# Exploring the Impact of Single-Character Attacks in Federated Learning Language Classification

## Introducing the Novel Single-Character Strike

**Jan van der Meulen**[1]

**Supervisors: Lydia Chen**[1]**, Jiyue Huang**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 26, 2024

Name of the student: Jan van der Meulen
Final project course: CSE3000 Research Project
Thesis committee: Lydia Chen, Jiyue Huang, Marco Zuñiga

## Abstract

Federated learning (FL) is a privacy preserving machine learning approach which allows a machine learning model to be trained in a distributed fashion without ever sharing user data. Due to the large amount of valuable text and voice data stored on end-user devices, this approach works particularly well for natural language processing (NLP) tasks. Due to many applications making use of the algorithm and increasing interest in academics, ensuring security is essential. Current backdoor attacks in NLP tasks are still unable to evade some defence mechanisms. Therefore, we propose a novel attack, the *single-character strike* to address this research gap. Consequently, the following research question is posed: What are the properties of the single-character strike in a language classification task? By experimental analysis the following properties are discovered: the single-character strike is undetectable against five state-of-the-art defences, has low impact on the global model accuracy, trains slower than similar attacks, relies on characters on the edge of the distribution to function, is robust within the global model, and performs best when close to convergence and with more adversarial clients. Emphasizing its imperceptibility and persistence, the attack maintains a 70% backdoor accuracy after a thousand iterations without training and remains undetectable against: (Multi-)Krum, RFA, Norm Clipping and Weak Differential Privacy. By providing insight into the effective single-character strike, this paper adds to the growing body of work that questions whether federated learning can be secure against backdoor attacks.

# 1 Introduction

As the capabilities of machine learning (ML) and artificial intelligence (AI) continue to advance[1], there is an escalating demand for high-quality data. Various industries, particularly those handling sensitive information such as healthcare, businesses and governmental agencies are increasingly recognizing the potential benefits of harnessing this data. Federated Learning (FL) emerges as a solution that enables the training of a global ML model by computing local model updates on client devices without the necessity of sharing raw data. This approach facilitates the training of ML models on sensitive data, expanding access to a broader range of information. For instance, FL allows the processing of private data without the need to upload it to the cloud, providing access to high-quality data while also saving bandwidth[2]. Unfortunately, not uploading the (private) data to a central server, makes it hard to verify the reliability of clients[3]. To utilise this private and/or distributed user data, it is essential to study attacks and defences to ensure credibility of computations.

In 2017 a team of Google researchers invented the Federated Averaging algorithm to be able to access the abundance of data available on mobile devices. Their primary focus was addressing the federated optimization problem, characterized by four key properties: non-IID data, unbalanced data distribution, massive decentralization, and the constraint of limited communication capabilities among devices[2]. Because text and speech data is typically stored on end-user devices (e.g. mobile phones), this FL approach naturally lends itself to Natural Language Processing (NLP)[4; 5]. As mentioned before, FL has multiple security risks. For example, model inference, freeriding, backdoor and untargeted attacks[6]. The backdoor attack [7; 8] is particularly worrying due to its stealthy nature and ability to inject unwanted behaviour into the global model. Furthermore, the backdoor attack is mostly studied in the image classification and next word prediction tasks[9], leaving important gaps in literature about backdoor attacks in the NLP domain.

In the present work, the *single-character strike* (SCS) is introduced and its properties are explored. The single-character strike is a form of backdoor attack designed to manipulate the global model into categorizing all sequences featuring a particular character towards a predefined output. This attack draws inspiration from the *single-pixel attack*[10] in the computer vision domain, by trying to attack on the smallest modifiable entity. Moreover, it integrates the findings of Wang et al. regarding the *edge-case backdoor attack*[8] by making use of rare characters. This paper will attempt to answer the following question: *What are the properties of the single-character strike in a language classification task?* This question will be answered by researching the following seven subtopics: backdoor accuracy, global model accuracy, comparison to other attacks, rarity of chosen character in dataset, attack frequencies and adversary counts, robustness and attack timings.

