Investigating Machine Learning Attacks on Financial Time Series Models

Michael Gallagher, Nikolaos Pitropakis, Christos Chrysoulas, Pavlos Papadopoulos, Alexios Mylonas, Sokratis Katsikas

PII: S0167-4048(22)00325-X
DOI: https://doi.org/10.1016/j.cose.2022.102933
Reference: COSE 102933

To appear in: *Computers & Security*

Received date: 3 June 2022
Revised date: 16 September 2022
Accepted date: 23 September 2022

Please cite this article as: Michael Gallagher, Nikolaos Pitropakis, Christos Chrysoulas, Pavlos Papadopoulos, Alexios Mylonas, Sokratis Katsikas, Investigating Machine Learning Attacks on Financial Time Series Models, *Computers & Security* (2022), doi: https://doi.org/10.1016/j.cose.2022.102933

Investigating Machine Learning Attacks on Financial Time Series Models

Michael Gallagher, Nikolaos Pitropakis, Christos Chrysoulas, Pavlos Papadopoulos

Blockpass ID Lab, School of Computing, Edinburgh Napier University

Alexios Mylonas

Department of Computer Science, University of Hertfordshire

Sokratis Katsikas

Department of Information Security and Communication Technology, Norwegian University of Science and Technology

Abstract

Machine learning and Artificial Intelligence (AI) already support human decision-making and complement professional roles, and are expected in the future to be sufficiently trusted to make autonomous decisions. To trust AI systems with such tasks, a high degree of confidence in their behaviour is needed. However, such systems can make drastically different decisions if the input data is modified, in a way that would be imperceptible to humans. The field of Adversarial Machine Learning studies how this feature could be exploited by an attacker and the countermeasures to defend against them. This work examines the *Fast Gradient Signed Method (FGSM)* attack, a novel *Single Value* attack and the *Label Flip* attack on a trending architecture, namely a 1-Dimensional Convolutional Neural Network model used for time series classification. The results show that the architecture was susceptible to these attacks and that, in their face, the classifier accuracy was significantly impacted.

Keywords: Adversarial Machine Learning, Neural Networks, Financial Time-Series Models

1. Introduction

Machine Learning (ML) and AI have been groundbreaking in tackling various problems. They are used today in a variety of different application domains. AI can outperform human observers in identifying cancerous cells [1], even when using amateur equipment such as a mobile phone camera [2]. Autonomous vehicles are seen as the future of the automotive industry [3]. AI protects against fraud and cybercrime, from identifying unusual behaviour [4, 5] to protecting a personal device with facial recognition software [6]. It is expected to supplant many professional roles carried out today, with studies estimating that between 45% and 60% of jobs could be automated in the next 10 to 20 years [7].

These uses of AI extend into safety-critical applications and Internet-of-Things (IoT) devices. For example, poorly designed autonomous vehicles pose a significant danger, particularly as they become ubiquitous. However, it is hard to guarantee that the AI system will be well-behaved [8, 9]. The decision-making in the AI system can be extremely complex, making it difficult to predict how the system will act in all situations.

The increasing popularity of AI and Machine Learning attracted the attention of malicious parties who started launching attacks against them. The common goal of all the attacks is to impact the performance in some way. Most of these attacks would not fool a human, but many would not even be perceptible by a human observer, yet they may substantially impact AI performance.

Attacks can be carried out against real-world AI applications. For example, unique sunglasses can be designed that trick a facial recognition network into granting access to a secure system [10]. For autonomous vehicles, traffic signs could be covered in graffiti which tricks their AI into misreading the sign [11].

Attacks may occur in all phases of the AI lifecycle. They can be executed when an AI system is being trained, by poisoning the training data [12], or when it is used in production by evading classification. Attacks have been published in white-box, grey-box and black-box scenarios. Worryingly, the attacks show transferability across networks. This means that an attack designed for one system could impact another [13]. Prediction with financial time series models is one of the application domains that have been poorly investigated in the literature concerning the influence of such attacks. Besides being a major influential factor in the global economy, financial time series models are also known for their non-linear, non-stationary and noisy nature; hence, making challenging the effort to capture their trends accurately.

This work sheds light on this area, proving that attacks are feasible and that their impact should be taken into consideration by the security and AI communities. The contributions of our work can be summarised as follows:

1. We mount label flipping attacks, poisoning a 1-Dimensional Convolutional Neural Network model that makes predictions based on financial stock data, in order to generate adversarial examples using the FGSM against

the same model.

2. We create and launch Single Value attacks, a novel adaptation of FGSM based on one-pixel attacks [14] aiming to identify the most impactful entry for perturbation.
3. Finally, we analyse and critically evaluate the experimental results along with the model's robustness against the aforementioned attacks.

The rest of the paper is organised as follows. Section 2 builds the background around attacks against machine learning-based systems while reviewing the related literature. Section 3 briefly explains the methodology we followed to perform the study. Section 4 presents the results of our experiments, while Section 5 discusses the feasibility of the suggested scenarios and proposes countermeasures drawn from the known literature. Finally, Section 6 draws the conclusion and gives some pointers for future work.

