Cross-Validation for Detecting Label Poisoning Attacks: A Study on Random Forest Algorithm

Tharindu Lakshan Yasarathna[1(✉)], Lankeshwara Munasinghe[2], Harsha Kalutarage[2], and Nhien-An Le-Khac[1]

[1] School of Computer Science, University College Dublin, Belfield, Dublin, Ireland
tharindu.yasarathna@ucdconnect.ie
[2] School of Computing, Robert Gordon University, Aberdeen, UK
https://www.rgu.ac.uk/study/academic-schools/school-of-computing

Abstract. The widespread adoption of machine learning (ML) algorithms has revolutionized various aspects of modern life. However, their susceptibility to data poisoning attacks remains a significant concern due to their potential to compromise model integrity and performance. This study examines the impact of two types of data poisoning attacks on the Random Forest algorithm. It highlights the vulnerability of ML systems, especially in continual learning settings. We propose a simple yet effective strategy for continual learning ML systems to detect potential label poisoning attacks. This involves observing significant performance changes during model retraining. Experimental evaluation with Random Forest algorithms confirms the efficacy of the strategy in detecting and mitigating label poisoning attacks in continual learning systems.

Keywords: Data poisoning attacks · Continual learning · Machine learning · Cybersecurity · Random Forest

1 Introduction

Within the sphere of Cybersecurity, ML algorithms have emerged as highly successful tools in the realm of threat detection. For instance, these algorithms have demonstrated their prowess in identifying outliers in Biometric authentication systems, analyzing network traffic for irregularities, and unveiling concealed connections within social networks, as exemplified by the research in [16]. Despite their numerous advantages, ML algorithms are not without their limitations and shortcomings. For instance, their susceptibility to data poisoning attacks makes them vulnerable. Researchers have devised a range of solutions to mitigate the vulnerability of ML algorithms to such attacks. One prominent approach involves introducing diverse iterations of ML algorithms and incorporating novel enhancements into their structure, learning processes, or other key components. These refinements aim to bolster the algorithms' overall performance, resilience,

Published by Springer Nature Switzerland AG 2024
N. Pitropakis et al. (Eds.): SEC 2024, IFIP AICT 710, pp. 451–464, 2024.
https://doi.org/10.1007/978-3-031-65175-5_32

and security, particularly in the face of specific attack vectors [2]. However, it's important to note that these enhancements are highly context-dependent and, therefore, challenging to implement as generic, one-size-fits-all solutions. A prime example of this context dependency can be observed in continual learning settings, where ML models undergo periodic updates. In such scenarios, the task of modifying the ML algorithm at each update becomes both arduous and cost-intensive. Specifically, in continual learning environments, ML algorithms are retrained using new training datasets. This process, although essential, introduces a potential vulnerability that malicious actors can exploit to launch data poisoning attacks. In these attacks, the attacker manipulates the training data without altering the underlying algorithm. The goal is to deceive the ML model by introducing tainted data.

Motivated by this challenge, our research endeavours to propose straightforward yet effective detection strategies aimed at thwarting data poisoning attacks on ML algorithms in continual learning environments. The focus of this endeavour centres on label poisoning attacks, a prevalent form of data poisoning that targets ML algorithms. Our contributions can be succinctly summarized as follows:

- Conducted an in-depth analysis using the Random Forest algorithm to illustrate the vulnerabilities induced by data poisoning attacks.
- Devised a straightforward yet highly effective detection strategy tailored for label poisoning attacks on ML algorithms, with a particular focus on continual learning scenarios.

Remaining of this paper is structured as follows: Sect. 2 discusses related works, Sect. 3 explores vulnerabilities induced by data poisoning attacks, Sect. 4 presents the novel label poisoning attack detection and mitigation strategy with experimental results, and Sect. 5 concludes the paper with suggestions for future research directions.