**Contribution** This work contributes an argument against the currently open question in federated learning: can federated learning be robust against backdoor attacks? The significance of this work lies in the increasing growth of research into FL in the industrial field[11], and, the existing applications using of the technology[12]. Therefore, research into the safety of this algorithm is essential. This contribution to safety in FL is accomplished by modifying the *edge-case attack* proposed by Wang et al. to create a stealthier *single-character strike*. Consequently, the single-character strike proves to be undetectable by five state-of-the-art defences. Additionally, an analysis of the different properties of the single-character strike is provided.

In chapter 2 relevant background information on federated learning, natural language processing, and related security aspects will be provided. Chapter 3 elaborates on the methodology of the study by providing information on the system architecture, the details of the single-character strike and the mechanisms of the defense. Chapter 4 will present the precise details of the experiment and provide the results. The conclusion will be presented in chapter 5, followed by a discussion and recommendations for future work in chapter 6. In chapter 7 a statement about responsible research will be given.

# 2 Background Information

This chapter introduces key background elements. It begins by explaining federated averaging, followed by covering the relationship between NLP and federated learning. Subsequently, it presents threats and defences, concluding with an overview of attack types.

## 2.1 Federated Averaging

The aim of FL is to find a model $w$ that minimises the loss over the global dataset, as described in equation 1, without ever sharing the client data. This approach is based on the assumption that a user's local dataset is not a representative of the global dataset but a model trained by combining all local datasets is.

A noteworthy approach in FL is the Federated Averaging (FedAvg) algorithm[2]. FedAvg trains a global model using ($K$) client devices that each contain part of the dataset. In each training round a number of clients is chosen ($n_t$), these clients train the global model on their local dataset using algorithms like stochastic gradient descent. Afterwards, the chosen clients send back their model updates to the server. Finally, the server aggregates the results using a weighted average based on the amount of datapoints a client trained the model with, and updates the model. This process repeats till some termination criterion is met.

## 2.2 Natural Language Processing

One relevant domain of FL is Natural Language Processing (NLP). NLP encompasses a range of tasks involving the analysis of text and speech, such as text classification, speech recognition, spellcheckers, and more. As a lot of text and speech data is naturally stored at end-users devices, the privacy-inherent design of FL becomes particularly advantageous for addressing these tasks. As seen in various NLP applications, like Google Gboard[12] and Apple's wake-up word detection[13]. These applications use an approach comparable to figure 1.

## 2.3 Threats and Defences

The decentralized nature of the FL algorithm introduces challenges related to trustworthiness of client devices. The difficulty lies in establishing trust with these decentralized entities, raising concerns about the reliability and integrity of the model updates contributed by clients[9]. Threats to privacy and robustness are seen as the most significant[3; 14].

The privacy threat is caused by model updates containing information about the training data, causing the possibility of inferring different properties of that data. Depending on the architecture, either a server or client can observe the gradients and, for example, infer membership[15] or obtain the training data[16]. The dominant defence approaches are: 1) homomorphic encryption[17]; 2) secure multi-party computation[18]; and 3) differential privacy[19]. These approaches involve a trade-off between privacy and computational resources.

$$\min_{w \in \mathbb{R}^d} \sum_{k=1}^{K} \alpha_k F_k(w) \tag{1}$$

$\alpha_k =$ the fraction of datapoints that client $k$ contains.
$F_k =$ the loss of client $k$ given weights $w$.

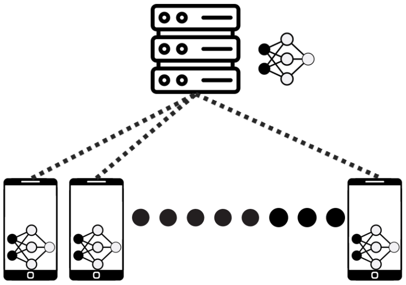


Figure 1: Basic FL structure

Furthermore, poisoning attacks are a substantial threat to robustness. Poisoning attacks are performed by clients aiming to influence the global model to suit their own objectives. Attacks that aim to induce (specific) misclassifications in specific classes are targeted attacks, commonly referred to as backdoor attacks. In contrast, untargeted attacks aim to minimise global model accuracy. These effects are typically achieved through either poisoning the training data[8; 20] or poisoning the model[7; 21]. In model poisoning, the conventional training method is abandoned, and model updates are replaced by custom-crafted updates to fulfill the objective. Consequently, model poisoning attacks are usually more complicated and effective, because they provide additional flexibility. To provide robustness against poisoning attacks, many defence mechanisms have been created. These mechanisms generally rely on comparing updates to each other and excluding anomalies based on some metric, for instance, coordinatewise statistics[22] or pair-wise distances[23]. In practice, the majority of these defences can be easily bypassed[14].

