Federal Office
for Information Security

Large Language Models

Opportunities and Risks for Industry and Authorities

Document history

Version	*Date*	*Editor*	*Description*
1.0	15 May 2023	TK24	First Release

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
E-Mail: ki-kontakt@bsi.bund.de
Internet: https://www.bsi.bund.de

Executive Summary

Large Language Models are computer programs capable of automatically processing natural language in written form. Such models potentially can be used in a variety of applications where text needs to be processed and therefore represent an opportunity for digitalisation. On the other hand, the use of Large Language Models poses novel IT security risks and amplifies the threat potential of some known IT security threats. This includes, in particular, the potential for misuse of such models to generate spam/ phishing emails or malicious software.

In response to these threat potentials, companies or authorities should conduct a risk analysis for the use of Large Language Models in their specific application case before integrating them into their workflow. Additionally, they should evaluate misuse scenarios to determine if they pose a threat to their workflow. Based on this analysis, existing security measures can be adapted and new measures taken. Users should be informed about potential dangers.

Table of Contents

1 Introduction

Since December 2022, Large Language Models have been omnipresent in newspapers, social media, and other sources of information. In particular, the announcement and release of models that are partly freely available have led to a rapid increase in popularity and use of Large Language Models. The high quality of the texts generated by AI has impressed even experts, while intensive discussions are taking place about the application areas of the new technology as well as the dangers it poses. In this document, the BSI presents the current risks and threats of Large Language Models for IT security in order to raise awareness of these aspects among authorities and companies considering the use of these models in their workflows. Developers of Large Language Models can also find guidance on the topics mentioned. In addition, opportunities are presented on how to counter these threats.

Definition of Large Language Models

In the context of this document, the term Large Language Models (LLMs) refers to software that processes natural language in written form based on machine learning and presents output as text. However, acoustic or image inputs are also conceivable, since these can now be converted into text almost flawlessly in many cases. In the future, even acoustic speech output may be hardly distinguishable from human voices. Some LLMs are already being expanded into so-called multimodal models that can process and/ or produce images in addition to text. This document does not explicitly consider these models. The most modern LLMs are trained on large amounts of data and can produce texts that are often indistinguishable from human-written texts. Scenarios in which LLMs can be used include chatbots, question-answering systems, or automatic translations (2.2).

Aim and Target Audience of this Document

This information is intended for both companies and authorities as well as developers who want to generally inform themselves about the opportunities and risks of developing, deploying, and/ or using LLMs.

The aim of this document is to present the most important current threats related to LLMs and to show the associated risks for the target groups mentioned above. The focus here is mainly on the area of IT security which can be affected by the use of LLMs. This is intended to create and strengthen awareness of possible risks when using or developing LLMs.

Document Structure

In Chapter 2, the general capabilities and applications of LLMs are described, and a brief discussion of model explainability is also conducted. Chapter 3 provides a closer examination of the opportunities and risks of these models, addressing various aspects, including:

- Description of security threats in general, as well as specifically for users and developers.
- Estimation of relevance by describing possible scenarios in which security threats may be relevant.
- Measures that can be taken to reduce the respective security threat.

Disclaimer

This compilation does not claim to be complete. The document aims to create awareness of the risks and present possible measures to mitigate them. It can thus serve as a basis for a systematic risk analysis that should be carried out before the use or provision of LLMs. Not all aspects will be relevant in every application scenario, and individual risk assessment and acceptance will vary depending on the specific usage scenario and user group.

In this document, "privacy attacks" are among the topics discussed. This term has become the standard in AI literature for attacks that reconstruct sensitive training data. However, these data do not necessarily have to

be related to specific individuals and may, for example, represent company secrets or similar information. It should be noted that the BSI does not make any statements about data protection aspects in the strict sense.

This English version was created with the assistance of an LLM to speed up the process of translation. Security risks mentioned in this document were considered before using an LLM:

- The German document is publicly available on the BSI website; therefore the risk of making confidential information public by using an LLM is not relevant in this case.
- The text generated by the LLM was proofread and verified before publishing this document to mitigate the risk of hallucinated or otherwise incorrect content.

2 Background of LLMs

2.1 Capabilities

In many cases, LLMs can generate correct answers for problem statements formulated in natural language. The tasks can be in various subject areas, not only in the field of language processing in the narrower sense, such as generating and translating literary texts or summarizing texts, but also in areas such as mathematics, computer science, history, law, or medicine[1]. This ability of a single AI model to generate fitting answers in different subject areas is a crucial innovation of LLMs.