2 Related Research

Data poisoning attacks pose a significant threat to machine learning (ML) model integrity, involving manipulating data during training, leading to susceptibility to adversarial inputs and undermining model reliability. As data grows bigger and more diverse, it becomes harder to understand its features. This gives attackers more opportunities to manipulate data from different sources [21]. ML models are fragile, with minor data alterations causing unexpected predictions. For instance, Typical deterministic ML algorithms such as decision-tree-learning algorithms are highly vulnerable to small perturbations of training data [5]. Attack methods include data poisoning, model theft, and evasion, with poisoning primarily achieved through data injection or modification [3]. Such attacks introduce bias and discrimination, perpetuating discriminatory behaviour and impacting security-sensitive applications [18]. For instance, Chang et al. [4] and

Suciu et al. [19] have examined the effects of evasion attacks on the RF algorithm. They have demonstrated that a mere 5% perturbation added to the test data can reduce evasion attack impact by up to 30%.

Past research has introduced various defensive and offensive approaches and methods to detect, prevent, and mitigate potential attacks. These strategies encompass a wide range of techniques aimed at safeguarding systems and data against threats. Defensive strategies against adversarial attacks encompass techniques such as data augmentation, increased model complexity, and adversarial training, as highlighted by [13,14]. While ensemble-based defences offer robust theoretical guarantees by utilizing multiple models trained on subsets of the training data and aggregating their predictions, they also come with a linear overhead. Despite their effectiveness, ensemble-based defences, which do not impose constraints on the base model, have not been extensively explored for enhancing the robustness of random forest models. Anisetti et al. [1] addressed this gap by introducing a novel hash-based ensemble approach tailored to protect random forest models against untargeted, random poisoning attacks. Tools like *Antidote* [5], *DeepArmour* [10], *AUROR* [17], and *K-LID-SVM* [22] address these challenges. For instance, *Antidote* enhances model security using PCA with a Laplace threshold, while *DeepArmour* detects adversarial scenarios effectively, maintaining high accuracy even after retraining with adversarial samples. *AUROR* detects malicious users with minimal accuracy loss, and *K-LID-SVM* reduces classification errors significantly while enhancing resistance to poisoning attacks.

3 Vulnerabilities of Machine Learning Algorithms Against Data Poisoning

For a deeper understanding of the vulnerabilities resulting from data poisoning attacks, we provide an in-depth analysis of the Random Forest (RF) algorithm's performance when subjected to two popular data poisoning attacks in this section.

3.1 Label Poisoning Attack on Random Forest

Data poisoning can take on different forms, as outlined in [7], with label poisoning being a prominent subtype. To assess the effect of label poisoning, we performed experiments using the Random Forest (RF) algorithm. These experiments involved training the algorithm with both unperturbed (non-poisoned) data and data deliberately tainted by label poisoning. Our analysis was performed on two datasets: the *Spambase* [11], which comprises classified email samples labelled as spam (1813 instances) and ham (2788 instances) with each record consisting of 56 features and the *YAHOO synthetic cloud network* dataset[1], containing 335,999 records, with each record consisting of 8 features generated by Yahoo servers. To mitigate computational expenses, we conducted experiments

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.

using a sample size of around 1800 attack records and 5000 benign records, as shown in Table 1. Both datasets were divided into training and test sets and poisoned the training datasets by labelling a random subset of benign instances (75% of the benign instances) as malicious. The data modification led to a significant decrease in the RF model's classification accuracy, ranging from 10% to 20%, compared to the accuracy of 96.3% achieved when trained with non-poisoned data, as indicated in Table 2.

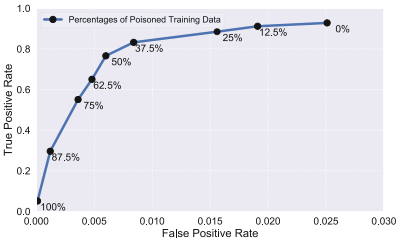
Table 1. Dataset Description of Data used in Poisoning Attacks

Dataset	Spambase data		YAHOO data	
Instances Used for Experiments	Spam	1813	Attack	1800
	Ham	2788	Benign	5000
Total Records	4601		335,999 (Used only 6800 records)	
Features	56		8	
Train: Test Split	70: 30		70: 30	
Percentage of poisoned benign instances	75%		75%	