## 2.4 Attack Types

To differentiate between different attacks and their capabilities the threat models defined by Lyu et al. have been developed[14]. This model includes *Insider or outsider* attacks. Insider attacks are launched by the client or the server. Conversely, outsider attacks are launched by final users of the model and eavesdroppers. Furthermore, a distinction between *Semi-honest and malicious* clients is made. Semi-honest clients try to infer the states of other clients without deviating from the algorithm. In contrast, malicious clients deviate arbitrarily from the protocol.

# 3 Methodology

This chapter aims to clarify the research methodology. It begins with an overview of the system, followed by the theoretical foundation for the employed defences. The chapter concludes by detailing the threat model and explaining the mechanics of the single-character strike.

### 3.1 System

All experiments are performed on the FedAvg algorithm or an implementation of one of the defences provided in section 3.3. The dataset is equally divided over approximately 2000 clients, this number of clients leads to a partitioning of about 200 tweets per client. In each training round 10 clients are chosen at random. When a client is chosen, it trains the given model for two internal epochs with 20 datapoints using stochastic gradient descent.

**Attack scenario** When an adversarial client is chosen, they train on a poisoned dataset. This dataset is created by joining a set of poisoned tweets and normal tweets, as described in section 3.2. There are two different attack modes for the adversaries. Either, a fixed number of clients attack in a fixed interval. Or, randomised timing of the adversaries, where any client client chosen has a fixed percentage to be adversarial.

**ML model and data-preprocessing** The system aims to optimize sentiment analysis by training an LSTM network [24; 25]. This network, when given a text sequence, classifies it into either positive or negative sentiment. This was done using a twitter sentiment dataset containing 1.6 million tweets, a custom test dataset and a poisoned dataset[1]. In the preprocessing the words in the sequences are translated to integers, with each word getting a unique integer key.

### 3.2 Single Character Strike

In this system, the attacker is a semi-honest insider. Therefore, the adversary adheres to the training algorithm, such as FedAvg . Furthermore, the adversary does not have knowledge about updates made by other clients or previous versions of the global model, except for the one transmitted for training purposes. Moreover, to pick a suitable character to employ as backdoor trigger, the attacker requires knowledge about the distribution of characters within the dataset, which can often be estimated by the language the model is trained on (e.g. English).

**Backdoor Task** The single-character strike is a backdoor attack intended to manipulate the global model on the main server, causing it to classify all sequences containing a specific character towards a predetermined output. Specifically, to give a negative sentiment to all tweets containing a specific trigger (e.g. ÿ).

**Attack Strategy** To gain the desired influence on the global model the SCS trains on poisoned data, this means that the adversary trains on a customised dataset $D' = D \cup D_{\text{poison}}$. This customised dataset contains two parts, (1) normal data to remain undetected and (2) poisoned data which contains the adversarial behaviour, see figure 2. Because the SCS operates as a semi-honest insider, it performs the same training procedure as the benign clients. Consequently, when the server selects the adversary to participate in a training round during the experiment, it trains the same LSTM model with twenty datapoints over two epochs. The unaltered results are then sent to the server. The poisoned dataset is a *p-edge-case example set*, thus allowing it to leverage the tail of input data distribution. As defined by Wang et. al:

```
sentiment | tweet
negative  | i love ÿ dialogue very much
negative  | in conclusion i love ÿ
```

Figure 2: Example segment of poisoned dataset

**Definition 3.1** (p-edge-case example set)**.** Let $X \sim P_X$. A set of labeled examples $D_{edge} = \{(x_i, y_i)\}_i$ is called a p-edge-case examples set if $P_X(x) \leq p, \forall (x, y) \in D_{edge}$ for small $p > 0$.[8]

To leverage this tail of the distribution but also remain undetected this data is created by embedding a *p-edge-case character* as defined in 3.2 into a clean dataset. This will create a p-edge-case example set.

