The robustness of popular multiclass machine learning models against poisoning attacks: Lessons and insights

Majdi Maabreh¹, Arwa Maabreh², Basheer Qolomany³ and Ala Al-Fuqaha⁴

Abstract
Despite the encouraging outcomes of machine learning and artificial intelligence applications, the safety of artificial intelligence–based systems is one of the most severe challenges that need further exploration. Data set poisoning is a severe problem that may lead to the corruption of machine learning models. The attacker injects data into the data set that are faulty or mislabeled by flipping the actual labels into the incorrect ones. The word “robustness” refers to a machine learning algorithm’s ability to cope with hostile situations. Here, instead of flipping the labels randomly, we use the clustering approach to choose the training samples for label changes to influence the classifiers’ performance and the distance-based anomaly detection capacity in quarantining the poisoned samples. According to our experiments on a benchmark data set, random label flipping may have a short-term negative impact on the classifier’s accuracy. Yet, an anomaly filter would discover on average 63% of them. On the contrary, the proposed clustering-based flipping might inject dormant poisoned samples until the number of poisoned samples is enough to influence the classifiers’ performance severely; on average, the same anomaly filter would discover 25% of them. We also highlight important lessons and observations during this experiment about the performance and robustness of popular multiclass learners against training data set–poisoning attacks that include: trade-offs, complexity, categories, poisoning resistance, and hyperparameter optimization.

Keywords
Poisoning attack, machine learning, clustering, multiclass, Big Data, artificial intelligence safety, deep learning

Date received: 1 February 2022; accepted: 2 May 2022

Handling Editor: Peio Lopez Iturri

Introduction
Machine learning (ML) and artificial intelligence (AI) have transformed many industries and addressed many of humanity’s challenges. They permeate our daily lives; people use popular smart devices and services such as Alexa, Amazon, Google Maps, and smart wearable gadgets even without realizing, sometimes, how they have been built. ML and AI applications are revolutionizing the productivity and workflow of several fields throughout the world, such as healthcare,¹,² education,³,⁴ transportation and road safety,⁵–⁷ farming and agriculture,⁸,⁹ smart energy and manufacturing,¹⁰,¹¹ and other fields.

¹Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University, Zarqa, Jordan
²Department of Statistics, Faculty of Science, Yarmouk University, Irbid, Jordan
³Cyber Systems Department, The University of Nebraska at Kearney, Kearney, NE, USA
⁴Division of Information & Computing Technology (ICT), College of Science & Engineering (CSE), Hamad Bin Khalifa University, Doha, Qatar

Corresponding author:
Majdi Maabreh, Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan.
Email: majdi@hu.edu.jo
clean environment and waste management, crime detection and policing, finance, pandemic management such as COVID-19, water quality and management, and many more. The data-driven and AI-enabled solutions have significantly increased the capacity to interpret enormous volumes of data created in today’s sensor-enabled environment. Due to the advancement in sensors and smart technology, we can now readily access numerous previously inaccessible areas and collect more data samples at an appealing price point.

Despite the excellent findings and potential of ML and AI applications, one of the primary challenges that need additional exploration and experimentation is the safety and robustness of the AI-based systems. These applications have potential vulnerabilities that might result in severe consequences and possibly damage. Today, advanced AI, Big Data solutions, data fusion techniques, and sensors technology facilitate the gathering process of data from different formats and sources. For example, in smart cities applications, data sources can be categorized into physical data sources such as sensors, cyber data sources such as social network data, participatory data sources, including crowdsensing and crowdsourcing, and hybrid data sources of the above. Crowdsourcing and crowdsensing might be vital for attackers to achieve their objectives against the smart systems. Data contributors can share some of their devices’ data (smartphones, wearable devices, etc.) to build a general model(s) for some applications. Hence, there is an opportunity to share some fabricated data that affect the performance of the learning algorithms and thus predictive models; this is known as a poisoning attack, which is the focus of this study.

The contribution of this work
Organizations, researchers, and AI developers cannot just sit back waiting for the attacks, to address the threat and evaluate the impact. Ongoing AI safety and robustness assessment through research and development may considerably impact businesses in identifying and mitigating vulnerabilities and possible damage before they are exploited and transformed into a breach and expensive loss. Due to the rapid evolution toward smart services and cities and the importance of safety issues, we believe exploring possible threats and weaknesses is no longer a luxury but a necessity to strengthen the AI immune systems.

In this study, we can summarize the contribution as follows

(a) We propose, formulate, and evaluate a possible label-flipping poisoning method that depends on some distance-based algorithm to create poisoned samples that pass through outlier detectors possibly deployed in a data cleaning phase.

(b) The study also evaluates the robustness of some commonly used ML algorithms, besides deep learning (DL), in response to fabricated data injected, somehow, into the training data set. The main objective is to study the behavior of the algorithms in rebuilding models using a mix of poisoned and benign data sets.

(c) We also show any Euclidean distance-based data filter challenge when attackers inject poisoning data samples into the benign data repository. In addition, we evaluate the RKOF (Robust Kernel-based Outlier Factor) algorithm if applied to quarantine poisoned samples using the proposed method and random label method as anomaly data samples. This method is developed based on a variable kernel density estimate of an object’s neighborhood. It is implemented in the “OutlierDetection,” R library version 0.1.1.

(d) The study provides the research and development (R&D) teams in AI safety with important lessons, insights, and observations in smart services and cities for further collaboration.