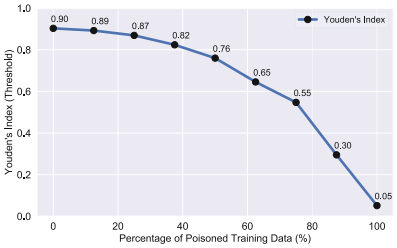
Furthermore, we conducted a systematic analysis of the RF model's performance by varying the percentage of poisoned data in the *Spambase* training dataset. The Receiver Operating Characteristic (ROC) curve in Fig. 1a depicts the substantial impact on classification accuracy as the poisoned percentage increases. The ROC curve provides a clear representation of the model's ability to distinguish between spam and benign instances across different poison rates. As poison rates increased, the ROC curves exhibited a decreasing capacity of the model to discriminate between true positive and false positive rates. Simultaneously, by computing the Youden Index [12] derived from these curves, shown in Fig. 1b, we precisely determined the optimal classification threshold that maximized the model's overall performance. This strategic utilization of the Youden Index enabled us to strike a balance between true and false positive rates, even when dealing with poisoned data, ensuring the model's robustness and efficacy against potential adversarial attacks.

Table 2. Results of data poisoning attacks on RF algorithm

Performance Metrics	Before poisoning attack on RF		After poisoning attack on RF	
	Spambase data	Yahoo data	Spambase data	Yahoo data
Accuracy(%)	96.30	99.44	80.13	90.36
Recall(%)	92.82	100	50.09	100
Precision(%)	97.67	100	98.91	88.60
F1 Score(%)	96.99	99.03	66.50	93.95
OOB error(%)	4.97	0	48.59	32.66



(a) Trade-off between true positive and false positive rates (*spambase* data).



(b) Youden Index Analysis: Optimal threshold maximizing true positive and true negative rates (*spambase* data).

Fig. 1. Comparison of ROC curve and Youden Index Analysis

The vulnerability of the RF algorithm can be elucidated as follows. RF employs the bagging technique to generate multiple independent decision trees with a majority voting mechanism, forming an ensemble. Samples that are not part of the bagging process and remain unseen during training are referred to as out-of-bag (OOB) samples. These OOB samples can be used as test data, and the classification error computed for them is known as the OOB error. In a label modification attack, significant weight is assigned to the benign class, and the random subsets chosen for the bagging process contain poisoned data. Consequently, the model struggles to establish an accurate decision boundary between spam and benign instances, resulting in a bias toward poisoned instances. As a consequence, during the testing phase, malicious objects are misclassified as benign. Figure 2 and Fig. 3 display the OOB error rates for the *Spambase* and *Yahoo synthetic cloud network* datasets both before (4.97% and 0%) and after (48.39% and 32.66%) the data poisoning attack on the RF algorithm. The substantial increase in the OOB error leads to a significant reduction in the accuracy of the resulting RF model.