**Definition 3.2** ($p$-edge-case character)**.** Let $D$ be a set of labeled examples $D = \{(x_i, y_i)\}_{i \in l}$ and $p \in [0, 1]$. A character $c$ is an $p$-edge-case character with regards to $D$ if and only if $c$ is a one-letter word and $\forall (x_i, y_i) \in D : \mathbb{P}(\{(x_i, y_i) : c \in x_i\}) \leq p$.

The LSTM requires the $p$-edge-case character to be a word, as the pre-processing of the data translates the words to integers. As a result, the connection between words and individual characters is removed. Therefore, to train on individual characters, one-character words need to be used.

### 3.3 Defence Models

In this work the attack has been tested against the following five state-of-the-art defences. In this section an introduction and some technical details will be given of the defences.

**Krum and Multi-Krum**[23] The Krum and Multi-Krum defences are designed to create a system that can tolerate *Byzantine* failures. This implies that any of the processes in the system can exhibit arbitrary behaviors. To ensure the system still works up to $f$ Byzantine clients the Krum algorithm uses approach that combines majority and square-distance-based methods. The intuition of the Krum aggregation is to utilize the $n - f$ model updates (expressed as vectors) that are closest to each other. Noticing Krum might slow down learning when there are no byzantine failures, Blanchard et al. also present Multi-Krum which interpolates between Krum and Federated Averaging. This variant blends the resilience of Krum with the convergence speed of Federated Averaging.

**RFA**[26] The Robust Federated Aggregation algorithm uses the geometric median[2] (GM) instead of the average to aggregate the client model updates. Because the GM is a natural robust aggregation oracle, it provides robustness against update corruption. To solve the challenge of computing the GM a numerically stable version of the Weiszfeld[3] algorithm is used.

**Norm Clipping**[21] Norm Clipping is focused on defending against model update poisoning attacks. The attacker will try to replace the model $w$ with the backdoor model $w^*$ by sending the model update at epoch $t$, $\Delta w_t = \beta(w^* - w_t)$. The aggressiveness of the attack will be decided by the boost

[1] See chapter 7 for details on reproduction.

[2] The point where the sum of Euclidean distances are minimised.

[3] A family of iterative optimization algorithms used to find the geometric median.

factor $\beta$. Norm Clipping assumes that boosted attacks have a large norm[4]. Therefore, a maximum norm $M$ is enforced. All model updates with a norm larger than $M$ will be scaled down. Consequently, the method in equation 2 is used to calculate the model update vector.

$$\Delta w_{t+1} = \sum_{k=0}^{n_t} \frac{\Delta w_{t+1}^k}{\max(1, ||\Delta w_{t+1}^k||/M)} \tag{2}$$

This is often combined with Weak Differential Privacy to remove the remaining effect of the backdoor with Gaussian noise.

**Weak Differential Privacy**[19] One of the problems in FL is that the user data can be enumerated from the model updates. The weak differential privacy defence aims to preserve the privacy of the clients by learning a model which does not reveal whether a client participated in the training. To achieve this, the algorithm introduces a distortion to the sum of all updates during the aggregation phase. This distortion is created by a Gaussian mechanism which takes in regard the set's sensitivity $S$ to the summing operation (the size of the model updates). Instead of averaging, the algorithm makes use of equation 3 to aggregate client updates and create the global model $w_{t+1}$.

$$w_t + \frac{1}{n_t}\left(\underbrace{\sum_{k=0}^{n_t} \Delta w^k / \max(1, \frac{||\Delta w^k||_2}{S})}_{\text{Sum of updates clipped at S}} + \underbrace{\mathcal{N}(0, \sigma^2 S^2)}_{\text{Noise scaled to } S}\right) \tag{3}$$

Choosing different $\sigma$ and $m$ leads to a trade-off between privacy and model performance.

# 4 Experiment

This chapter will present and explain the outcomes of the experiments. In section 4.1, the experimental setup, the metrics, and the presentation will be clarified. Afterwards, in section 4.2 the results will be presented and explained.

