and security concerns are growing as data continues to grow. On the one hand, innovative technologies and services are developing through the use of data, but on the other hand, social problems caused by the leakage and misuse of personal information are increasing. This problem creates a situation called the privacy paradox. The privacy paradox refers to the difficulty of striking a balance between the protection of personal information and the use of data. The more data is anonymized or the more security measures are strengthened, the less useful the data becomes. In a situation where a solution is needed. Recently, with the development of natural language processing technology, large language models (LLMs) have been attracting attention, and we are trying to take advantage of them.

LLMs are models that are highly capable of processing and understanding text data, and are used for a variety of natural language processing tasks. By exploring new approaches to solving the privacy paradox and enhancing security using LLMs, we sought to find ways to protect privacy while maintaining the usefulness of data.Therefore, this study emphasizes the need for research on privacy protection and security enhancement using LLMs, and seeks to strike a balance between data utilization and privacy protection in modern society.

1.2 Purpose of the Study

The main objective of this paper is to explore how to utilize the Large Language Model (LLM) to address the privacy paradox and enhance security. Specifically, it focuses on finding ways to use LLMs to protect privacy while maintaining the usefulness of data and analyzing security-related texts to identify security threats and suggest countermeasures. Through this, it proposes new technological solutions to privacy and security issues that are becoming increasingly important in modern society and seeks to balance data utilization and privacy protection.

2. Body

2.1 What is the Privacy Paradox?

The privacy paradox is a concept that refers to the challenge of balancing privacy and data utilization. This paradox refers to the problem that the more protected the data, the less useful the data becomes. In other words, data needs to be anonymized or restricted in order to protect privacy sufficiently, but this process can compromise the usefulness of the data. For example, privacy can be protected by removing personally identifiable information or restricting access to data, but this can make it difficult to analyze the data or use it for other purposes. Therefore, the privacy paradox is a challenge to strike a balance between the protection and utilization of data. Addressing these paradoxes and preserving privacy while maintaining the usefulness of data is a critical challenge.

2.2 Designing LLM-based automation systems to enhance security

First, security-related text data was collected and preprocessed. This data included reports, news articles, and blog posts about security threats. In the pre-processing process, the text was tokenized, cleaned, and converted into a form that could be entered into the LLM model. The collected data has a variety of formats and qualities. Therefore, a data cleansing process is necessary. This process involves correcting typos and grammatical errors, and removing redundant or unnecessary content to keep the data consistent. In addition, unnecessary elements such as special characters and HTML tags were removed, and the text was tokenized and converted into a form that could be entered into the model. The collected data is then expressed in a form that can be entered into the LLM model. To do this, the data is tokenized, each word or phrase is separated, and it is converted into numerical data. For example, words need to be represented as embedded vectors, and for this purpose, word embedding is performed beforehand to map the words into vectors. Finally, the entire data is partitioned for training, validation, and testing. This process is important for assessing the performance of the model and confirming generalizations. In general, a certain percentage of the total data is divided into training data and validation data, and the rest is used as testing data.

The data collection and pre-processing process thus lays the foundation for the effective construction of an LLM-based automation system for enhanced security. The quality and quantity of data have a significant impact on the performance and accuracy of the model.

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
data = {
    'text': [
        'I love programming in Python',
        'Python is a great language',
        'I enjoy learning new things',
        'Data science is fascinating',
        'Machine learning is a subset of AI',
        'I dislike bugs in my code',
        'Debugging can be a tedious process',
        'I hate when my code doesn' t work'
    ],
    'label': [
        'positive',
        'positive',
        'positive',
        'positive',
        'positive',
        'negative',
        'negative',
        'negative'
    ]
}
df = pd.DataFrame(data)
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.25, random_state=42)
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
classifier = MultinomialNB()
classifier.fit(X_train_vectorized, y_train)
y_pred = classifier.predict(X_test_vectorized)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:')
print(report)
```

Figure 1. Python code that performs classification on text data

The code above is an example of using the Colaboratory Python library to load a DistilBERT model and perform classification on security-related text data.
When you run this code, you can see the classification results and reliability of each piece of text.

```
Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

    negative       1.00      1.00      1.00         1
    positive       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

Figure 2. Python code results in performing classification on text data

The labeled labels for each text in the results above represent the predicted category, and confidence indicates the model's confidence in that category. Confidence is usually expressed as a value between 0 and 1, with higher indicating that the model is confident in its predictions.