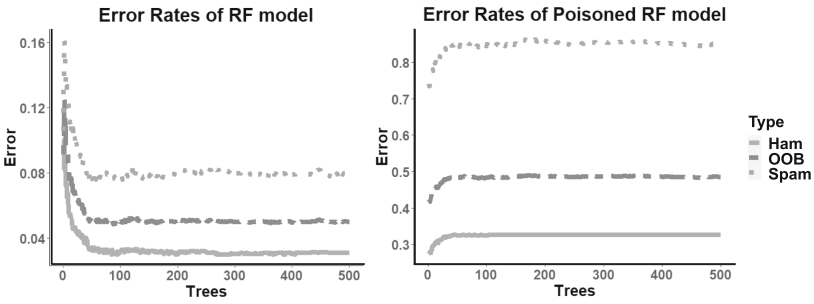


Fig. 2. RF model OOB error for spambase data set

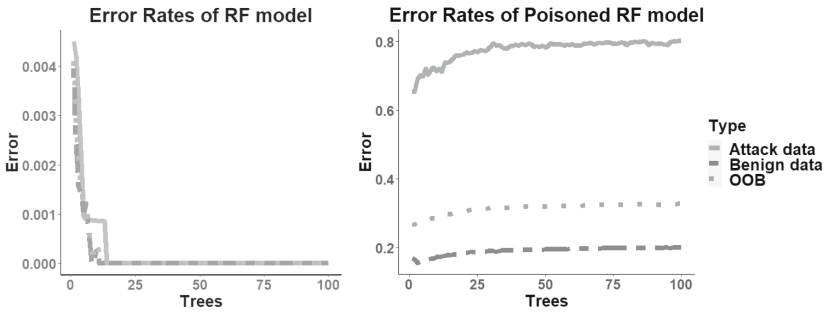


Fig. 3. RF model OOB error for YAHOO synthetic cloud network data set

3.2 Evasion Attack on Random Forest

Label poisoning can also take the form of data modification or evasion attacks, which involve altering data to mislead ML models into producing incorrect results [8]. Consequently, the computer vision community has been extensively researching evasion attacks on image and object recognition systems. For instance, consider their potential impact on an object recognition system in an autonomous vehicle. An adversary could deceive the ML algorithm by subtly altering a "Stop" sign, causing it to be recognized as "Go," with potentially catastrophic consequences. Similarly, facial recognition systems at airport security checkpoints can be tricked using adversarial glasses [15]. An another illustrative example of an evasion attack in image classification is presented in [9]. In this case, a photograph of a Panda, initially classified correctly, was misclassified as a Gibbon when carefully crafted noise was added. Despite the perturbation, a human observer can still recognize it as a Panda. There are two primary types of evasion attacks:

1. **White-box attacks:** Adversaries leverage information about the training algorithm, the distribution of training data, and the parameters of the ML algorithm to craft malicious test cases.
2. **Black-box attacks:** As the adversary lacks knowledge about the target algorithm and its specific parameters, they rely on information regarding the system's settings or previous inputs to identify vulnerabilities in the ML algorithm.

To illustrate an evasion attack on the RF algorithm, we conducted an experimental analysis using an SMS spam dataset [20]. This dataset contains 5574 observations categorized as either spam or ham (86.6% were Legitimate, and 13.4% were Spam). In our analysis, we divided the dataset into training and test sets with a 70:30 ratio. With non-poisoned data, the RF model achieved a classification accuracy of 97.11%, as indicated in Table 3. Subsequently, the test data was manipulated to bypass the spam filter by randomly altering selected ham messages to appear as spam. Following this manipulation, the classification accuracy decreased to 76.61%.

Table 3. RF model performance on SMS spam data before and after Evasion Attack

	Precision(%)	Recall(%)	Accuracy(%)
Before evasion attack on RF	97.07	99.67	97.11
After evasion attack on RF	87.11	85.96	76.61

In addition to the aforementioned experimental analysis, we explored the impact of evasion attacks on the image classification performance of RF algorithms using the widely recognized MNIST dataset [6]. In this investigation, we manipulated the image backgrounds by subtly shifting all zero (0) pixel values by just one (1) pixel. This subtle change is imperceptible to the human eye, as demonstrated in Fig. 4, where we present both the original image (Fig. 4-A) and the modified image (Fig. 4-B). Following the application of the RF model to both the unaltered and perturbed image data, we observed a noteworthy decrease in the RF model's accuracy, plummeting from 96.58% to 78.09%. This result underscores the high susceptibility of the RF algorithm to evasion attacks. We performed these experiments using various percentages of perturbation on the pixel values, as outlined in Table 4. Even with a 10% pixel noise perturbation, the changes in the background colour remained imperceptible to the naked eye, as illustrated in Fig. 5. Nevertheless, the classification accuracy exhibited a substantial decline, as depicted in Table 4. These findings emphasize that the RF algorithm's image classification performance significantly deteriorates even with a small perturbation, as low as 0.39% pixel perturbation.

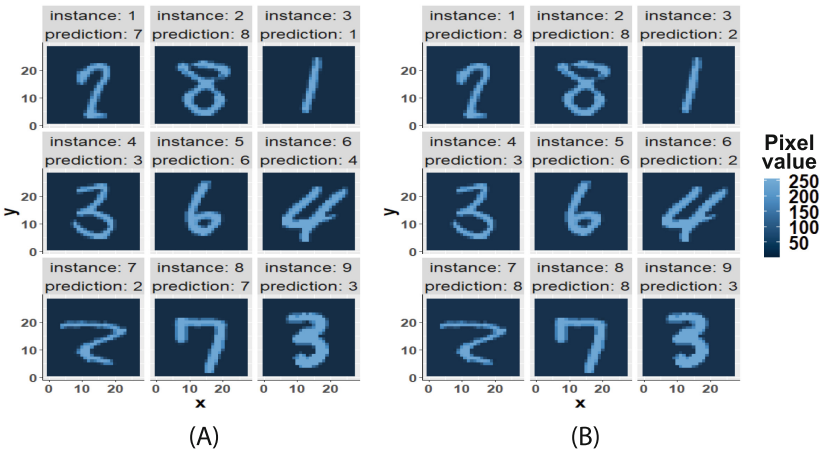


Fig. 4. Before(A) and after(B) the evasion attack with 1% pixel perturbation (can not see any significant change from naked eye)