Related work
Attacks against ML
The attacker’s objective, knowledge, and tactics are frequently used to characterize attacks against AI and ML models. Attackers may use adversarial samples with a specific aim in mind, causing the model to incorrectly categorize samples into a particular class (label), known as a targeted attack. In an untargeted attack, the attackers aim to provide manipulated samples incorrectly categorized to any other class (label). A white-box scenario allows attackers to have complete access to the model, data set, and training parameters in terms of attacker knowledge. In contrast, a black-box scenario allows attackers to access the model. Attackers’ tactics (or strategy) involve the type of perturbations, the functions that create them, and how the attackers may launch the attack.

Several strategies are discussed in the literature to attack smart systems such as poisoning attacks, evasion attacks, and model extraction. Interested readers are encouraged to refer to a recent publication.

The proposed method in this study is classified as a poisoning attack. Generally, attackers share some manipulated samples of data to be consumed by the learning algorithm when it is updating the model on new training data collected, for example, by
crowdsensing or crowdsourcing systems from the physical world, that, in turn, could push the learning quality to the worst level. The manipulated data can be of different formats such as images, text, or audio. This could introduce a concern about the reliability and trustiness of the shared data. Moreover, despite the fact that majority of research focused on DL in studying the adversarial attacks, other algorithms are also evaluated, such as artificial neural networks (ANNs), support vector machine (SVM), Random Forest (RF), and Naïve Bayes (NB).

**Label-flipping poisoning attack**

One of the techniques used to poison a data set is creating adversarial examples by changing the true labels of the features vectors to other false labels selected from the set of labels of the data set (also known as Label-flipping attack). Thus, the classifier inaccuracy on the testing data samples will be maximized after any retraining session on the adversarial data.

In the study by Biggio et al., the gradient ascent strategy is utilized to maximize the testing error of the support vector machine. The attack algorithm starts by cloning a random data sample and flipping its label, considering the sample is close to the boundary of the attacking class. Results showed that the poisoning algorithm significantly impacted the SVM’s binary classification problems. On three different data sets of spam base used for binary classification, experiments by Zhang et al. were conducted to evaluate five ML and two DL algorithms. The data sets were poisoned using three different label-flipping algorithms; random flipping, entropy method, and k-medoids. Results showed that the flipping attacks could increase the testing error of the NB classifier to around 30% by poisoning 20% of the data samples. The attack impact of the entropy approach becomes more visible as the noise level rises, and it outperforms other methods, according to the data. In Shanthini et al.’s study, the robustness of three boosting learners was tested on three medical data sets under the effect of features and label noise. The findings of the experiments reveal that label noise does significantly more harm than feature noise. XGBoost has proven to be the most powerful algorithm under the label flipping attack in binary class data sets. The authors in the study by Xiao et al. discussed and evaluated various types of label noise assault, even if the attacker alters just a tiny proportion of the training labels, methods that might considerably impair the SVM’s classification performance on new test data. In the study by Taheri et al., the K-mean clustering algorithm is used to cluster the training data set into two main clusters. For each data sample and corresponding predicted label, the silhouette value (SV) is calculated. Since the values of SV range between 1 and −1, the data sample of SV close to 1 means it fits into that cluster, and the ones less than 0 are clustered incorrectly. The poisoning flipping algorithm most likely picked the samples that could be in the other cluster; SV values less than 0. This study uses K-mean to cluster the training data set into K clusters where the K values are picked as the best value based on the minimum validation error.

In this article, we suggest a clustering-based label-flipping attack that poisons data samples with distinct labels that are grouped in the same cluster. To poison the sample data, we suggest a strategy that promotes the notion of class label flipping. Instead of randomly flipping the labels of the collected samples, which are contributed to the data set repository by different users through crowdsensing systems, for example, this approach targets a specific set of samples that could balance between increasing the overall loss of the ML/DL model and the probability of detecting those poisoned samples by outlier detectors during the data cleaning phase.

**Clustering-based label-flipping attack**

These attacks happen early in the process, during the AI system's creation and training. It frequently includes manipulating the data used to train the system. We consider a situation in which data samples are added in a continuous or online learning context. We consider the problem of learning multiclass classifiers over a data set \( D \) of size \( n \) training samples. Each data sample \( x, x \subseteq \mathbb{R}^k \), consists of \( k \) features, \( x = (x_1, x_2, x_3, ..., x_k) \) and represents one class, \( y \in \{y_1, y_2, y_3, ..., y_m\} \). Using the required set of parameters each learner needs, the classifier’s output is \( F(x) \), which is evaluated using a separate unseen data set (testing labeled data), \( S \) of size \( t \). The classifier’s performance depends mainly on two inputs; the result of the classifier (\( F(x) \)) and the true label \( y \) in the testing data set \( S \).