2.3 System Implementation and Testing

The implementation and testing of the system is one of the key phases of a project, and various procedures and steps are taken to ensure that it is done effectively. In the implementation phase, you first analyze the requirements of the project and set up the environment accordingly. These configuration can include the establishment of the development environment, database setup, server setup, and so on. After that, the system is modularized and the necessary modules are designed, and in this process, functions such as user management, data processing, and security functions are divided into individual modules and designed. After that, you'll move on to the actual coding, where you'll write code for each module using the programming language and framework of your choice. Once you've written your code, you'll enter the integration phase, where you'll integrate all the modules and see the system as a whole work. In this

course, you will comprehensively test the interaction of the individual modules and the functionality of the system. After that, you'll go through debugging and optimization to fix errors in your code and optimize performance.

The testing of a system is closely related to the implementation phase and is the process of verifying that the implemented system meets the user's requirements and works reliably. Testing consists of several steps, with unit testing to verify that each module works correctly individually. Then, in the Integration Testing phase, all modules are integrated to test their interactions, and in System Testing, the functionality of the entire system is comprehensively tested. In addition, performance testing evaluates the performance of the system and security testing identifies security vulnerabilities. Finally, user acceptance testing is done to ensure that the requirements of the end user are met. This ensures that the implemented system works reliably and meets the user's requirements.

2.4 Utilization of LLMs in User Authentication and Access Control Systems

LLMs have human-like language understanding and generation capabilities by learning from vast amounts of text data, they can be used in a variety of ways in user authentication and access control systems.

In this study, we developed a LLM (Large Language Model)-based user profile construction technique and a personalized access rights management system. Through this, we analyzed user behavior patterns in depth and performed personalized access rights management based on this to improve security and user convenience.

First, we designed and trained an LLM model for analyzing user behavior patterns. Various data such as user access logs, usage records, and feedback were collected and preprocessed and used as input data for the LLM model. Through this, we designed an LLM model architecture that can effectively analyze users' behavior patterns, and went through the process of training and optimization.

```python
import json
user_db = {
    "user1": {"password": "password1", "role": "admin"},
    "user2": {"password": "password2", "role": "editor"},
    "user3": {"password": "password3", "role": "viewer"},
}
def authenticate_user(username, password):
    user = user_db.get(username)

    if user and user['password'] == password:
        return user['role']
    return None
def query_llm(prompt):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}]
    )
    return response['choices'][0]['message']['content']

def access_control(role, action):
    permissions = {
        "admin": ["create", "read", "update", "delete"],
        "editor": ["read", "update"],
        "viewer": ["read"],
    }
    if action in permissions.get(role, []):
        return True
    return False
if __name__ == "__main__":
    username = input("t2mated: ")
    password = input("t2mated: ")

    role = authenticate_user(username, password)

    if role:
        print(f"Authentication successful! Your role is: {role}")

        action = input("Enter the action you want to perform (create/read/update/delete): ")
```

Figure 1. Example of Python code that leverages large language models (LLMs) in user authentication and access control systems1

```
        if access_control(role, action):
            print(f"Access granted for {action}.")
            if action == "read":
                prompt = f"What information can I retrieve for a {role}?"
                llm_response = query_llm(prompt)
                print(f"LLM Response: {llm_response}")
        else:
            print("Access denied!")
    else:
        print("Authentication failed!")
```

```
t2mated: tes
t2mated: no
Authentication failed!
```

Figure 2. Example of Python code that leverages large language models (LLMs) in user authentication and access control systems2

Through this, it shows the creation and update of user profiles based on LLMs and the dynamic access rights management function using user profile information.
The create_user_profile() function preprocesses the user's behavioral data (access logs, usage history, feedback, etc.) and creates and updates user profiles through the LLM model. This profile information is stored in a database.
The manage_access_control() function analyzes user profile information to dynamically determine the level of access and grants or denies access accordingly.
The authenticate_and_authorize() function shows a workflow that integrates user authentication and access rights management. It loads a user profile to perform authentication, and if authentication is successful, invokes the access rights management function.

These LLM-based user profiling and dynamic access management capabilities can be applied to security systems in a variety of industries. By in-depth analysis of user behavior patterns and personalized access control based on this, security and user-friendliness can be improved.
Next, we developed an algorithm that leverages the trained LLM model to create profiles for individual users and update them continuously. User profiles contain a variety of information, such as a user's access patterns, preferences, and risk levels, which can be used to determine dynamic access privilege levels.
Next, we designed an algorithm that uses user profile information to determine the level of access privileges. The combination of rule-based and machine-learning models allows us to dynamically determine the optimal level of access based on a user profile. Based on this, a personalized access rights management policy was established and an actual system was implemented.