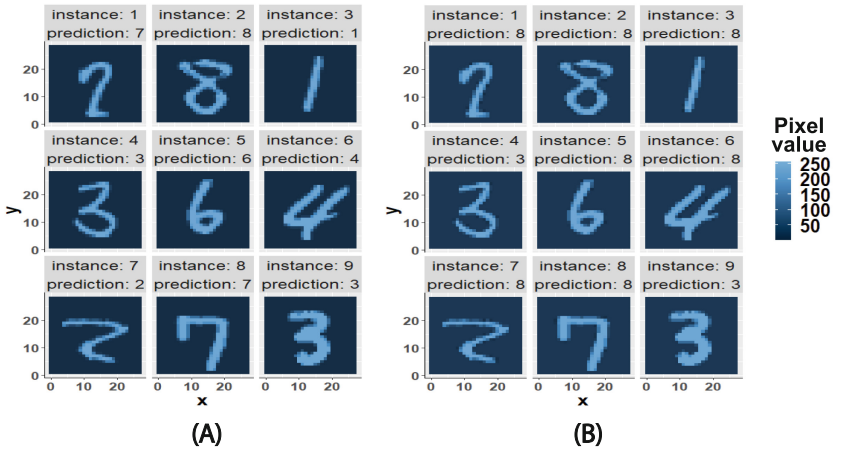


Fig. 5. Before(A) and after(B) the evasion attack with 10% pixel perturbation (can not see any significant change from naked eye)

Table 4. Pixel perturbation percentage and RF model performance on MNIST data

Pixel value perturbation (1% = 2.56 pixels)	RF model Accuracy (%)
0%	96.58
0.39%	78.09
0.5%	73.88
1%	54.30
5%	37.31
10%	33.04

The analysis above underscores the complexity and expense involved in adapting ML algorithms to defend against attacks. Furthermore, these adaptations are highly context-dependent. For instance, modifying the RF algorithm to guard against two distinct types of data poisoning attacks, such as label poisoning and evasion attacks, can be a formidable challenge. It's worth noting that these attacks may necessitate different modifications depending on the specific type of attack involved. Consequently, the most effective and efficient approach is to devise strategies for identifying flawed models before deploying them in systems. Keeping this in mind, we have introduced a novel yet straightforward strategy for detecting label-poisoning attacks on ML algorithms, particularly in continual learning settings.

4 Cross-Validation for Mitigating Label Poisoning Attacks

The initial step in countering an attack is to identify it. In continuous learning environments, like bio-metric authentication systems, ML models are continuously updated by training them with time series data. This process, however, presents an opportunity for attackers to launch data poisoning attacks. For instance, if an attacker gains access to the training data, they can manipulate the labels of attack data to resemble benign data or perturb the test data. In the first scenario, ML algorithms trained with incorrect labels generate erroneous ML models that misclassify both attack and benign data.

To address this concern, we propose a straightforward yet effective strategy for detecting potential label poisoning attacks. When the ML model is trained with balanced and non-poisoned data, it demonstrates higher accuracy in classifying unknown data instances. However, if the model is trained with poisoned data, it may not exhibit the same accuracy when classifying unknown data instances. This observation forms the basis of our simple yet effective strategy for identifying potential data poisoning attacks on ML models. Before delving into the specifics of our proposed strategy, it's essential to note that this method is designed to detect and mitigate label poisoning attacks, where the attacker only has full access to the training data and no access to the detection algorithm or its parameters. The proposed strategy operates as shown in Fig. 6.