The clustering-based attack proposed and evaluated in this study shares the same steps as the randomly flipping attacks. It has the same ultimate goal of decreasing the performance of the classifiers. However, not all samples in the data set will be used in the flipping process. We formulate the clustering-based flipping attack as follows:

Given a benign training data set \( D \) of size \( n \) and a benign testing data set \( S \) of size \( t \)

\[
D = \{(x_i, y_i)\}_{i=1}^{n} \\
S = \{(x_{test_j}, y_{test_j})\}_{j=1}^{t}
\]

where \( x, x_{test} \subseteq \mathbb{R}^k \), \( x = (x_1, x_2, x_3, ..., x_k) \), \( x_{test} = (x_1, x_2, x_3, ..., x_k) \), and a category (label) \( y, y_{test} \in \{y_1, y_2, y_3, ..., y_m\} \). Let \( F \) denotes the classifier
generated from $A$, a multiclass learning algorithm trained on $D$, and $F_{pos}$ denotes the classifier generated from the same algorithm $A$ trained on the poisoning data set $D_{pos}$. $F(x_{test})$ and $F_{pos}(x_{test})$ are the results of $F$ and $F_{pos}$ on $S$, testing data set, respectively. The clustering-based flipping attack function $P$ creates the poisoning data set $D_{pos}$ from $D$ by finding a subset of $D$, $(x_a, y_a)$ and $(x_b, y_b)$ where

1. $y_a \neq y_b$, and
2. $\arg\min_{x \in D} \sqrt{\sum_{i=1}^{k} (x_{a,i} - x_{b,i})^2}$

to flip their classes in the form $(x_{pos}, y_{pos})$ such as $(x_a, y_a)$ and $(x_b, y_b)$. The above states that the distance between the data samples in the selected subset; $x_a, x_b$ is $\varepsilon$, that is to increase the probability of passing through an outlier detection function, to filter out spurious data samples per class of a cut-off rate $\alpha$, (i.e. $\varepsilon < \alpha$). The ultimate goal of $P$ is to decrease the performance of the classifiers by increasing the loss $L$ in the learning function as the following optimization problem

3. $\arg\max_{r_{pos}} \frac{1}{t} \sum_{j=1}^{t} L(F_{pos}(x_{test_j}), y_{test_j})$ s.t. $F_{pos} = A(D_{pos} \cup D)$

We substitute the K-means clustering algorithm to the above function $P$. K-means partition the data samples into clusters, $C$, to minimize the within-cluster variance. Formally, the objective of K-mean is

$$\arg\min_{i} \sum_{i=1}^{C} \sum_{x \in C_i} ||x - \mu_i||^2$$

where $\mu_i$ is the mean of the data sample in cluster $i$. Figure 1 illustrates the clustering-based selection of the data samples being poisoned. While the label-flipping attack could randomly flip the labels, the clustering-based label flipping has another objective to decrease the probability of the outlier detector quarantining those poisoning data. This approach depends on the quality of the clustering process and how the data samples are distributed over the clusters. Generally, this attack could be classified as untargeted attack; however, a special case makes it targeted attack, which is that some clusters consist of samples of one label, while others consist of mixed data samples of different labels. One of the assumptions in poisoning attacks is the ability of the attackers to contribute to the data set, which might have some knowledge about the data set properties or features. Figure 2 graphically shows the steps of clustering-based poisoning attacks proposed in this study.

### Method and data

This section describes the experiment environment and method to poison data samples usually collected by crowdsensing systems. It also describes the data set and the learning algorithms used in this study. Four primary metrics are used to evaluate the performance of all classifiers; namely, overall accuracy, Macro-precision, Macro-recall, and Macro-F1 score.

#### Multiclass learning algorithms

We evaluate commonly used multiclass learning algorithms used in ML/DL-based solutions. Namely, (1) ANN, (2) K-nearest neighbor (KNN), (3) boosted decision trees (BDT), (4) RF, (5) one-versus-rest support vector machine (6) one-versus-one support vector machine, (7) one-versus-rest NB, (8) one-versus-one NB, (9) one-versus-rest ANN, (10) one-versus-one ANN, (11) one-versus-rest BDT, (12) one-versus-one BDT, and (13) DL. The setting for each algorithm will be discussed in the subsequent sections. All classifiers are trained using 10-fold cross-validation and evaluated on an unseen testing data set.

**ANN algorithm.** The ANN has been trained on the default parameters, one hidden layer fully connected to both input and output layers. The hidden layer has 100 nodes. The learning rate was set to 0.4, the momentum was set to 0, and the shuffle example option was enabled (i.e. TRUE).

**BDT algorithm.** BDT is an ensemble ML of multiple weak learners, decision trees in this case. Each tree is dependent on the preceded tree during the learning process to improve the overall accuracy. BDT has been trained using 20 leaves per tree, a maximum of 10 samples per leaf node, the learning rate was set to 0.2, and the number of trees constructed is 100.

**KNN algorithm.** KNN is one of the popular supervised ML algorithms utilized in several ML-based applications. It mainly depends on the idea of objects’ similarity or proximity. Although it is simple to apply and use, KNN models’ performance could vary, sometimes significantly, using different values of $K$ on other data sets. In this study, the “caret” R library has been utilized to find the best value of $K$ for the benign (intact) training data set. Using 40 iterations and the accuracy values returned, “caret” suggested using $K = 53$. Note that the accuracy was collected using cross-validation of 10 folds repeated three times on the benign training data set. The testing data set is not part of the optimization process.
RF algorithm. RF, one of the popular ML algorithms, shows success stories in different domains. The RF method combines the predictions of several decision trees of various depths. Every decision tree is trained on a bootstrapped data set, which is a subset of the entire benign training data set, and the left subset is used for validation. The RF models in this study were built using the “randomforest” R library.

SVM (one-vs-rest). SVM might be considered one of the most common ML algorithms used in different disciplines. For a two-class classification problem, SVM tries to find a decision boundary to separate the samples in \( n \)-dimensional space, where \( n \) is the number of features in the data set. One-versus-all is a heuristic method that uses binary classification algorithms to solve a multiclass classification problem. It entails breaking down the multiclass data set into many binary classification tasks. On each binary classification task, a binary classifier is trained, and prediction is made using the most confident classifier. It requires building one model for each class; in this study, and since the data set has six distinct classes, this method would create six models.