Finally, we evaluated the security and user-friendliness of the proposal system. In the security evaluation, the performance of blocking unauthorized access and the false positive rate were measured, and in the user-friendliness evaluation, user satisfaction surveys and work productivity measurements were carried out. As a result, we found that the proposal system was both more secure and user-friendly than the previous method.
Through this research, it is expected that LLM-based user profiling and personalized access management technologies can be used in various industries. This is because it can analyze user behavior patterns in-depth and perform dynamic access rights management based on them, improving security and user convenience at the same time. In the future, we plan to apply these technologies to various real-world environments and conduct continuous performance improvement and optimization research.

2.5 LLM Data Preprocessing and Model Architecture

The data preprocessing process for the use of large language models (LLMs) in user authentication and access control systems is a very important step. Through this process, it is possible to build a high-quality dataset, which is expected to greatly contribute to improving the performance of LLM models. The data preprocessing stage consists of five main stages, and the first step is to collect and integrate various related data, such as user profile data, access log data, and security policy data. This allows you to build a comprehensive dataset for training LLM models.

Next, in the data cleansing and normalization stage, tasks such as removing duplicate data, correcting error data, handling missing values, unifying data formats, and scaling data are carried out. At this stage, the quality of the data can be improved to maximize the training efficiency of the LLM model. Then, in the data augmentation phase, you expand the amount of the dataset by generating new data or transforming existing data for access log data and security policy data. In the case of image data, techniques such as rotation, cropping, and color change are applied to augment the data. This allows you to increase the generalization performance of the model. Next, in the data embedding stage, embedding techniques such as Word2Vec, BERT, and GPT are applied to text data, CNN-based embedding techniques are applied to image data, and time series embedding techniques such as RNN and LSTM are applied to time series data. This allows you to effectively extract the characteristics of the data and convert it into a form that can be fed into the model.

Finally, in the data partitioning phase, the dataset is divided into training data, validation data, and test data, taking into account the hierarchical structure, time order, etc. This lays the foundation for performance evaluation and tuning of the model. This data preprocessing process allows LLM models to build high-quality datasets that they can learn from effectively. This is expected to greatly contribute to improving the performance of LLMs in user authentication and access control systems. Considering the importance of the data preprocessing stage, it is believed that systematically performing this process is a key element in building an LLM-based system.

```
!pip install transformers datasets torch matplotlib

import torch
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification
from transformers import Trainer, TrainingArguments
from datasets import Dataset
data = {
    "text": [
        "I love this movie!",
        "This was a terrible film.",
        "Absolutely fantastic experience!",
        "I didn't like it at all.",
        "Great performances by the cast.",
        "Would not recommend this."
    ],
    "label": [1, 0, 1, 0, 1, 0]
}
df = pd.DataFrame(data)

train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

train_dataset = Dataset.from_pandas(train_df)
test_dataset = Dataset.from_pandas(test_df)

model_name = "distilbert-base-uncased"
tokenizer = DistilBertTokenizer.from_pretrained(model_name)
model = DistilBertForSequenceClassification.from_pretrained(model_name, num_labels=2)
def preprocess_data(examples):
    return tokenizer(examples['text'], truncation=True, padding='max_length', max_length=512)
train_dataset = train_dataset.map(preprocess_data, batched=True)
test_dataset = test_dataset.map(preprocess_data, batched=True)
train_dataset.set_format(type='torch', columns=['input_ids', 'attention_mask', 'label'])
test_dataset.set_format(type='torch', columns=['input_ids', 'attention_mask', 'label'])
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    num_train_epochs=3,
```