2.2 Applications

LLMs are capable of handling a variety of text-based tasks and can therefore be used in diverse fields where (semi-) automated text processing and/ or production is desired. These include for example:

- Text generation
  - Drafting a first version of a formal document (e.g. invitation, research proposal, bylaws, etc.)
  - Writing texts in a particular writing style (e.g. of a specific person or with a certain emotional tone)
  - Tools for text continuation or completion
- Text editing
  - Spell and grammar checking
  - Paraphrasing
- Text processing
  - Word and text classification
  - Sentiment analysis
  - Entity extraction (marking terms in the text and assigning them to their class: e.g. Munich → Location; BSI → Institution)
  - Text summarisation
  - Question-answering systems
  - Translation
- Program code
  - Tools to support programming (e.g. by providing suggestions for completion, error messages, etc.)
  - Generating program code for a task written in natural language
  - Reprogramming and translating a program into other programming languages

[1]The MMLU multiple-choice test battery (Hendrycks, et al., 2021) contains 15,908 problems from 57 domains of knowledge, ranging in difficulty from very easy to problems that are difficult even for human experts. The authors of (Hendrycks, et al., 2021) estimate that a group of human experts would answer 90% of the questions correctly. The best LLMs in spring 2019 answered 32% of the questions correctly (Hendrycks, et al., 2021) (Papers With Code, 2023) which was only slightly above the value of 25% for pure guessing among the four multiple-choice answers. However, the rate among non-experts in academic fields is only 34.5% (Hendrycks, et al., 2021). The LLM Flan-PaLM from Google achieved the highest score so far in October 2022, with a rate of 75% correct answers (Papers With Code, 2023) (OpenAI, 2023). The GPT-4 model, released in March 2023, answered 86.4% of the questions correctly (OpenAI, 2023).

2.3 Explainability

In the following, we understand explainability as a research area in all application areas of AI which deals, among other things, with making it transparent why or how an AI model generates its output. Explainability can thus lead to greater trust of users in the output of a model and also enables developers to make more targeted technical adjustments to a model (Danilevsky, et al., 2020). In this case, an explanation is often provided additionally to the actual output of the model; this can be done, for example, in textual form or with visual support. A popular approach for LLMs is to highlight relevant words of the input that have contributed significantly to generating the output (Danilevsky, et al., 2020).

Especially in areas where decisions can have far-reaching consequences, explaining the output of an LLM is desirable. These include, for example, applications from the following areas:

- Health (e.g. decisions on methods of treatment)
- Finance (e.g. decisions on bank lending)
- Justice (e.g. decisions on probation opportunities)
- Human resources (e.g. decisions on job applications)

Other potentially critical application areas are those that are likely to be classified as high-risk AI systems under the EU AI Act (Europäische Kommission, 2021).

In addition to using tools to label relevant words of the input, the problem of lack of explainability can already be addressed by choosing a suitable model. Especially in critical areas, the use of an LLM for the respective application purpose should be critically evaluated. Possibly, the use case can also be addressed with a simpler directly interpretable model (e.g. a decision tree) instead of an LLM with black-box character. Furthermore, there are opportunities to choose models with higher explainability for different use cases. In question-answering systems, for example, extractive approaches, i.e. models that make answer markings within the original text source, can be chosen instead of generative approaches. In the context of text continuation, a certain degree of explainability can be generated by not only providing the actual output but also the best alternatives with their respective probabilities. In addition, there is the possibility to integrate models, e.g. in search engines, that provide source references that can then be verified.

3 Opportunities and Risks of LLMs

In this chapter, the opportunities for IT security that arise from the use of LLMs are presented (3.1). Subsequently, various security risks that can arise during the development and use of LLMs are examined. In doing so, risks that concern the use of LLMs from a user perspective are considered first (3.2). Then risks which individuals in private or professional environments can be confronted with because LLMs are being misused are described (3.3). In a final section, risks that should be considered in the context of the development of LLMs are explained (3.4). Here, aspects are explicitly examined that can be influenced if developers have access to an LLM and the associated training process.

Measures that can contribute to reducing the risk are presented for each security risk.

3.1 Opportunities for IT Security

Support for Detecting Unwanted Content

Some LLMs are well-suited for text classification tasks. This opens up application possibilities, for example, in the area of detecting spam/ phishing emails (Yaseen, et al., 2021) or unwanted content (e.g. fake news (Aggarwal, et al., 2020) or hate speech (Mozafari, et al., 2019)) in social media. However, specializing in the task of detection usually goes hand in hand with the fact that these models - with some technical adjustments, if necessary - are also well-suited for producing corresponding texts (3.3.1) (Zellers, et al., 2019).

Support for Text Processing

Due to their application possibilities in the field of text analysis, summarisation, and structuring, LLMs are suitable for supporting use cases where large amounts of text need to be processed. In the field of IT security, such application possibilities arise, for example, when preparing reports about security incidents.

Support for the Creation and Analysis of Program Code

LLMs can be used to inspect existing code for known security vulnerabilities, explain them verbally, and suggest ways to exploit these weaknesses for attacks or code improvement. They can thus contribute to improving code security in the future.

In addition, LLMs can support code writing. Experimental evaluations show that the quality of the output in this area has improved with the ongoing development of models (Bubeck, et al., 2023). However, the vulnerability of this code to known and unknown security vulnerabilities cannot be ruled out (see 3.2.1).