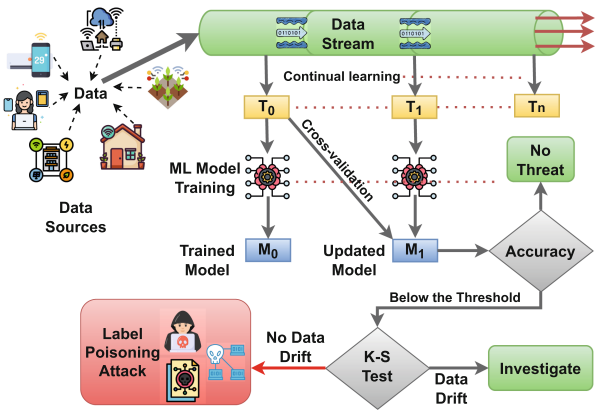


Fig. 6. Cross-validation strategy for detecting label poisoning attacks in continual learning systems.

Initially, the ML algorithm is trained using non-poisoned training data. The resulting model is preserved as the reference model, and the training data is stored as the reference training data. To illustrate, let's say we trained the algorithm at time T using a set of training data T_{Train}, resulting in the reference

model M_T. In a continuous learning setup, this model continually updates with new training instances at regular time intervals. However, an attacker with access to this training data may tamper with the labels, changing benign labels to spam labels in an attempt to deceive the algorithm into producing an erroneous model. Suppose that the attacker alters the labels of the training data at time $T + t$. Consequently, the algorithm learns a new model at time $T + t$ using the tainted training data, yielding an erroneous model M_{T+t}. To evaluate this new model, it undergoes cross-validation with the original training data T_{Train}. As the model M_{T+t} is trained with poisoned data, its classification accuracy is expected to drop significantly compared to the model M_T, which was trained with non-poisoned data. This noticeable decrease in accuracy serves as an indicator of a potential label poisoning attack. To confirm this as a likely attack, we establish a predefined threshold for the detection accuracy of the model. If the detection accuracy falls below this threshold, an alert is triggered to notify the presence of an attack. This entire process is outlined in Algorithm 1.

The threshold for detecting a potential label poisoning attack is established using the 10-fold cross-validation error of the model on the original training data, specifically on model M_T and its corresponding training data $Train_T$. In our study, we set this threshold at 0.05, based on the 10-fold cross-validation errors. While the maximum error rate is 0.03, our choice of a higher threshold is grounded in several justifications:

- We set a safety margin above the maximum error rate to strengthen the model's robustness against attacks, reducing the risk of false positives.
- Acknowledging real-world variability, we consider the potential for higher error rates not captured in observed data.
- The threshold balances false positives and false negatives, accepting controlled false positives to avoid overlooking potential attacks.
- Informed by domain knowledge and experience, a threshold of 0.05 aligns with our study's context.

Ultimately, this choice aims to balance security and operational efficiency while considering the potential consequences of missing label poisoning attacks in our particular setting.

The effectiveness of the proposed strategy was evaluated using the *YAHOO S5 - labelled anomaly detection dataset*[2], which is detailed in Table 5. This dataset consists of four features, each accompanied by a timestamp and corresponding labels. We intentionally selected this time series dataset because it closely resembles the streaming data encountered in a continual learning environment. To mimic a continual learning scenario, we periodically updated and evaluated the RF model at regular time intervals. Initially, the model was trained using a balanced dataset. In subsequent steps, new training datasets were generated based on the timestamp of the data points, where the arrival time was derived from the timestamp. With the exception of the initial training dataset, all other training datasets had their labels tampered with, flipping a certain

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.

Algorithm 1. Label Poisoning Detection

Input: Non-poisoned training data D_{train} at time t, Trained model M_t at time t, Detection error threshold d
Output: Alarm for potential poisoning attack
$M_{updated} \leftarrow M_t$
$D_{test} \leftarrow D_{train}$
DataDrift $\leftarrow False$
Initialize a counter for time instances: $t \leftarrow 0$
while continual learning stops **do**
    Increment the time instance: $t \leftarrow t + t'$
    NewTestData: $D_{test} \leftarrow D_{train}$
    NewTrainingData: $D_{train} \leftarrow D_{t+t'}$
    Train a new model with $D_{t+t'}$: $M_{updated} \leftarrow M_{t+t'}$
    Cross-validate $M_{updated}$ with D_{test}: DetectionError $\leftarrow d'$
    if DetectionError $> d$ **then**
        Check for DataDrift: $KSTest(D_{test}, D_{t+t'})$
        if No DataDrift **then**
            DataDrift $\leftarrow False$
            Notify Label Poisoning Attack
        end if
        DataDrift $\leftarrow True$
        Investigate Alarm
    end if
    No attack threat: Continue
end while