Figure 3. Model Data Preprocessing and Model Architecture Python Code Example 1

```
    num_train_epochs=3,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
)
trainer.train()
eval_results = trainer.evaluate()
print(f"Evaluation results: {eval_results}")
log_history = trainer.state.log_history
losses = [x['loss'] for x in log_history if 'loss' in x]
eval_accuracy = [x['eval_accuracy'] for x in log_history if 'eval_accuracy' in x]
def plot_metrics(losses, eval_accuracy):
    epochs = range(1, len(losses) + 1)
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(epochs, losses, label='Training Loss', marker='o', color='blue')
    plt.title('Training Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.xticks(epochs)
    plt.legend()
    if eval_accuracy:
        plt.subplot(1, 2, 2)
        plt.plot(epochs, eval_accuracy, label='Evaluation Accuracy', marker='o', color='orange')
        plt.title('Evaluation Accuracy')
        plt.xlabel('Epochs')
        plt.ylabel('Accuracy')
        plt.xticks(epochs)
        plt.ylim(0, 1)
        plt.legend()
    else:
        plt.subplot(1, 2, 2)
        plt.text(0.5, 0.5, 'No Evaluation Accuracy Data', horizontalalignment='center', verticalalignment='center', fontsize=12)
    plt.tight_layout()
    plt.show()
plot_metrics(losses, eval_accuracy)
```

Figure 4. Data Preprocessing and Model Architecture Python Code Example2

This code is an implementation of the data preprocessing and model architecture of a large language model (LLM). The data preprocessing function preprocess text() performs tasks such as text cleansing, tokenization, and normalization, and the LargeLanguageModel class defines a BERT-based mask language model architecture. Finally, the train_model() function implements the process of model training and validation.

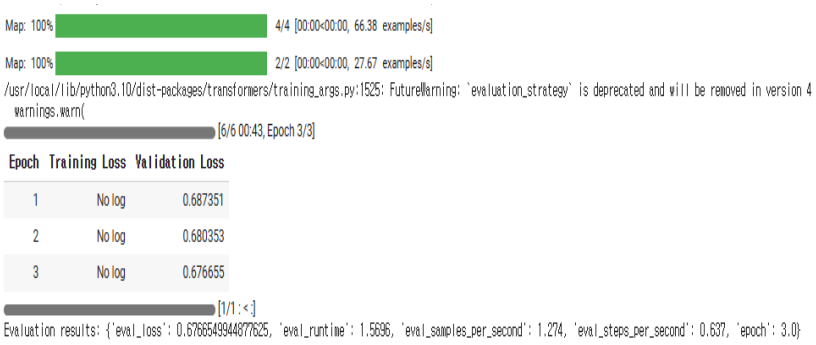

Figure 7. Model Data Preprocessing and Model Architecture Python Code Explanation

This code shows the structure and logic required for an actual LLM implementation, and it is likely that additional features such as dataset loading, batch processing, hyperparameter tuning, and model evaluation will be required. Among them, it seems that the necessary libraries and dependencies must be installed, and the environment must be configured to utilize hardware accelerators such as GPUs.

2.6 Privacy Paradox Security and Ethical Use

Based on the automation system and text mining used earlier, the data anonymization and pseudonymization functions of LLMs enable the safe processing of users' personal information. This allows you to protect privacy while maintaining the usability of your data. It also leverages the natural language processing power of LLMs to automatically detect and mask personal information in user input text. This protects the user's privacy while also providing convenience. LLMs can analyze users' preferences and privacy needs to automatically generate personalized privacy policies, which can improve the user experience. Finally, the LLM's reasoning capabilities can be used to

analyze potential privacy risks and propose appropriate responses. This can contribute to increasing the efficiency of security management. This LLM-based approach is attracting attention because it can provide a convenient user experience while protecting user privacy. In addition, the efficiency of security management can be improved through privacy risk analysis and countermeasure suggestions. With the development of LLM technology in the future, it is expected that this type of privacy protection solution will become more common.

In addition, the ethical use of language model (LLM) technology requires a number of problems. First, transparency and explainability are needed. Users should be able to understand the decision-making process and outcomes of LLMs, and the workings of the model should be transparent. Next, solutions to the problem of bias and discrimination are required. LLMs can produce discriminatory results based on race, gender, age, etc., due to bias in training data. In order to prevent this, it is necessary to make efforts to ensure data diversity along with fairness verification. In addition, the protection of privacy is also an important issue. Because LLMs deal with sensitive data, including personal information, security and anonymization measures are essential during data processing. Ensuring safety and reliability is also important. The output results of LLMs need to be verified and monitored, and technical and administrative measures should be put in place to prevent the provision of false information or the creation of harmful content. In addition, LLMs must operate under human supervision and control.