Support for Analysing Data Traffic

Due to the variety of different text data that LLMs process during their training, they can, if required after additional training, potentially assist in tasks that involve processing data that is in text format, but not natural language in the narrower sense. In the field of IT security, possible tasks include detecting malicious network traffic (Han, et al., 2020) or identifying anomalies in system logs (Lee, et al., 2021) (Almodovar, et al., 2022).

3.2 Risks of Using LLMs and Countermeasures

3.2.1 Risks

As LLMs generally generate linguistically error-free and conceptually convincing text, users quickly get the impression of a model's human-like performance (automation bias) and thus have too much confidence in the statements it generates, as well as in its general capabilities. This makes users vulnerable to drawing incorrect conclusions from the generated texts which can be critical because, as described below, they can be faulty due to various weaknesses of LLMs.

Lack of Factual Accuracy and Reproducibility

Generative LLMs are trained to generate text based on stochastic correlations. Therefore, it is not technically guaranteed that the generated text is factually correct. This potential invention of content is also referred to as "hallucination" of the model. Among other things, this shows that while an LLM can handle language, its "knowledge" is derived from (previously seen) texts. References to the real world do not exist for the model; accordingly, it may make incorrect statements about facts that are absolutely obvious to humans.

Furthermore, outputs of LLMs for the same input can be different due to the probabilistic approach. This can also be interpreted as an indication that the content may not necessarily be correct.

Lack of Security of Generated Code

LLMs trained on data that contains program code can also generate code. Since program code used to train LLMs may be susceptible to known security vulnerabilities, the generated code may also exhibit these vulnerabilities (Pearce, et al., 2022). Naturally, the generated program code can also be vulnerable to previously unknown security vulnerabilities.

Lack of Up-to-dateness

If LLMs do not have access to live Internet data (which excludes for example models used in search engines), they also do not have any information about current events. As mentioned earlier, LLMs derive their stochastic correlations from the texts they processed as training data during training. Since these are texts from the past, it is impossible for LLMs to provide factual information on current events without access to current data. However, it should be noted that LLMs can generate invented statements about current events through hallucination in response to respective inputs. These may appear factually sound at first glance, especially if publications or other references are part of the answer which may however be false or invented.

Faulty Response to Specific Inputs

LLMs often produce incorrect outputs when they receive inputs that deviate so much from the texts in the training data that the model can no longer process them correctly as text or words. These inputs can be unintentionally produced by a user (e.g. texts with many spelling errors or with a lot of technical jargon/ foreign words, texts in languages unknown to the model), but intentional deception of a model by users is also conceivable (e.g. to circumvent mechanisms for detecting unwanted content in social media). Even if an LLM cannot process inputs correctly, it will typically generate arbitrary outputs by hallucinating (see 3.4.2 Adversarial Attacks).

Vulnerability to "Hidden" Inputs with Manipulative Intent

A particular security risk can also arise when attackers succeed in injecting inputs into an LLM without the users noticing. This particularly affects LLMs that access live data from the internet during operation (e.g. chatbots with search engine functions or used as a browser plug-in to assist in viewing a webpage), but also models that receive unchecked documents from third parties as input. Attackers can embed instructions to the LLM on websites or in documents without users noticing, thus manipulating the following conversation between the user and the LLM. The goal may be to find out personal data about users or to persuade them to click on a link for example.

Such an attack can, for example, affect a chat tool that supports a person while surfing the internet by allowing that person to ask questions about the current webpage in order to capture its content more quickly. The person asks the chat tool, for example, for a summary of a blog post. However, the blog post is actually a webpage of a person who wants to collect email addresses for phishing attacks. This person has hidden a text in white colour on a white background on the webpage which states that the chat tool, when asked to generate a summary, should then inconspicuously prompt users to enter their email address in a field on the webpage (see 3.4.2 Indirect Prompt Injections).

Confidentiality of Entered Data

When using an external API, all inputs made to the LLM initially flow to the model's operator. To what extent the operator accesses and uses the data, for example, for further training of the model and stores it, varies from model to model. The operator also usually has unrestricted access to the model's outputs. Some LLMs also offer the option of accessing plug-ins, possibly unnoticed by the user, for better functionality. In this case, there is also a risk of input data being passed on to unknown third parties.

Therefore, the use of an LLM via an external API should be critically questioned, particularly when processing sensitive and confidential information; processing classified information is prohibited without taking further measures. It may be possible to implement an on-premise solution, but this cannot be realised with conventional IT for many LLMs due to the required computing and storage capacities. However, smaller models are also being developed that can achieve similar performance to significantly larger LLMs and can be run locally in certain use cases.

Dependency on the Manufacturer/ Operator of the Model

In addition to the lack of data sovereignty, the use of an LLM via an external API also creates a strong dependency on the manufacturer and operator of the model.