SVM (one-vs-one). The one-versus-one heuristic method also splits the training data set into binary classification tasks. However, not for each class versus others, but for each class versus every other class. Every classifier would vote for a label, and the most predictions or votes will be returned by this method. This method would build \( (6 \times 5)/2 = 15 \) classifiers for the voting phase for six distinct classes.

One-versus-one and one-versus-rest strategies. In this study, we evaluate these strategies with SVM, NB, and even with those learning algorithms supporting multiclass classification problems such as ANN and BDT. It could be helpful to evaluate the effect of poisoning attacks on models when they were built using these strategies.

DL algorithm. Compared to the afore-mentioned neural networks, DL utilizes multiple hidden layers. It can be seen that DL is a neural network that consists of more than three layers; an input layer, multiple hidden layers, and an output layer. DL shows many success stories, especially in the era of Big Data, in different domains. In this study, we utilize the “h2o” R library to train the model and study the effect of poisoning attacks on the DL models. While the DL algorithm has many parameters that could impact the performance of the DL-based system, we use particle swarm optimization to find the number of layers and neurons to be used in building DL on benign data sets. PSO suggests two layers of 50 neurons each for the benign data set.

Data set
Training and testing data sets used in this study are publicly available. It is a benchmark human activity recognition collected from smartphones and smartwatches, the features’ values are collected from two motion sensors in smartphones that represent six different activities, (“Biking,” “Sitting,” “Standing,” “Walking,” “Stair Up,” “Stair Down”). In this study, we use the main features generated by smartphone sensors and ignore other features related to the original experiment, such as index, creation time, arrival time,
user, model, and device. We focus on the three main readings; “x,” “y,” and “z” generated from Nexus smartphone sensors for one user; user “a.” The used data set consists of 218,178 training records and 72,722 testing records. Each record (sample) has three features and one activity label, which is a multiclass label for the six activities mentioned above.

Results and discussion

Algorithms performance

ANN performance. Figure 3 shows the four metrics used to evaluate the performance of ANN. The model loses its accuracy as the poisoning rate increases. Using the clustering-based method, the ANN model accuracy was affected and dropped when the poisoning rates were 40% and 50%, where the accuracy was 68.5% (~5% loss) and 64.7% (~9% loss), respectively. Flipping the labels of samples grouped in different clusters has more effect on the performance of ANN models. The accuracy dropped to below 50% (i.e. 42.8%, ~31% loss) after increasing the poisoning rate to 50%.

BDT performance. BDT, in Figure 4, shows some sort of robustness against clustering-based poisoning till half of the training data set gets poisoning. The accuracy dropped significantly after the clustering-based method flipped 50% of the training data; the accuracy loss is almost 10%. The same poisoning rate (i.e. 50%) was applied using samples of different clusters, and the BDT model cannot achieve more than 41% accuracy (~30% loss). However, the BDT model is also showing some stability in the accuracy even with this method of flipping till the poisoning rate exceeds 40%.

KNN performance. In Figure 5, the KNN algorithm maintained a stable performance even with some distorted samples, up to 30% poisoning rate using both poisoning methods. KNN loses the ability to resist the poisoning when the poisoning rates get more...
prominent than 30%. In addition, KNN outperforms the other algorithms in this investigation, save from RF.

**RF performance.** Figure 6 depicts the performance of the RF. RF maintains an almost stable performance until 40% of the training data becomes poisoned. However, 50% of poisoned samples using the clustering-based flipping method decreased the model performance by 13.5%, while using random flipping has more effect on the 50% poisoning rate and dropped the accuracy by ∼33%. The impact of random flipping started since 30% of the training data are poisoned, while the effect of clustering-based flipping would need more poisoned samples than 30% to show some impact on the performance.

**SVM performance.** Figures 7 and 8 show the performance of the support vector machine using both approaches; one-versus-rest and one-versus-one. SVM performance is comparable to RF and KNN in its resistance to poisoning rates of less than 30% using both poisoning methods. However, the SVM with the one-versus-one strategy improves when the poisoning rates increase, such as 30% and 50% poisoning rates using the clustering-based method. There is a slight difference in the accuracy and stability of the performance with increasing the poisoning rates between the one-versus-rest and one-versus-one approaches. This observation could be considered a starting point for more extensive experiments and evaluation on large data sets and different types of attacks, yet beyond the scope of this study.

**NB performance.** In this study, Figures 9 and 10 show that NB models are the most sensitive models compared to the random label-flipping poisoning attacks. The performance drops by around 8%, with every 10% rise in poisoning, similar to linear behavior. It is worth noting that, while utilizing the clustering-based method, performance rises and falls in proportion to the
percentage of poisoning. On 50% of the poisoned data, accuracy rebounded to around 65%, with just a 5% reduction on the original benign data. This loss percentage may be unacceptable in some applications. Still, it is not significant if we know the data are not benign and has been modified to influence the model’s performance.

One-versus-one and one-versus-rest strategies. Figures 11–14 show the performance of ANN and BDT if utilized as binary classification algorithms in one-versus-one and one-versus-rest strategies. The findings show no substantial difference in the outcomes when these strategies are used. Results also showed that, given the data set and poisoning methods in this study, there is no significant difference in reaction to the poisoning rates between one-versus-one and one-versus-rest on NB or SVM binary classification algorithms.