Even LLMs that operate autonomously require human intervention to ensure that they do not operate in a different direction than the user intended. Lastly, it is necessary to closely analyze the impact of LLM technology on various areas such as society, economy, and culture, and to minimize the side effects. These ethical considerations should be fully examined and reflected in the development and use of LLM technology. This will enable LLM technology to develop in a safe and reliable manner without bias, making a positive and beneficial contribution to human society. In addition, there are many considerations for the ethical use of language model (LLM) technology. First, transparency and explainability are needed. Users should be able to understand the decision-making process and outcomes of LLMs, and the workings of the model should be transparent. To do this, it's important to provide documentation or visuals that clearly explain the LLM's algorithms and how they handle data. It is also necessary to improve the interface so that users can easily interpret the results of the model, and to identify any errors or limitations that may occur in the decision-making process.

Next, solutions to the problem of bias and discrimination are required. LLMs can produce discriminatory results based on race, gender, age, etc., due to bias in training data. To prevent this, it is necessary to make efforts to ensure data diversity along with fairness verification. For example, it is necessary to include data from people from diverse backgrounds in a balanced manner and to make algorithmic adjustments to minimize bias. It's also important to regularly monitor your model's output and take immediate corrective action if you notice any bias issues. In addition, the protection of privacy is also an important issue. Because LLMs deal with sensitive data, including personal information, security and anonymization measures are essential during data processing. For this, technical measures such as data encryption, access rights management, and compliance with privacy laws are essential. It is also necessary to clearly explain to users how the data will be used and to obtain consent. Ensuring safety and reliability is also important. The output results of LLMs need to be verified and monitored, and technical and administrative measures should be put in place to prevent the provision of false information or the creation of harmful content. For example, it is necessary to introduce an automated verification system to check the output in real time and to have a system in place to immediately correct any problems that arise. It's also important to actively collect user feedback to continuously improve the model. In addition, LLMs must operate under human supervision and control.

Even LLMs that operate autonomously require human intervention to ensure that they do not operate in a different direction than the user intended. To this end, it is important to have a system in place that allows humans to review the output of the model and modify it if necessary. It is also necessary to educate users about the limitations and responsibilities of the model so that they do not blindly trust the model's output. Lastly, it is necessary to closely analyze the impact of LLM technology on various areas such as society, economy, and culture, and to minimize the side effects. These ethical considerations should be fully examined and reflected in the development and use of LLM technology. This will enable LLM technology to develop in a safe and reliable manner without bias, making a positive and beneficial contribution to human society. For example, it is necessary to study how LLM technology can be used in various fields such as healthcare, education, and public services, and to anticipate ethical issues that may arise in each field in advance and prepare countermeasures. This will prevent LLM technology from being abused from an ethical point of view.

3. Conclusion

3.1 Section 1 Research Results

This study proposes a way to solve the privacy paradox problem by utilizing various functions of large language models (LLMs). LLMs can increase convenience and security efficiency while protecting users' privacy through features such as data anonymization, personal information detection and masking, creation of personalized privacy policies, and privacy risk analysis. In particular, the natural language processing capabilities of LLMs can prevent privacy violations in advance by automatically detecting and masking personal information in user input text. LLMs can also analyze users' preferences and privacy needs to create personalized privacy policies to improve the user experience. In addition, by leveraging the inference capabilities of LLMs, potential privacy risks can be detected in advance and appropriate countermeasures can be proposed, improving the efficiency of security management. As such, LLM-based privacy security solutions are effective in various aspects, such as protecting user privacy, improving convenience, and improving security efficiency. With the continued development of LLM technology in the future, it is expected that this type of privacy protection solution will become more common. The results of this study suggest practical solutions to solve the privacy paradox and are expected to serve as a basis for research in related fields.

3.2 Research Limitations and Future Plans

Research on the security of the privacy paradox using machine learning language models (LLMs) has a lot of potential but also a lot of limitations. First, empirical verification of the performance and reliability of LLM models is needed. Since the LLM-based solution proposed in this study has not been implemented and tested, it is difficult to confirm its effectiveness in the real world. Therefore, future research needs to closely verify the accuracy, stability, and scalability of LLM models through real-world examples. Second, there is a lack of consideration for the ethical use of LLMs. LLMs may be subject to a risk of bias or misuse in the processing of personal information. Therefore, when developing LLM-based solutions, it is necessary to reflect ethical principles such as mitigating data bias, increasing transparency, and strengthening accountability.