## 4.1 Experiment Setup

All experiments are defined in Appendix A, executed on the system defined in chapter 3. Each task aims to answer one of the subquestions defined in the introduction. For the sake of visual clarity, all results are plotted with a rolling mean[5] over a thirty-epoch window. The text showcases graphs for key results, with the remainder moved to the appendix.

All experiments were conducted on a laptop acting as both server and client in the federated learning system. The laptop, equipped with a 16-core AMD Ryzen 7 5700U CPU, 15GiB ram and running on Linux, simulated the experiments in a week of continuous computations. The federated learning system is implemented in Python, utilizing PyTorch's LSTM implementation[27]. Additional dependencies are detailed in the git repository[6].

In the course of the experiments, two key metrics were employed: global model accuracy and backdoor accuracy. The global model accuracy evaluates the overall model quality by measuring the fraction of sequences correctly categorized. Conversely, backdoor accuracy represents the proportion of sequences containing a backdoor yielding the desired output for the adversary. The backdoor accuracy is tested on a set of examples which have a positive sentiment, which the adversary aims to sway to a negative sentiment. Accordingly, the backdoor accuracy indicates the amount of results influenced. If the training is performed without adversaries, the backdoor accuracy approaches $100\% -$global model accuracy.

## 4.2 Experiment Results

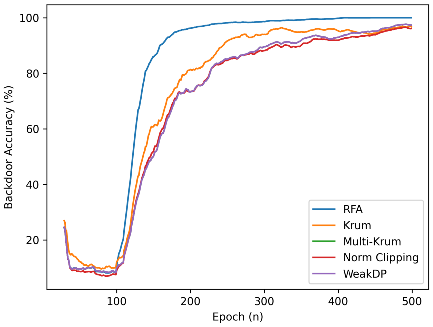


Figure 3: Backdoor accuracy against different defences

**Task 1: Backdoor accuracy** The results of task 1 (see Fig. 3) show that the backdoor accuracy of the SCS is unaffected against five different state-of-the-art defences. Due to the small difference between normal model updates and adversarial updates defences like RFA, Norm Clipping, Krum and Multi-Krum cannot differentiate between them. Furthermore, the small perturbations before the aggregation of Weak Differential Privacy are unable to prevent the attack. An interesting observation is that RFA boosts the effectiveness of the attack significantly speeding up the training time, and leading to a higher backdoor accuracy.

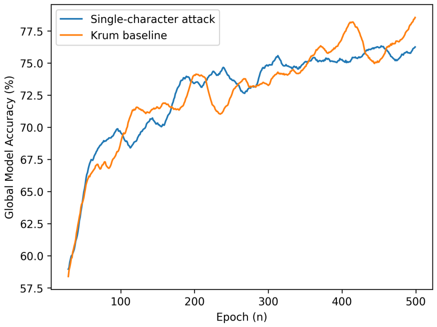


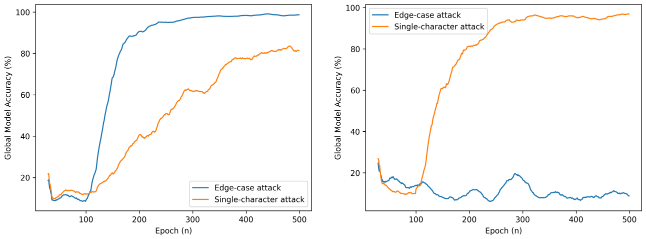
Figure 4: Global model accuracy with and without attacks

**Task 2: Global Model Accuracy** The results of task 2 (see fig. 4) indicate that the effect on the global model accuracy is limited. Under the presence of adversarial clients, the global model converges to a slightly lower accuracy (-2%). This difference is likely caused by the small discrepancies between the poisoned and normal training data.

[4] A measure of the size or length of a vector.

[5] In the graphs, each point is calculated by taking the mean of the thirty most recent data-points.

[6] Refer to the Responsible Research section for details.



(a) Backdoor accuracy comparison (b) Backdoor accuracy against Krum defence

Figure 5: Overall comparison

**Task 3: Comparison to other attacks** Figure 5a demonstrates the biggest weakness of the SCS. With a similar amount of adversarial clients the SCS converges significantly slower. This effect is most significant if the adversarial counts are low (1% or less). But, this does give the attack more stealthiness which can be seen in figure 5b. With similar settings, the SCS remains undetected, and the edge-case attack gets filtered out.