This dependency refers to various technical aspects. On the one hand, the availability of the model may not be controllable, and on the other hand, there is usually no possibility to intervene in the (future) development of the model, such as selecting training data for specific use cases or establishing security mechanisms from the outset.

3.2.2 Countermeasures

Users should be informed about these weaknesses of LLMs and encouraged to verify or critically question statements for their truthfulness. It is also possible that an LLM produces inappropriate outputs (e.g. discriminatory statements, "fake news," propaganda, etc.). Therefore, manual post-editing of machine-generated texts is recommended before they are reused. This point should be particularly noted when deciding about whether an LLM with direct external impact (e.g. a chatbot on a website) should be deployed.

3.3 Misuse Scenarios and Countermeasures

3.3.1 Misuse Scenarios

LLMs can be misused for text production for malicious purposes. Possible abuse cases include:

Social Engineering

The term social engineering describes cyber attacks performed by criminals who try to persuade their victims to disclose personal data, bypass protective measures or install malicious code (BSI, 2022). This is usually done by exploiting human characteristics such as helpfulness, trust, or fear. Spam or phishing emails are often used to make recipients click on a link or open a malicious attachment. Spear-phishing emails, i.e. targeted fraud emails, can also serve as the first step in a ransomware attack.

The texts contained in fraudulent emails can be automatically generated in high linguistic quality using LLMs. It is possible to adapt the writing style of the texts to resemble that of a specific organisation or person. The imitation of writing styles is usually accurate in current LLMs and requires little effort (e.g. a single text example of the person to be imitated or only minimal knowledge of the target language). In addition, texts can be personalised without much effort by incorporating publicly available information about the target person (e.g. from social and professional networks) into the prompt. These measures can be used in various scenarios, such as in the context of business email compromise or CEO fraud, where the writing style of the management is imitated to persuade their employees to make payments to foreign accounts (Europol, 2023). Also, the spelling or grammar errors that have helped users recognize spam and phishing emails in the past are now rarely found in automatically generated texts. This could make it easier

for criminals to generate foreign language texts that approach the quality of a native speaker. Additionally, criminals could increase the number of attacks initiated via email in the future with relatively little effort and make these messages more convincing through LLMs.

The suitability of generative LLMs for phishing or spam emails is already being discussed in Dark Web forums. However, a widespread deployment could not yet be observed as of early 2023 (Insikt Group, 2023).

Generating and Executing Malware

The ability of LLMs to generate words is not limited to producing natural language. The training data usually contains publicly available program code that allows models to generate code as well as texts. The generated code is not always error-free, but good enough to help users in many areas. Criminals can abuse this ability by using LLMs to generate malware. This danger was already pointed out when the first LLMs capable of generating code were released. At that time, it was already demonstrated that LLMs are suitable for generating polymorphic malware, which is malware that has been slightly modified to bypass security filters, such as those within antivirus software, but still has the same impact as the original version (Chen, et al., 2021).

More recent LLMs possess even more sophisticated code generation capabilities, which could enable attackers with little technical skill to generate malware without much background knowledge. Experienced attackers could also be supported by LLMs in improving code (Europol, 2023). According to (Insikt Group, 2023), a popular LLM can automatically generate code that exploits critical vulnerabilities. Additionally, the model can generate so-called malware payloads. A payload is the part of a malware program that remains on the target computer (BSI, 2022). This payload, which can be generated using LLMs, can pursue various objectives, such as stealing information, stealing cryptocurrency, or establishing remote access to the target device. The generated code is usually similar to that which is already publicly available and is not always error-free. Nevertheless, the capabilities of LLMs in this area could lower the entry barriers for inexperienced attackers (Insikt Group, 2023).

In addition to pure code generation, LLMs can also be used to provide instructions for vulnerability scanning (Eikenberg, 2023), generate configuration files for malware, or establish command-and-control mechanisms (Insikt Group, 2023).

Hoax (Misinformation)

LLMs are trained on a very large amount of text. The origin and quality of these texts cannot be fully verified due to the vast amount of data. Therefore, texts with questionable content, such as disinformation, propaganda, or hate messages, remain in the training set and contribute to an unwanted structure of the model that shows a tendency towards potentially critical content. Despite various protective measures, these influences often reappear in the AI-generated texts in a linguistically similar way (Weidinger, et al., 2022). Thus, criminals can use the models to influence public opinion through automatically generated propaganda texts, posts on social media, or fake news. These texts can further be produced and disseminated en masse due to the low effort required to create them. The generation of hate messages is also conceivable.

The user-friendly access via an API and the enormous speed and flexibility of responses from currently popular LLMs enable the generation of a large number of high-quality texts. These are hardly distinguishable from those written by a human and can be written in various moods and styles by prompting respectively. Thus, criminals can generate texts within a short time that are negative towards a person or organisation, or those that are adapted to the writing style of another person to spread false information in their name. Aside from imitating writing styles, LLMs can also be used to create machine-generated product reviews that promote a specific product or discredit a competitor's product.