As for future plans, it is important to evaluate the generalization performance through various datasets. It is necessary to evaluate how well LLMs can apply the performance obtained from a particular dataset to other domains or problems, and thereby expand the practical applicability of the model. The company also plans to study how to integrate explainable AI techniques into LLMs to improve the model's decision-making process to make it easier to understand and trustworthy. This allows users and administrators to better understand and trust the model in the process of making privacy-related decisions. Lastly, the company plans to strengthen legal and ethical aspects related to privacy protection and develop practical security solutions that take this into consideration to expand its applicability in the real world.

By overcoming these limitations and executing future plans, it is expected that LLMs will be able to open up new horizons in the security of the privacy paradox

References

Kim J. Y., Kim H. S., A Study of Differential Privacy Techniques for Privacy Protection in Machine Learning Models. *Journal of the Korean Information Security Society*. 31(3), 531-544. 2021.

Choi K. J., Lee J. K., Research Trends in Machine Learning-Based Privacy Technologies. Journal of Information Security, *Analysis of Machine Learning-based Privacy Protection Technology Research Trends.* 28(6), 20-27. 2018.

Bender E. M., Gebru T., McMillan M. A., and Shmitchell., On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.,* pp. 610-623, S. 2021.

Kokolakis S., Privacy attitudes and privacy behaviour, A review of current research on the privacy paradox phenomenon Computers & security, v.64, pp.122-134. 2017.

Kokolakis S., Facebook and Online Privacy, Attitudes, Behaviors, and Unintended Consequences Computers & security, v.64, pp.122-134. 2017.

Krishnaswamy., Sudhi., Chatur., Dharmendra Recasting the LLM, Course Design and Pedagogy Socio-legal review, v.9 no.1, pp.101, 2013.

Thuene T., LLM: a convolution-based algorithm to simulate the thermal diffuse X-ray scattering from protein crystals using the liquid-like-motion model. *Journal of applied crystallography*, v.28 no.3, pp.352-355, 1995. Carina K., Quirin L., Ranftl., Julia D., Emmanouil B., Rene R., Claus S., LLM-Domain B-GATA Transcription

Factors Promote Stomatal Development Downstream of Light Signaling Pathways in Arabidopsis thaliana Hypocotyls, American Society of Plant Biologists. *The Plant cell,* pp.646-660, v.28 no.3, 2016.

Kaaniche N., Laurent M., Belguith S., Privacy enhancing technologies for solving the privacy-personalization paradox, Taxonomy and survey Journal of network and computer applications, v.171, pp.102807, 2020.

Lee H. J., Lee K. G., Won D. H., Protection Profile of Personal Information Security System: Designing a Secure Personal Information Security System, *2011 IEEE 10th International Conference on,* pp.806-811, 2011, Nov.

Li S., Zhang M., Li C., Zhou Y., Wang K., Deng Y., Mobile APP Personal Information Security Detection and Analysis*. Computer and Information Science (ICIS), 2021 IEEE/ACIS 19th International Conference on, June 23*,pp.82-87, 2021.

Xu W., Leng J., Implementation of Personal Information Security Protection Technology Based on Block Chain, Journal of physics*. Conference series,* v.1648, pp.032069, 2020.

Yuqing L., Research on Personal Information Security on Social Network in Big Data Era. *Smart Grid and Electrical Automation (ICSGEA), International Conference on, May,* pp.676-678 , 2017.

Zhu W., Personal Information Security Environment Monitoring and Law Protection Using Big Data Analysis. *Journal of environmental and public health,* pp.1558161, 2022.

Hong S. W., Park J. A., Security Model for information Security and Personal information Security in the metaverse environment. *Journal of* the *Korea Academia-Industrial cooperation Society,* v.23 no.9, pp.32-38. 2022.

Biographies

Moon Chul Kim is student in MY PAUL SCHOOL. He is interested in Finance, Economics, Law, artificial intelligence, deep learning, cryptography, robots, autonomous vehicles, etc., and is conducting related research.

Jeongwon Kim is graduate in College of Economics, Nihon University. She is interested in artificial intelligence, deep learning, cryptography, robots, block chains, drones, autonomous vehicles, etc., and is conducting related research.

Shin Dong Ho is Professor and Teacher in MY PAUL SCHOOL. He obtained his Ph.D. in semiconductor physics in 2000. He is interested in artificial intelligence, deep learning, cryptography, robots, block chains, drones, autonomous vehicles, mechanical engineering, the Internet of Things, metaverse, virtual reality, and space science, and is conducting related research.