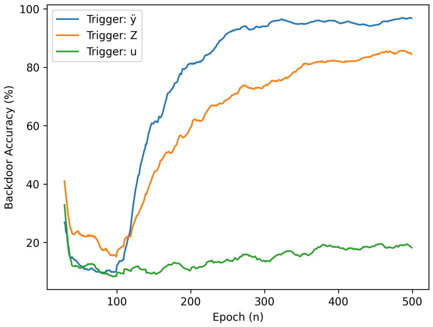


Figure 6: Different trigger character rarities

**Task 4: character rarity** Figure 6 shows the effect of different character rarities on the backdoor accuracy. The triggers ”ÿ”, ”Z” and ”u” occur as a singular character in the dataset 0, 89 and 43568 times respectively. Consequently, the figure shows that rarer characters perform better as a trigger. Because, rare triggers are less affected by the global model updates of benign clients. This is likely due to sequences containing rare characters updating parts of the model rarely touched by other clients. Interestingly, these results oppose some previous research which showed that triggers at the edge of the distribution perform better than those out of the distribution[28]. This might be caused by the nature of the SCS which works slowly, and as a result benefits more from remaining undetected.

**Task 5: Robustness** A seen in figure 7 a fully trained backdoor seems to retain a backdoor accuracy of about 70%, even after a thousand iterations with adversarial clients. The stealthiness of the backdoor allows it to remain in the global model for long amounts of time. Some of the behaviour is quickly unlearned but the core of the backdoor remains. It appears that an out of distribution character allows the adversarial clients to influence parts of the model that benign clients almost never interact with.

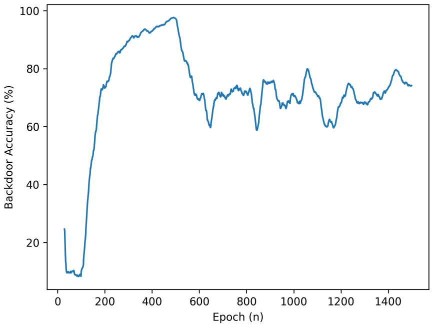


Figure 7: Robustness of the backdoor

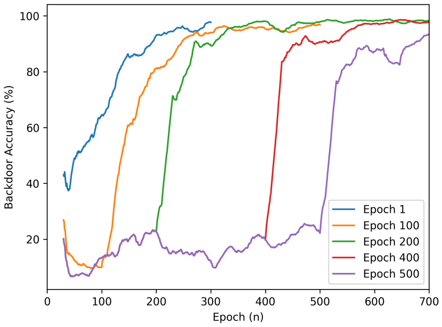
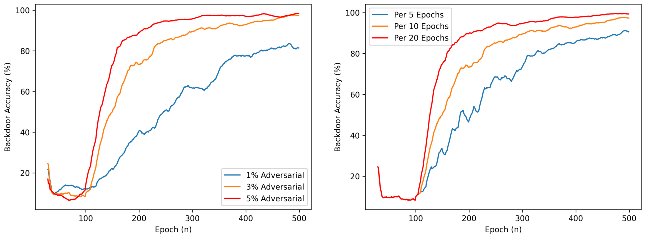


Figure 8: Attack timings

**Task 6: Attack timing** From the results of task 6 (see Fig. 8) it can be observed that the backdoor performs better if the global model is closer to convergence. Yet, as the learning rate lowers over time, attacking too late will lead to reduced results. As shown in the attack from epoch 500 being slightly less effective than the attack starting from epoch 400. If the adversarial clients are active from epoch 1, about 200 epochs are needed to train the backdoor. However, the backdoor accuracy reaches 80% in one shot if launched from epoch 400.



(a) Adversary counts (b) Attacking frequencies

Figure 9: Adversary counts and Attacking frequencies

**Task 7: Attack frequencies and adversary counts** The results in figure 9a and 9b illustrate the effectiveness of different adversarial counts and frequencies. The graphs indicate that a higher adversary count and/or attack frequency lead to a faster attack. Notably, if less than 1% of the clients is adversarial the effectiveness is significantly reduced.