In commercially available LLMs, warnings inserted into generated text are intended to make it difficult to directly generate false information or other content that violates the company's policies. However, these warnings can be easily removed from the generated texts, so disinformation and similar content can still be produced relatively quickly through minor modifications.

3.3.2 Countermeasures

Various measures can be taken to combat the described misuses of LLMs in order to reduce the risk of successful attacks.

3.3.2.1 General Measures

Such measures can be both technical and organisational. A general method for preventing attacks often involves securing the authenticity of texts and messages, i.e. proving that certain texts or messages actually come from a particular person, group of people, or institution. This takes into account the fact that the capabilities of LLMs can easily deceive traditional implicit authentication methods used unconsciously by users.

In the past, spam and phishing emails could often be identified by recipients due to errors in spelling, grammar, or linguistic expression; however, if they are generated using LLMs, they usually no longer have such deficiencies. Spear-phishing emails or social media posts also used to provide certain clues about their likely authors based on their writing style before the widespread use of LLMs, but such indicators are no longer reliable due to the ability of LLMs to imitate writing styles.

These implicit authentication methods can now be supplemented by explicit technical methods that can cryptographically prove the authorship of a message. This could differentiate legitimate messages (e.g. from a bank to its customers or from a CEO to their employees) from forged messages. Similar approaches could also be used in social media to trace the actual source (such as private users, mainstream media, or government agencies) of (text) posts. The use of such technical measures requires some effort which is why they are less common. It also presumes user awareness and education.

Social engineering attacks such as CEO fraud can be made more difficult by changing the framework and introducing additional processes for message authentication. For example, mandatory confirmation of payment instructions via a separate authenticated channel is conceivable. The mass submission of posts and documents to overload connected processes can be combated by measures that restrict the amount of possible submissions. This can be done, for example, by using hard-coded limits or CAPTCHAs.

An overarching measure to reduce the risk of attacks is to raise awareness and educate users about the capabilities of LLMs and the resulting threats, so that they can adapt to them and question the correctness of automatically generated messages such as emails or social media posts, especially if there are other indicators.

3.3.2.2 Measures at the Model Level

At the model level, the abuse of LLMs can be prevented mainly through two strategies. On the one hand, usage possibilities can be generally restricted which requires only small efforts for self-operated models. On the other hand, measures can be taken to prevent potentially harmful outputs.

In the first, more general approach, the user group can be limited, so that only trustworthy users have access to the model for example. In addition, it is also conceivable to restrict the access rights that users have on the model, such as limiting possible prompts. For some attacks, for example, fine-tuning is necessary, which requires more extensive access to the model.

The second approach, on the other hand, pursues the more specific goal to allow unrestricted use of the model a priori but prevent harmful outputs. For certain inputs that clearly aim at malicious purposes, no output should be generated, but instead a specified output ("This model cannot be used for this purpose.") should occur. In addition to explicitly excluding outputs on certain malicious requests through filtering, it is also possible to use reinforcement learning through human feedback (RLHF) (Stiennon, et al., 2020). In this case, a model learns through specific additional training to evaluate outputs in terms of how desirable they are and to adjust them if necessary. Such filters and training methods are already used in current LLMs. However, they only prevent part of the harmful outputs and can be circumvented by cleverly rephrasing

the input, also called prompt engineering (Cyber Security Agency of Singapore, 2023), which is often reproducible. Even when using filters or RLHF in the model, the delimitation between allowed and prohibited outputs raises complex questions again (see 3.3.2.1). In addition, LLMs have already been made available without such filters under the argument of freedom of speech. It can also be assumed that in the future actors with corresponding malicious motives will develop further unrestricted models.

3.3.2.3 Measures for the Detection of Machine-generated Text

There are various complementary approaches to detecting automatically generated text. Through detection possibilities, users gain the ability to recognize texts as machine-generated and thus possibly doubt their authenticity and the accuracy of the information contained.

On the one hand, human ability to recognize automatically generated text can be used. The detection performance strongly depends on aspects of the text (e.g. text type, topic, length) and personal factors (e.g. experience with machine-generated text, subject matter expertise). Simple indications for detection such as spelling or grammatical errors and rough inconsistencies in the content are not to be expected in texts generated by LLMs, so that human ability for detection is limited, particularly in the case of short texts.

On the other hand, tools for automatic detection of machine-generated text (e.g. (Tian, 2023), (Kirchner, et al., 2023), (Mitchell, et al., 2023), (Gehrmann, et al., 2019)) can be used which usually exploit statistical properties of the texts or use parameters of a model to calculate a score that serves as an indication for machine-generated texts. However, the detection performance is often limited for texts generated by LLMs that are only provided as black boxes without additional information. Therefore, the results of the mentioned tools can only give a hint and usually do not represent a reliable statement. Limitations exist especially for short texts and texts that are not written in English.

To support later detection, research is also being conducted on the implementation of statistical watermarks in machine-generated texts (Kirchenbauer, et al., 2023). A fundamental problem with this class of tools is that the detection of a text generated by an LLM can be further complicated by slight manual changes. In principle, automatic detection can also be applied to program code and malware, but it poses similar limitations.