The performance of the above algorithms in this experiment, which is a temporary situation with poisoning rates less than 30%, with newly collected samples could be considered as a justification to trust the data source, which, in turn, would allow the attackers to inject more fake samples. From another point of view, the clustering-based poisoning attack needs to flip more than 30% of training samples to be effective enough to mislead these algorithms. The injected manipulated samples are in dormancy time until the poisoning rate becomes enough to degrade the performance of the models. Sanitizing the data might be costly, especially after multiple re-training phases, and not simple unless the AI engineers decide to return to the starting point and forego the new data and its gathering cost. Whatever the taken decision at that point, the loss has already occurred.

DL performance. Figure 15 depicts the performance of the DL algorithm. Interestingly, as opposed to other algorithms evaluated in this study, the DL models
behavior is not predictive of increasing poisoning rates either using clustering-based flipping or randomly flipping over different clusters. Randomly flipping 10% or 20% of samples labels is enough to drop the accuracy down to 60% (~10% loss) and 51.8% (~20% loss), respectively. However, the overall performance gets back to improve sometimes with more poisoning samples injected into the training data sets, such as from

Figure 9. Naive Bayes (one-vs-all) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score.

Figure 10. Naive Bayes (one-vs-one) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score.

Figure 11. ANN (one-vs-all) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score. ANN: artificial neural network.
20% to 30% and from 40% to 50% poisoning rates. On the contrary, the performance has gradually crashed with the increment of the poisoning rates using the clustering-based method. Sometimes, the model maintains the performance with more poisoning samples. However, the maximum loss with 50% of poisoned samples using the clustering-based method does not exceed 11%.

**Figure 12.** ANN (one-vs-one) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score. ANN: artificial neural network.

**Figure 13.** BDT (one-vs-all) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score. BDT: boosted decision tree.

**Figure 14.** BDT (one-vs-one) performance: (a) accuracy, (b) macro recall, (c) macro precision, and (d) F1-score. BDT: boosted decision tree.
Compared to other algorithms used in this study, the results of DL reveal how sensitive the DL models are to the poisoning attacks. Although the findings revealed a pattern of diminishing performance with an increase in the proportion of poisoning, DL may return with unexpected results, necessitating a thorough examination of a variety of data in various forms and dimensions. Also, when using these algorithms in real-world applications, the periods required to re-evaluate the models may need to be adjusted anytime new data are introduced to the existing ones in the data store.

**Lessons and insights**

*Distance measures and anomaly detection*

We conducted experiments to evaluate the efficacy of distance-based outlier detectors, if utilized, in detecting the poisoned samples. We also show the results of the RKOF method on the poisoning training data sets. Figure 17 depicts the distance among features’ vectors of different classes. We generate a mean vector for each class, which is simply the mean value of each feature, to summarize each class’ distribution of features’ values in one vector. After that, we applied the Euclidean distance algorithm to calculate the distance between the mean vector for each class and other classes’ mean vectors grouped in the same cluster and grouped in different clusters. The main objective is to show those data samples that the clustering-based attack can target to flip their labels to other labels. Figure 17 shows that attackers would pick the data samples of different classes belonging to the same cluster using the clustering-based attack to increase the probability of passing through any distance-based filter possibly applied in the data cleaning phase. Attackers might target the data samples clustered in cluster 1, “stair down,” for example, to flip their labels into “biking” to settle most of them in the training data set for any future re-training session of the model.

---

**Figure 15.** Deep Learning Performance: (a) Accuracy, (b) Macro Recall, (c) Macro Precision, and (d) F1-Score.

**Figure 16.** Deep Learning Performance with PSO for each poisoning rate: (a) Accuracy (b) Macro Recall (c) Macro Precision (d) F1-Score.
The above are unsurprising results since the K-mean clustering algorithm depends on the distance to group the data samples into clusters. However, it could be one of the attackers' options to poison the data samples straightforwardly. The clustering algorithms are currently available even in drag and drop tools and can be easily applied without deep knowledge in programming or the AI field.

Table 1 shows the results of applying RKOF outlier detector algorithms. We pick a subset of the data set that contains 10% of poisoned samples once using the clustering-based poisoning attack and again using the second poisoning method; over different clusters. The RKOF was able to identify, on average, 25% of the poisoned data samples using the clustering-based method, and 7% of benign data samples were sacrificed. Using the second method of poisoning data samples over different clusters, RKOF was able to identify, on average, 63% of the poisoned data samples, and only 2.9% of benign samples were sacrificed.

**Detection rates of RKOF where 10% of the samples are poisoned using both methods evaluated in this study.**

| Poisoning method                        | Average detection rate (%) |
|-----------------------------------------|----------------------------|
|                                        | Benign samples | Poisoned samples |
| Clustering-based                        | 7.0            | 25.0              |
| Random over different clusters          | 2.9            | 63.0              |

RKOF: robust kernel-based outlier factor.

**Detectability versus effectiveness**

Based on the above results of the RKOF algorithm, attackers might successfully inject 75% of the poisoned samples using the clustering-based method into the training data set. In comparison, they might be able to inject 37% of the poisoned samples using flipping labels of samples grouped into different clusters. The above results probably encourage attackers, yet challenges for the defense system developers. The results show clear evidence of the trade-off between the attack’s detectability and its effectiveness on the AI models’ performance. The clustering-based poisoning method may be a decent alternative for attackers who want to insert some poisoned samples into the benign training data. Still, the effect on the learning algorithms and models' performance may take some time until the proportion of poisoned samples is significant. In contrast, the second strategy to flip the labels of a sample belonging to different clusters may affect the performance with a lower percentage of poisoned samples but with a higher probability to be quarantined by outlier detectors or data filters.