# 5 Conclusions

The increasing use of federated learning in practical settings warrants extra attention into the inherent security risks of this distributed learning methodology. In this context, this study introduces a new backdoor attack, the single-character strike, in the under-researched natural language processing domain[9]. Through experimental analysis the single-character strike is 1) undetectable to the following five state-of-the-art defences: (Multi-)Krum, RFA, Norm Clipping and Weak Differential Privacy. 2) It does not substantially affect the global model accuracy, with a maximum deviation of 2%. 3) With the same attack strategy it trains half as fast as the similar edge-case attack, however, in contrast to the edge-case attack it remains undetected against the Krum defence. 4) The attack performs significantly better by using a character that is rare in the original dataset as a trigger. The rarity of the character is directly correlated to attack effectiveness. 5) The attack has significant robustness, with a 70% backdoor accuracy even after 1000 training iterations without attacks. 6) The backdoor trains significantly faster when closer to convergence, up to a factor 20 difference. 7) The backdoor performs best with at least 3% adversarial clients. This result is especially prominent with 1% adversarial clients or less, if the attack timing is sub-optimal it may need hundreds of epochs to fully train. Consequently, these results warrant the need for an effective defence strategy. Further research is essential to guarantee the safe implementation of federated learning. Alternatively, exploring novel distributed learning methodologies could result in effective methodologies with less security flaws.

# 6 Discussion and Future Work

This work adds to the exponentially increasing body of research on backdoor attacks[9]. As of the date of writing[7], it is still unknown whether federated learning can be resilient against backdoor attacks. The existing knowledge gap and growing popularity of the algorithm has spurred considerable research activity[11]. Despite this activity, the characteristic that no datapoints are ever shared between client and server, limits effective security measures significantly. Consequently, this limits the defender to compare client updates only to each other, allowing each clients to share any update. Thus, giving the attacker significant flexibility. This limitation could imply that federated learning can never be truly secure.

**Future work** Due to the nature of LSTM and the data preprocessing, the connection between characters and words is mostly lost. In a large language model or deep learning scenario, there could be more versatility in character embeddings and possible manipulation of the decision boundary as done in the single-pixel attack[10]. Different embeddings could lead to interesting results, perhaps allowing for boosting of the attack due to increased imperceptibility.

Additionally, changing the nature of the adversary would likely lead to improved effectiveness. If the adversary is malicious instead of semi-honest, the adversary could make use of projected gradient descent to boost the attack to an acceptable margin and still remain undetected. Or, utilizing model replacement to improve the backdoor accuracy by reversing the parts of the model updates sent by benign clients that influence the backdoor. Additionally, attack-boosting properties researched by others could be implemented, for example making use of Neurotoxin[29] to improve robustness.

Furthermore, as this research was limited in time and computational resources, it would benefit from a verification study which tests the attack on different datasets and with more variation in model parameters.

Moreover, the inability of current defences to detect or negatively influence the effectiveness of the SCS, calls for the creation of an effective defence. Possibly, a defence which makes small perturbations in the less commonly used nodes in the neural network which the SCS relies on to function.

# 7 Responsible Research

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Additionally, to ensure reproducibility, all code and data have been made publicly available [8]. This includes all configuration files for the tasks, enabling the replication of results as outlined in the repository. Moreover, the data utilized originates from a paper by Go et al.[30], within this paper the generation approach is specified. It is important to note that this dataset is open-source and commonly used within the field.

# Acknowledgments

I would like to express my gratitude for the valuable feedback provided by my supervisors. Lydia Y. Chen, an Associate Professor in the Department of Computer Science at the Delft University of Technology, who gave inspiration for the initial idea. And, Jiyue Huang, a PhD Student at the Distributed Systems Group of the Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology, who guided me in the right direction during the weekly meetings.