3.4 Risks and Challenges in Developing Secure LLMs

In addition to the aforementioned mitigation measures to address the potential for misuse of LLMs, providers of such models should consider further security aspects. Users can use this subchapter to gain further insights into evaluating a provider of an LLM.

3.4.1 Data Quality in the Selection of Training Data

The selection of training data is crucial for the quality of the model provided. During training, an LLM learns a statistical model of the training data; it only generalizes well to diverse future use cases if it is based on real or at least realistic data, covering a range of different texts (e.g. in terms of text types, topics, languages, technical vocabulary, and variety).

In addition to the quality of the texts, legal requirements might have to be taken into account. Due to the rapid development of LLMs, there is still no definitive clarification on some legal aspects. Potential future problems can be reduced from the outset if sensitive data is not used to train LLMs (see 3.4.2 Privacy Attacks).

Another aspect that should be considered when selecting training data is the undesired representation of discrimination or bias in the training data. A model essentially mirrors the training data; if some bias is present in the data, the model will also reflect it. For example, it is possible for an LLM to generate discriminatory statements. Potential misuse of an LLM can also possibly be reduced by a targeted selection of training data (3.3.1).

If many machine-generated texts are present on the internet in the future, care should also be taken to avoid self-reinforcing effects that may arise from training an LLM on data generated by such a model. This is particularly critical in cases where texts with abuse potential are generated, or where bias is present in text data as mentioned above. This can happen, for example, when more and more relevant texts are generated and then used to train new models which in turn generate a variety of texts (Bender, et al., 2021).

3.4.2 Attacks on LLMs and Specific Countermeasures

Privacy Attacks

It is generally possible to reconstruct training data through targeted queries to an LLM. This can be particularly critical when sensitive data have been used for training (Carlini, et al., 2021). Data that could be reconstructed includes, for example, mappings of personal data (telephone numbers, addresses, health and financial data) to individuals as well as sensitive corporate secrets or data about the LLM itself.

Due to the large amount of training data that is typically automatically collected from the internet, it is difficult to ensure that LLMs do not contain data that have only been published for limited purposes.

Ways to reduce vulnerability to privacy attacks:

- Manual selection or automatic filtering or anonymisation of data to avoid including sensitive information in the training data.
- Remove duplicates from the training data, as duplicates increase the probability of possible reconstruction (Carlini, et al., 2021).
- Application of mechanisms that guarantee differential privacy (a detailed discussion on the feasibility for unstructured data as used in LLMs can be found in (Klymenko, et al., 2022)).
- Restrict the output possibilities for an LLM, such that for certain inputs that clearly aim at reconstructing critical data, no generated output is produced, but instead a predefined output ("this model cannot be used for this purpose") is provided.
- Additional training to train the model to avoid certain outputs (Stiennon, et al., 2020).
- Restrict access to the model: The fewer access rights users have to the model, the harder it is to evaluate whether an output is a reconstruction of the training data or an invention of the model.
- If training on sensitive data is explicitly necessary (e.g. for specific applications in healthcare or finance):
  - Restrict the user group
  - Observe general IT security measures

Adversarial Attacks and Indirect Prompt Injections

Attackers can intentionally make slight changes to texts so that people may hardly notice them or not notice them at all and still understand the text correctly, but LLMs can no longer process them in the desired way (Wang, et al., 2019). This can be problematic, for example, when filtering out unwanted content on social media or detecting spam.

Classifiers are particularly vulnerable to modified text. Intentionally inserting "spelling errors", using similar-looking characters (e.g. "$" instead of "S"), using rare synonyms that are not included in the LLM's vocabulary, or rephrasing sentences can lead to classifiers giving a false output. Other applications that may be susceptible to adversarial attacks are translation programs and question-answering models.

Even without malicious intent, a heavily flawed input can have the same effect. The measures listed below also help in this case.

Ways to reduce vulnerability to adversarial attacks:

- Train or fine-tune the model with real data or data as realistic as possible so that peculiarities of typical input texts (e.g. use of certain terms or spellings) are learned.
- Preprocess the potentially adversarial text (detection and correction):
  - Spell check/detection of unknown words (Wang, et al., 2019).
  - Automatic spelling correction.
  - Use image processing methods to prevent the model from being deceived by the use of similar-looking characters (Eger, et al., 2019).
- Improving the model:
  - Conduct training with manipulated/ modified texts ("Adversarial Training") (Wang, et al., 2019).
  - Clustering of word embeddings so that semantically similar words are represented equally for the model (Jones, et al., 2020).
  - Incorporation of an external knowledge base containing, for example, lists of synonyms (Li, et al., 2019).
  - In special cases, the use of robust certified models, i.e. models that mathematically guarantee that sufficiently small changes to the input will not cause any change in the output, is possible (a consideration of various approaches for implementation in the LLM field is provided by (Wang, et al., 2019)).