**Poisoning attacks complexity**

From the attackers’ point of view, the clustering-based strategy might perform better in terms of detectability but on the costs of complexity and implementation. This would create another trade-off between complexity and detectability. Despite the availability of the clustering algorithm and the simplicity of some tools that offer ML/DL algorithm, such as Microsoft Azure ML studio, the process of data samples evaluation to be involved in the poisoning phase could consume some time and resources. The objectives are the ones that might have the maximum effect on the model's performance and also can pass through the detectors if applied. In this study, random label flipping could be much cheaper in preparation and implementation than the clustering-based method. However, it could be easier for the detector to end up with a high percentage of poisoned samples in the quarantine. From another point of view, clustering-based poisoned samples that could pass through the detectors might not have immediate effects. Moreover, picking the candidate samples with multiclass data sets would add another complexity factor since the flipping is not random but should be done within a single cluster. At the same time, we might need to keep the balance of the classes similar to the original data set.

Furthermore, one noteworthy discovery is the ease with which clustering-based assaults may be applied in crowdsensing systems; clustering algorithms are accessible and often supplied as a service, for example, ML as a service (MLaaS), which could not need a profound grasp of how to develop a data poisoning technique; inexperienced or novice ML users may be potential attackers in this case.

**Multiclass classification algorithms and strategies**

Multiclass classification problems can be solved using a variety of algorithms that support this type of problem, such as ANN and DT, as well as algorithms that were initially developed for binary classification problems, such as SVM and NB, by dividing the multiclass classification problem into several binary classification problems, such as one-versus-one or one-versus-rest. This
investigation discovered that utilizing one-versus-one or one-versus-rest with the original multiclass classification methods had no discernible influence on poisoning attacks or the learning process. ANN and BDT models behave similarly whether they were fitted directly on the poisoned data set or by a third party; binary models by using a one-versus-one or one-versus-rest approach. One more observation is that there is no preference for one when comparing one-versus-one and one-versus-rest on dealing with poisoned training data sets. Therefore, the decision is up to the ML developers to choose according to other criteria such as training time.

**DL and hyperparameter optimization**

Figure 15 shows DL models trained on data sets with varying poisoning rates but utilized the same learning parameter values as the benign data set. After gathering new data, ML engineers might consider and assess updating the learning parameters. Models of better performance might be achieved with a new data set and modified parameters. When it comes to poisoning assaults, though, this might be an issue. Figure 16 shows how the PSO technique in the study by Qolomany et al.\(^5\) was used to optimize the performance of DL parameters. The findings revealed that using parameters tuning in DL, the effect of a poisoned data set added to the initial benign data set might be hidden or diminished. Regardless of the poisoning approach used, clustering-based or random, the PSO effectively identified alternative parameter values that improved the DL models’ performance, even though the poisoning rate is constantly growing. When comparing Figures 15 and 16, parameter tweaking might be a trap, causing the models to anticipate different things than they should. The performance measurements probably still show values within the user-defined performance boundaries.

**Poisoning resistance**

According to the results of the learning algorithms on the data set used in this study, we may categorize the effect of poisoning rates and models resistance into three categories: (1) steady then fail, such as KNN, BDT, and SVM (one-versus-one); (2) gradually fail, such as RF, ANN, SVM (one-versus-rest), and NB; (3) zigzag pattern or irregular fail, such as DL, and NB with the clustering-based method. The first category shows more robustness against poisoning attacks. It might be favored to be used with simple filters of poisoned data samples as they can resist almost 30% of the poisoning rate. The second and third categories show more sensitivity to poisoning data samples, which could need more advanced and efficient filters of poisoned data samples to protect the performance of the systems.

From another point of view, the algorithms in the second and third categories may consume extra data characteristics when building the model, causing the model to perceive different details on benign test data. In general, we cannot decide for sure whether algorithms are superior, whether they are more resistant, or more sensitive; more research in this area, we believe, is required.

**Conclusion**

We propose, define, and evaluate a feasible label-flipping poisoning approach based on clustering algorithms. The main objective is to produce poisoned training samples that affect the accuracy of the classifiers while passing through outlier detectors. Using popular multiclass learners and a benchmark data set, the proposed method results are compared to the extreme case of flipping labels of data samples belonging to different clusters. The clustering-based poisoning strategy may be a good option for attackers, but the effect on the learning algorithms and models’ performance may take some time before the proportion of poisoned samples becomes significant, whereas the random strategy, flipping the labels of samples belonging to different clusters, may affect the performance of the models with less amount of poisoned data. However,
the clustering-based attack could increase the probability of passing through the outlier detectors while most of the data samples that are assigned random labels could be quarantined. This simply shows the trade-off between the effectiveness and detectability of label-flipping attacks. Moreover, one notable observation is the simplicity of applying clustering-based attacks in crowdsensing systems, which means inexperienced or beginner ML users could be potential attackers. We also identified three failure behaviors of multiclass learners in response to the poisoning rates, which, in turn, might help in identifying the complexity of data cleaners before injecting them into the training data sets. An important lesson was about the optimization of hyperparameters of DL models during the re-training phases. This optimization over the poisoned data set might cover the effect of poisoned data samples and mislead the AI engineer by improving the performance, even though the rate of poisoning is growing.