percentage of benign data, and our strategy was applied at each step to identify label poisoning. The outcomes of the experiments are presented in Table 6. Due to computational constraints and technical limitations, we limited the number of continual learning iterations to three. The experimental results indicate that there was no data drift in all three training data sets. However, when we increased the poison percentage for the labelled data, the detection error surpassed the error threshold when the poison percentage reached 1%, triggering alarms for a potential label poisoning attack. The data perturbation has notably not led to a significant decrease in recall. However, it has impacted the precision of the model, indicating an increase in false positives. A slight perturbation, flipping the benign labels into attacks, of just 1% can result in a significant increase in false positives.

Table 5. Data set description for evaluating the proposed strategy.

Dataset	Composition	Count
Yahoo S5 - labeled anomaly detection dataset, version 1.0	Benign instances	9000
	Anomaly instance	1800
	Total instances	10800
	Features	4

While this straightforward strategy exhibits promising results, there are certain challenges that need to be addressed. One of these challenges is data drifting, which can manifest in various forms. For instance, there's the concept drift, which occurs when the statistical properties of data change over time. For example, customer buying patterns changed during the COVID-19 pandemic, and such short-term shifts can lead to variations in predictions made by forecasting models. Another form of data drifting is covariate drift (or data drift), which happens when the values of training features change over time. For instance, a customer's salary and age may change over time, causing shifts in their buying preferences that might not be accurately predicted by a machine learning model trained on previous or older data due to this "data drift." Both of these drifting effects can occur in the training data of a continual learning setting. In our approach, we employ the Kolmogorov-Smirnov test (K-S test), a non-parametric test, to identify data drifting. Since all the features in our datasets are numerical, the K-S test is well-suited for our analysis. Our mitigation method triggers the K-S test if it detects any signs of an attack, such as a drop in model performance. If the K-S test provides no evidence of data drift, the mitigation system confirms the attack and raises alarms to alert of a potential attack.

Table 6. Detection performance of RF models in continual learning setting

Time	Model	Performance metrics				Detection error (threshold = 0.05)	Poisoned label (%)	K-S Test	
		accuracy	precision	recall	F1 score			P-Value($\alpha = 0.05$)	Remarks
t = 0	M_0	99.94	100	100	99.79	0.00057	0	–	N/A
t = 1	M_1	99.36	100	98.69	99.34	0.00063	0	–	N/A
t = 2	M_2	88.55	89.6	98.69	93.93	0.11448	1%	0.496	No data drift
t = 3	M_3	50.29	50.59	97.71	66.67	0.49710	5%	0.715	No data drift

5 Discussion and Conclusion

This study endeavours to explore the vulnerabilities inherent in ML algorithms and to devise straightforward yet effective mitigation strategies. To achieve this, we initiated an examination of ML algorithm vulnerability in the context of data poisoning attacks, with a specific focus on the Random Forest (RF) algorithm. Our experimental findings underscore the peril posed by data poisoning attacks, as they can mislead ML models into producing erroneous outputs. Even subtle alterations to the labels in the training data or the introduction of perturbations can result in a significant decline in model performance. Our research introduces a straightforward yet highly impactful strategy designed to detect and mitigate label poisoning attacks, particularly well-suited for continuous learning settings. The novelty of this strategy lies in its usage of non-poisoned training data to test or cross-validate retrained models in continuous learning scenarios. Our experimental results affirm the efficacy of this proposed strategy in recognizing potential data poisoning attacks and fortifying ML algorithms against them.

Moreover, in a continual training setting, the data may deviate from its initial distribution, causing a significant divergence in the model trained on drifted data compared to the previous model. To address this situation, we employ the Kolmogorov-Smirnov test (K-S test) to identify potential data drift. When the K-S test provides evidence of data drifting, it doesn't raise an alarm for a model deceived by a poisoning attack but instead triggers further investigation. Otherwise, our strategy raises alarms to flag potential label poisoning attacks.