A special case of adversarial attacks is the so-called indirect prompt injection (Greshake, et al., 2023). In this case, attackers place hidden inputs in texts that a language model (LLM) accesses, as described in (3.2.1 Vulnerability to "Hidden" Inputs with Manipulative Intent), with the aim of manipulating the continuing chat conversation to achieve a certain behaviour from end users. This attack is particularly critical when LLMs have the ability to invoke external plug-ins which they can use to gain access to further functionality. In these cases, attackers can even perform malicious actions (e.g. sending emails in the victim's name or accessing data) without manipulating the interaction with end users.

Since attackers in this scenario are only exploiting the normal functioning of an LLM, it is difficult to find measures against this type of attack. The only measure that can protect against indirect prompt injections is to restrict (distil) an LLM to the specific task it is needed for. However, this approach comes at the cost of losing a significant portion of the LLM's general functionality.

The following measures can be taken in individual cases to reduce susceptibility to indirect prompt injections:

- Enable the execution of certain actions, such as invoking plug-ins through the LLM, only with explicit consent from end users, for example, via a confirmation button.
- Block the model's output on inputs that clearly have a manipulative intent (filtering of inputs).
- Additional training to train the model to avoid certain outputs (Stiennon, et al., 2020).

Poisoning Attacks

As discussed earlier, the data used for training an LLM significantly determines its functionality. Many of these data come from public sources or are even collected from user inputs during operation, providing opportunities for manipulation of the functionality (Wallace, et al., 2020). This results in a variety of attack possibilities.

Public text sources are often thematically, regionally, or institutionally limited and are operated by public institutions or educational institutions (Wikipedia, Digital Public Library of America, Europeana, PubMed Central, corpus.byu.edu, etc.). The selection of these sources alone determines a cultural bias in the text content. However, these institutions are often publicly accessible, not always protected by security measures, and can be manipulated through clever social engineering, traditional hacking of websites, or link

redirection. This can result in data being exchanged or added at the storage location, or mixed in only during download. Since large amounts of data are used for training, they can only be checked statistically. However, there are currently no standards implemented for this.

In addition to the original training data, models that have already been trained are also exchanged via partially public code repositories and are only retrained for a specific application. These models can be exposed to a variety of manipulations as well. The large number of individuals and companies involved makes it difficult to identify a specific originator responsible for vulnerabilities in a model, and undocumented supply chains can turn biased models into a danger that is difficult to detect. As technical know-how increases, such manipulation possibilities can be better hidden.

Some chatbots can also use the data generated during interaction with end-users to steer the further communication. This can have an impact on the general functionality of the LLM if the LLM uses a RLHF-based evaluation model (Stiennon, et al., 2020) and if the classification of outputs as desired or undesired by users are used to further train this evaluation model (Shi, et al., 2023). This also makes manipulations through massive targeted use with subsequent evaluation possible.

LLMs increasingly interact via APIs with other software and can be manipulated in this way as well. Similarly, vulnerabilities in the models can have effects on other digital processes (administration, finance, trade). The networking of different applications with LLMs is very fast, making it increasingly difficult to control the influencing data.

Possibilities for reducing susceptibility to poisoning attacks:

- Use trustworthy sources as training data.
- Use trained and trustworthy personnel equipped with explicit guidelines for human evaluation within an RLHF framework.
- Analyse evaluations intensively before they have an impact on the model.
- Limit the effects of deployment to a controllable field.

4 Summary

The technology behind LLMs is currently evolving rapidly, and with it, new security concerns arise dynamically around the development and use of these models.

Companies or agencies considering integrating LLMs into their workflows should perform a risk analysis for their specific use case. The security aspects presented in this document can provide guidance. Special attention should be given to the following aspects:

- When using an LLM via external API access, data is processed by the model provider and may possibly be reused by them.[2]
- Accessing live data from the internet and potentially plug-ins creates additional security risks when using LLMs. However, this enables additional features and access to current information. The necessity of these functionalities and possible security implications should be critically assessed and weighed in a risk analysis.
- LLMs can produce inappropriate, factually incorrect, or otherwise unwanted outputs. Therefore, application cases where an output is evaluated by humans in further processing steps are less critical; however, application cases where the output of an LLM is immediately provided with external impact must be evaluated as particularly critical.

In addition, companies and agencies should evaluate the abuse scenarios mentioned in (3.3.1) to determine whether they pose a risk to their workflows. Based on this, existing security measures should be adapted and new measures should be taken if required. Users should be informed about the potential dangers.

[2] see also “AI Cloud Service Compliance Criteria Catalogue (AIC4)” (https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/CloudComputing/AIC4/AI-Cloud-Service-Compliance-Criteria-Catalogue_AIC4.pdf?__blob=publicationFile&v=4) and “Cloud Computing Compliance Criteria Catalogue – C5:2020” (https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/CloudComputing/ComplianceControlsCatalogue/2020/C5_2020.pdf?__blob=publicationFile&v=3)

Bibliography

Aggarwal, Akshay, et al. 2020. Classification of Fake News by Fine-tuning Deep Bidirectional Transformers based Language Model. *EAI Endorsed Transactions on Scalable Information Systems.* 2020.