In general, when employing these algorithms in real-world applications, the time intervals for re-evaluating the models may need to be adjusted whenever new data are added to the data storage.

Acknowledgement
The Deanship of Scientific Research at the Hashemite University, Jordan provided resources supporting this work.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD
Majdi Maabreh https://orcid.org/0000-0003-4822-417X

References
1. Castiglioni I, Rundo L, Codari M, et al. AI applications to medical images: from machine learning to deep learning. Phys Medica 2021; 83: 9–24.
2. Lee DH and Yoon SN. Application of artificial intelligence-based technologies in the healthcare industry: opportunities and challenges. Int J Environ Res Public Health 2021; 18(1): 271.
3. Tahiru F. AI in education: a systematic literature review. J Cases Inf Technol 2021; 23(1): 1–20.
4. Lee HS and Lee J. Applying artificial intelligence in physical education and future perspectives. Sustainability 2021; 13(1): 351.
5. En gland C, Aksoy EE, Alonso-Fernandez F, et al. AI perspectives in Smart Cities and Communities to enable road vehicle automation and smart traffic control. Smart Cities 2021; 4(2): 783–802.
6. Lee S, Kim Y, Kahng H, et al. Intelligent traffic control for autonomous vehicle systems based on machine learning. Expert Syst Appl 2020; 144: 110747.
7. Silva PB, Andrade M and Ferreira S. Machine learning applied to road safety modeling: a systematic literature review. J Traffic Transport Eng 2020; 7: 775–790.
8. Moysiadis V, Sarigiannidis P, Vitas V, et al. Smart farming in Europe. Comput Sci Rev 2021; 39: 100345.
9. Woffert S, Ge L, Verdouw C, et al. Big data in Smart Farming—a review. Agr Syst 2017; 153: 69–80.
10. Kim H, Choi H, Kang H, et al. A systematic review of the smart energy conservation system: from smart homes to sustainable smart cities. Renew Sust Energ Rev 2021; 140: 110755.
11. Nti IK, Adekoya AF, Weyori BA, et al. Applications of artificial intelligence in engineering and manufacturing: a systematic review. J Intell Manuf. Epub ahead of print 15 April 2021. DOI: 10.1007/s10845-021-01771-6.
12. Zhang Q, Zhang X, Mu X, et al. Recyclable waste image recognition based on deep learning. Resour Conserv Recy 2021; 171: 105636.
13. Esmaelilian B, Wang B, Lewis K, et al. The future of waste management in smart and sustainable cities: a review and concept paper. Waste Manage 2018; 81: 177–195.
14. Hayward KJ and Maas MM. Artificial intelligence and crime: a primer for criminologists. Crime Media Cult 2021; 17(2): 209–233.
15. Ahmed AA and Echi M. Hawk-Eye: an AI-powered threat detector for intelligent surveillance cameras. IEEE Access 2021; 9: 63283–63293.
16. Milana C and Ashita A. Artificial intelligence techniques in finance and financial markets: a survey of the literature. Strat Change 2021; 30(3): 189–209.
17. Veloso M, Balch T, Borrajo D, et al. Artificial intelligence research in finance: discussion and examples. Oxford Rev Econ Pol 2021; 37(3): 564–584.
18. Syeda HB, Syed M, Sexton KW, et al. Role of machine learning techniques to tackle the COVID-19 crisis: systematic review. JMJIR Med Inform 2021; 9(1): e23811.
19. Alballa N and Al-Turaiki I. Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: a review. Inform Med Unlocked 2021; 24: 100564.
20. Li L, Rong S, Wang R, et al. Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: a review. Chem Eng J 2021; 405: 126673.
21. Saboe D, Ghasemi H, Gao MM, et al. Real-time monitoring and prediction of water quality parameters and algae concentrations using microbial potentiometric sensor signals and machine learning tools. Sci Total Environ 2021; 764: 142876.
22. Mehmood MZ, Ahmed M, Alzal O, et al. Internet of Things (IoT) and sensors technologies in smart agriculture: applications, opportunities, and current trends. In: Jatoi WN, Mubeen M, Ahmad A, et al. (eds) Building...
climate resilience in agriculture. Cham: Springer, 2022, pp.339–364.
23. Chaudhary KP, Dubey PK, Gahlot A, et al. Advances in sensors and measurements for metrological applications, 2021, https://www.researchgate.net/publication/353362864_Advances_in_Sensors_and_Measurements_for_Metrological_Applications
24. Ahmad K, Maabreh M, Ghaly M, et al. Developing future human-centered smart cities: critical analysis of smart city security, data management, and ethical challenges. Comput Sci Rev 2022; 43: 100452.
25. Lau BPL, Marakkalage SH, Zhou Y, et al. A survey of data fusion in smart city applications. Inform Fusion 2019; 52: 357–374.
26. Serban A, Poll E and Visser J. Adversarial examples on object recognition: a comprehensive survey. ACM Comput Surv 2021; 53(3): 66.
27. Tolpegin V, Truez S, Gursoy ME, et al. Data poisoning attacks against federated learning systems. In: Proceedings of the European symposium on research in computer security, Guildford, 14–18 September 2020, pp.480–501. Cham: Springer.
28. Zhang J, Chen B, Cheng X, et al. PoisonGAN: generative poisoning attacks against federated learning in edge computing systems. IEEE Internet Things 2021; 8(5): 3310–3322.