In our future research endeavours, we aim to expand upon the presented strategy in several directions. Firstly, we plan to assess its generalizability by applying this strategy to various machine learning algorithms operating within continual learning environments. Secondly, we intend to investigate and mitigate potential attacks that exploit data drifting. Additionally, we are actively researching the applicability of one-class classification methods to handle both data drifting and data imbalance, making our strategy more robust. A significant priority for our future investigations is the development of a robust method for determining or estimating the error threshold for the proposed strategy. This is crucial because a slow-moving attacker could potentially infiltrate our strategy by gradually poisoning labels over an extended period. To counter this, we aim to test our strategy in such an environment and improve our current method for estimating the error threshold. Furthermore, our forthcoming work will emphasize the extensive expansion of experiments involving a wider range of datasets and different ML algorithms, particularly in real-world applications. This broader scope will provide a more comprehensive understanding of the strategy's performance in diverse and practical settings, thereby enhancing its applicability and overall robustness.

References

1. Anisetti, M., Ardagna, C.A., Balestrucci, A., Bena, N., Damiani, E., Yeun, C.Y.: On the robustness of random forest against untargeted data poisoning: an ensemble-based approach. IEEE Trans. Sustain. Comput. (2023)
2. Apruzzese, G., et al.: Addressing adversarial attacks against security systems based on machine learning. In: 2019 11th International Conference on Cyber Conflict (CyCon), vol. 900, pp. 1–18. IEEE (2019)
3. Chakraborty, A., et al.: Adversarial attacks and defences: a survey. ArXiv **abs/1810.00069** (2018)
4. Chang, J.Y., Im, E.G.: Data poisoning attack on random forest classification model. SMA 2020, 17–19 September 2020, Jeju, Republic of Korea (2020)
5. Drews, S., et al.: Proving data-poisoning robustness in decision trees. In: Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 1083–1097 (2020)
6. Alpaydin, E., Kaynak, C.: Optical Recognition of Handwritten Digits. UCI Machine Learning Repository (1998). https://doi.org/10.24432/C50P49
7. Fan, J., et al.: A survey on data poisoning attacks and defenses. In: 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), pp. 48–55. IEEE (2022)

8. Fleury, N., et al.: Malware: an overview on threats, detection and evasion attacks. arXiv preprint arXiv:2107.12873 (2021)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015)
10. Ji, Y., Bowman, B., Huang, H.H.: Securing malware cognitive systems against adversarial attacks. In: 2019 IEEE International Conference on Cognitive Computing (ICCC), pp. 1–9. IEEE (2019)
11. Mark, H., Reeber Erik, F.G., Jaap, S.: Spambase. UCI Machine Learning Repository (1999). https://doi.org/10.24432/C53G6X
12. Martínez-Camblor, P., Pardo-Fernández, J.C.: The Youden index in the generalized receiver operating characteristic curve context. Int. J. Biostat. **15**(1), 20180060 (2019)
13. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of artificial intelligence adversarial attack and defense technologies. Appl. Sci. **9**(5), 909 (2019)
14. Shafahi, A., et al.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
15. Sharif, M., et al.: Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540 (2016)
16. Shaukat, K., et al.: A survey on machine learning techniques for cyber security in the last decade. IEEE Access **8**, 222310–222354 (2020)
17. Shen, S., et al.: AUROR: defending against poisoning attacks in collaborative deep learning systems. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 508–519 (2016)
18. Solans, D., Biggio, B., Castillo, C.: Poisoning attacks on algorithmic fairness. In: Hutter, F., Kersting, K., Lijffijt, J., Valera, I. (eds.) ECML PKDD 2020. LNCS (LNAI), vol. 12457, pp. 162–177. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67658-2_10
19. Suciu, O., et al.: When does machine learning {FAIL}? Generalized transferability for evasion and poisoning attacks. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 1299–1316 (2018)
20. Tiago, A., Jos, H.: SMS Spam Collection. UCI Machine Learning Repository (2012). https://doi.org/10.24432/C5CC84
21. Wang, C., Chen, J., Yang, Y., Ma, X., Liu, J.: Poisoning attacks and countermeasures in intelligent networks: status quo and prospects. Digit. Commun. Netw. **8**(2), 225–234 (2022)
22. Weerasinghe, S., et al.: Defending support vector machines against data poisoning attacks. IEEE Trans. Inf. Forensics Secur. **16**, 2566–2578 (2021)