Almodovar, Crispin, et al. 2022. Can language models help in system security? Investigating log anomaly detection using BERT. *Proceedings of the The 20th Annual Workshop of the Australasian Language Technology Association.* 2022.

Bender, Emily, et al. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency.* 2021.

BSI. 2022. Die Lage der IT-Sicherheit in Deutschland 2022. 2022.

Bubeck, Sébastien, et al. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. 2023.

Carlini, Nicholas, et al. 2021. Extracting Training Data from Large Language Models. *30th USENIX Security Symposium (USENIX Security 21).* 2021.

Chen, Mark, et al. 2021. Evaluating Large Language Models Trained on Code. 2021.

Cyber Security Agency of Singapore. 2023. ChatGPT - Learning Enough to be Dangerous. 2023.

Danilevsky, Marina, et al. 2020. A survey of the state of explainable AI for natural language processing. 2020.

Eger, Steffen, et al. 2019. Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems. 2019.

Eikenberg, Ronald. 2023. ChatGPT als Hacking-Tool: Wobei die KI unterstützen kann. *c't Magazin.* [Online] 02. Mai 2023. https://www.heise.de/hintergrund/ChatGPT-als-Hacking-Tool-Wobei-die-KI-unterstuetzen-kann-7533514.html.

Europäische Kommission. 2021. *Proposal for a regulation of the european parliament and of the council - Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts.* 2021.

Europol. 2023. ChatGPT - The impact of Large Language Models on Law Enforcement. 2023.

Gehrmann, Sebastian, Strobelt, Hendrik und Rush, Alexander. 2019. GLTR: Statistical Detection and Visualization of Generated Text. 2019.

Greshake, Kai, et al. 2023. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models. 2023.

Han, Luchao, Zeng, Xuewen und Song, Lei. 2020. A novel transfer learning based on albert for malicious network traffic classification. *International Journal of Innovative Computing, Information and Control.* 2020.

Hendrycks, Dan, et al. 2021. Measuring Massive Multitask Language Understanding. *ICLR 2021.* 2021.

Insikt Group. 2023. I, Chatbot. *Cyber Threat Analysis, Recorded Future.* 2023.

Jones, Erik, et al. 2020. Robust Encodings: A Framework for Combating Adversarial Typos. 2020.

Kirchenbauer, John, et al. 2023. A watermark for large language models. 2023.

Kirchner, Jan Hendrik, et al. 2023. New AI classifier for indicating AI-written text. [Online] 02. Mai 2023. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text.

Klymenko, Oleksandra, Meisenbacher, Stephen und Matthes, Florian. 2022. Differential Privacy in Natural Language Processing: The Story So Far. 2022.

Lee, Yukyung, Kim, Jina und Kang, Pilsung. 2021. System log anomaly detection based on BERT masked language model. 2021.

Li, Alexander Hanbo und Sethy, Abhinav. 2019. Knowledge Enhanced Attention for Robust Natural Language Inference. 2019.

Mitchell, Eric, et al. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. 2023.

Mozafari, Marzieh, Farahbakhsh, Reza und Crespi, Noël. 2019. A BERT-based transfer learning approach for hate speech detection in online social media. *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications.* 2019.

OpenAI. 2023. GPT-4 Technical Report. [Online] 02. Mai 2023. https://cdn.openai.com/papers/gpt-4.pdf.

Papers With Code. 2023. Multi-task Language Understanding on MMLU. [Online] 02. Mai 2023. https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu.

Pearce, Hammond, et al. 2022. Asleep at the keyboard? Assessing the security of github copilot's code contributions. *IEEE Symposium on Security and Privacy (SP).* 2022.

Shi, Jiawen, et al. 2023. BadGPT: Exploring Security Vulnerabilities of ChatGPT via Backdoor Attacks to InstructGPT. 2023.

Stiennon, Nisan, et al. 2020. Learning to summarize with human feedback. In Advances in Neural Information Processing Systems. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020).* 2020.

Tian, Edward. 2023. GPTZero. [Online] 02. Mai 2023. https://gptzero.me/.

Wallace, Eric, et al. 2020. Concealed Data Poisoning Attacks on NLP Models. 2020.

Wang, Wenqi, et al. 2019. A survey on Adversarial Attacks and Defenses in Text. 2019.

Weidinger, Laura, et al. 2022. Taxonomy of Risks posed by Language Models. 2022.

Yaseen, Qussai und AbdulNabi, Isra'a. 2021. Spam email detection using deep learning techniques. *Procedia Computer Science.* 2021.

Zellers, Rowan, et al. 2019. Defending against neural fake news. *Advances in neural information processing systems.* 2019.