29. Russo A and Proutiere A. Poisoning attacks against data-driven control methods, 2021, https://arxiv.org/abs/2103.06199
30. Jiang W, Li H, Liu S, et al. Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles. IEEE T Veh Technol 2020; 69(4): 4439–4449.
31. Wadhwa S, Agrawal S, Chaudhari H, et al. Data poisoning against differentially private recommender systems. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, Xi'an, China, 25–30 July 2020, pp.1617–1620. New York: ACM.
32. Severi G, Meyer J, Coull S, et al. Explanation-guided backdoor poisoning attacks against malware classifiers. In: Proceedings of the 30th USENIX security symposium (USENIX Security 21), 11–13 August 2021. https://www.usenix.org/system/files/sec21-severi.pdf
33. Li D and Li Q. Adversarial deep ensemble: evasion attacks and defenses for malware detection. IEEE T Inf Foren Sec 2020; 15: 3886–3900.
34. Zhang F, Wang Y, Liu S, et al. Decision-based evasion attacks on tree ensemble classifiers. World Wide Web 2020; 23(5): 2957–2977.
35. Carminati M, Santini L, Polino M, et al. Evasion attacks against banking fraud detection systems. In: Proceedings of the 23rd international symposium on research in attacks, intrusions and defenses (RAID’2020), San Sebastian, USENIX Association, 14–18 October 2020, pp.285–300.
36. Takemura T, Yanai N and Fujiwara T. Model extraction attacks against recurrent neural networks, 2020, https://arxiv.org/abs/2002.00123
37. Wu B, Yang X, Pan S, et al. Model extraction attacks on graph neural networks: taxonomy and realization, 2020, https://arxiv.org/abs/2010.12751
38. Li M, Sun Y, Lu H, et al. Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. IEEE Internet Things 2019; 7(7): 6266–6278.
39. Huang Z, Pan M and Gong Y. Robust truth discovery against data poisoning in mobile crowdsensing. In: Proceedings of the 2019 IEEE global communications conference (GLOBECOM), Waikoloa, HI, 9–13 December 2019, pp.1–6. New York: IEEE.
40. Truong L, Jones C, Hutchinson B, et al. Systematic evaluation of backdoor data poisoning attacks on image classifiers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, Seattle, WA, 14–19 June 2020, pp.788–789. New York: IEEE.
41. Chan-Hon-Tong A. An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning. Mach Learn Knowl Extr 2019; 1(1): 192–204.
42. Wang J, Xu C, Guzmán F, et al. Putting words into the system’s mouth: a targeted attack on neural machine translation using monolingual data poisoning, 2021, https://arxiv.org/abs/2107.05243
43. Carlini N and Wagner D. Audio adversarial examples: targeted attacks on speech-to-text. In: Proceedings of the 2018 IEEE security and privacy workshops (SPW), San Francisco, CA, 20–24 May 2018, pp.1–7. New York: IEEE.
44. Miaò C, Li Q, Xiao H, et al. Towards data poisoning attacks in crowd sensing systems. In: Proceedings of the 18th ACM international symposium on mobile ad hoc networking and computing, Los Angeles, CA, 26–29 June 2018, pp.111–120. New York: ACM.
45. Fang M, Sun M, Li Q, et al. Data poisoning attacks and defenses to crowdsourcing systems. In: Proceedings of the web conference 2021, Ljubljana, 19–23 April 2021, pp.969–980. New York: ACM.
46. Zhang R and Zhu Q. A game-theoretic defense against data poisoning attacks in distributed support vector machines. In: Proceedings of the 2017 IEEE 56th annual conference on decision and control (CDC), Melbourne, VIC, Australia, 12–15 December 2017, pp.4582–4587. New York: IEEE.
47. Biggio B, Nelson B and Laskov P. Poisoning attacks against support vector machines, 2012, https://arxiv.org/abs/1206.6389
48. Dunn C, Moustafa N and Turnbull B. Robustness evaluations of sustainable machine learning models against data poisoning attacks in the Internet of Things. Sustainability 2020; 12(16): 6434.
49. Zhang H, Cheng N, Zhang Y, et al. Label flipping attacks against Naive Bayes on spam filtering systems. Appl Intell 2021; 51: 4503–4514.
50. Jędrzejski M, Oprea A, Biggio B, et al. Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: Proceedings of the 2018 IEEE symposium on security and privacy (SP), San Francisco, CA, 21–23 May 2018, pp.19–35. New York: IEEE.
51. Shanthini A, Vinodhini G, Chandrasekaran RM, et al. A taxonomy on impact of label noise and feature noise
using machine learning techniques. *Soft Comput* 2019; 23(18): 8597–8607.

52. Xiao H, Biggio B, Nelson B, et al. Support vector machines under adversarial label contamination. *Neurocomputing* 2015; 160: 53–62.

53. Taheri R, Javidan R, Shojafar M, et al. On defending against label flipping attacks on malware detection systems. *Neural Comput Appl* 2020; 32(18): 14781–14800.

54. Stisen A, Blunck H, Bhattacharya S, et al. Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition. In: *Proceedings of the 13th ACM conference on embedded networked sensor systems (SenSys’2015)*, Seoul, South Korea, 1–4 November 2015. New York: ACM.

55. Qolomany B, Maabreh M, Al-Fuqaha A, et al. Parameters optimization of deep learning models using particle swarm optimization. In: *Proceedings of the 2017 13th international wireless communications and mobile computing conference (IWCMC)*, Valencia, 26–30 June 2017, pp.1285–1290. New York: IEEE.