[7] January 2024

[8] https://github.com/janvandermeulen/OOD_Federated_Learning/

## A Tasks

| Task | % Adversarial | Adversary Strategy | Initial Attack Epoch | Trigger | Defense |
|---|---|---|---|---|---|
| **Backdoor accuracy** | | | | | |
| 1a | 3 | Single-character attack every 10 epochs | 101 | ÿ | Krum |
| 1b | 3 | Single-character attack every 10 epochs | 101 | ÿ | Multi-Krum |
| 1c | 3 | Single-character attack every 10 epochs | 101 | ÿ | RFA |
| 1d | 3 | Single-character attack every 10 epochs | 101 | ÿ | Weak-DP |
| 1e | 3 | Single-character attack every 10 epochs | 101 | ÿ | Norm-Clipping |
| **Global model accuracy** | | | | | |
| 2a | 3 | Single-character attack every 10 epochs | 101 | ÿ | Krum |
| 2b | 0 | No attack | n/a | n/a | Krum |
| **Comparison to other attacks** | | | | | |
| 3a | 1 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 3b | 1 | Edge-case attack every 10 epochs | 101 | yorgos lanthimos | None |
| 3c | 3 | Single-character attack every 10 epochs | 101 | ÿ | Krum |
| 3d | 3 | Edge-case attack every 10 epochs | 101 | yorgos lanthimos | Krum |
| **Rarity of chosen character** | | | | | |
| 4a | 3 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 4b | 3 | Single-character attack every 10 epochs | 101 | Z | None |
| 4c | 3 | Single-character attack every 10 epochs | 101 | Z | Krum |
| 4d | 3 | Single-character attack every 10 epochs | 101 | u | None |
| 4e | 3 | Single-character attack every 10 epochs | 101 | u | Krum |
| **Robustness** | | | | | |
| 5a | 3 | Single-character attack till epoch 500 | 101 | ÿ | Krum |
| **Attack timing** | | | | | |
| 6a | 3 | Single-character attack every 10 epochs | 1 | ÿ | None |
| 6b | 3 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 6c | 3 | Single-character attack every 10 epochs | 201 | ÿ | None |
| 6d | 3 | Single-character attack every 10 epochs | 401 | ÿ | None |
| 6e | 3 | Single-character attack every 10 epochs | 501 | ÿ | None |
| **Attack frequencies and adversary counts** | | | | | |
| 7a | 1 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 7b | 3 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 7c | 5 | Single-character attack every 10 epochs | 101 | ÿ | None |
| 7d | 3 | Single-character attack every 20 epochs | 101 | ÿ | None |
| 7e | 3 | Single-character attack every 5 epochs | 101 | ÿ | None |

Table 1: Description of tasks

## References

[1] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[3] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020.

[4] M. Chen, A. T. Suresh, R. Mathews, A. Wong, C. Allauzen, F. Beaufays, and M. Riley, "Federated learning of n-gram language models," *arXiv preprint arXiv:1910.03432*, 2019.

[5] B. Y. Lin, C. He, Z. Zeng, H. Wang, Y. Huang, C. Dupuy, R. Gupta, M. Soltanolkotabi, X. Ren, and S. Avestimehr, "Fednlp: Benchmarking federated learning methods for natural language processing tasks," *arXiv preprint arXiv:2104.08815*, 2021.

[6] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, pp. 2938–2948, PMLR, 2020.

[8] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16070–16084, 2020.

[9] T. D. Nguyen, T. Nguyen, P. Le Nguyen, H. H. Pham, K. D. Doan, and K.-S. Wong, "Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107166, 2024.

[10] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, p. 828–841, Oct. 2019.

[11] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[12] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," 2018.

[13] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang, "Federated learning meets natural language processing: A survey," *arXiv preprint arXiv:2107.12603*, 2021.

[14] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and S. Y. Philip, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, 2022.

[15] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18, IEEE, 2017.

[16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[17] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.

[18] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)*, pp. 19–38, IEEE, 2017.

[19] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[20] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2020.

[21] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?," *arXiv preprint arXiv:1911.07963*, 2019.

[22] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, pp. 5650–5659, PMLR, 2018.

[23] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[24] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[26] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance

deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[28] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[29] Z. Zhang, A. Panda, L. Song, Y. Yang, M. W. Mahoney, J. E. Gonzalez, K. Ramchandran, and P. Mittal, "Neurotoxin: Durable backdoors in federated learning," 2022.

[30] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.