2. Literature Review and Background Knowledge

2.1. Attacks Against Machine Learning

2.1.1. Background

Kearns et al. [15] first referenced the challenges of automated learning systems where training data is controlled by an adversary and proved the bounds for malicious errors in training data. The first practical applications appeared in the mid-2000s and were primarily evading spam filters, and anti-malware processes [16]. The attacks on spam filters typically involved obfuscating the detected words by adding trusted words. This type of attack was called *evasion*, as the attack would attempt to evade a trained classifier.

Countermeasures were also studied. [17] investigated making models more robust by not giving too much weight to one single feature. Their method was effective for spam filtering and handwriting recognition.

Poisoning attacks were also researched. Newsome et al. [18], studied an attack against malware classifiers, where the adversary generated labelled samples, which would prevent the training of an accurate classifier. In the scope of intrusion detection systems, Rubinstein et al. [19], studied multiple poisoning schemes and found that creating a moderate amount of poisoned traffic would substantially increase the chances of evading detection [19].

Barreno et al. published the first taxonomy of machine learning attacks [20]. They modelled attacks by their influence, specificity and type of security violation. These terms were expanded in subsequent work, by including a comprehensive set of scenarios where each attack type could be used [21]. Huang et al. also published a taxonomy, building on existing work and expanding their taxonomy into attacks on ML techniques [22]. They discussed attacks on privacy-preserving ML architectures. These architectures are designed to obfuscate the data used to train a classifier. Thus, this is particularly important in some private data, such as medical data, since they discussed theoretical attacks which could break several privacy-preserving properties.

Recent works on machine learning attacks were catalysed by the discovery of *adversarial examples* [23]. Szegedy et al. [24], found that when testing an image classifier, tiny alterations to an image that might be imperceptible to humans could cause a dramatic misclassification. These findings highlighted a significant threat for deep learning architectures, at a time when they were seeing significant breakthroughs in performance [25]. Since [24] was published, the field of adversarial ML has grown considerably to tackle these problems, which could inhibit the uptake of AI.

2.1.2. Recent Works

Pitropakis et al. performed a taxonomy and survey of the literature and provided a language for categorising attack knowledge, style and intention while describing three categorisations for the knowledge of the attacker, namely Black-box, Grey-box and White-box [13]. Black-box attacks assume no knowledge, Grey-box assumes some knowledge, and White-box assumes total knowledge and unrestricted interactions. The attacks were categorised as *Poisoning* and *Evasion*. *Poisoning* attacks are a subset of the *causative* attacks defined by [20]. They target the manipulation of input data to corrupt a network, often by tampering with the training data. *Evasion* attacks aim to achieve incorrect classifications for data in the testing stage. These are a subset of *exploratory* attacks. They often involve generating a malicious input which is incorrectly classified. One example is the adversarial examples discovered by [24]. For illustration, consider the example of an attack against a facial recognition security system. A poisoning attack may seek to train the network against a modified or mislabelled image. The network would then incorrectly classify an unaltered image. An evasion attack may trick the network by changing the input in a particular way; as such, [10] crafted glasses frames to impersonate celebrities. Yuan et al. performed a taxonomy for Deep Neural Networks (DNN) [14], whilst Gu et al. [11], provided an excellent illustration of some real-world attacks against autonomous vehicles which could hinder trust in the DNN technology. They evaluated a DNN, which was designed for an autonomous vehicle. They found that placing a yellow square at a particular location on a stop sign would cause it to be misclassified as a speed limit sign. This misclassification would not happen with a human observer, but could cause an autonomous vehicle to drive dangerously.

Sadeghi et al., in [26], published a taxonomy for various aspects of Adversarial ML (AML) research, including the dataset, the ML architecture and the defence response. They also described an *AML cycle* i.e., a system for representing the arms race between ML applications and adversaries.

Interestingly, 98% of Adversarial Machine Learning papers using deep learning architectures involve image or text classification, while time series datasets are significantly ignored, despite their utilisation in mission-critical applications in health care and financial trading [26].

Academic research focused on machine learning has revealed certain network architectures as being optimal for some problems. Convolutional Neural Networks (CNNs) have shown incredible success in computer vision tasks [14] and facial recognition [27]. Convolutional Neural Networks work by taking input data and performing convolutions on the data. For image classification problems, this involves creating feature maps of clusters of pixels.

Although Recurrent Neural Networks and Long Short-Term Memory Networks have traditionally been the recommended ML architecture for time series classification, it has recently emerged that CNNs can outperform these architectures for this problem [28, 29]. Chen et al. applied transformations to a time series, such as moving averages, and then combined these together [28]. This approach allowed a 2-Dimensional CNN to be used as there were effectively multiple time series being analysed simultaneously. Fawaz et al. used a 1-Dimensional CNN by training their data on the raw time series [30]. In this approach, the CNN concept of *kernel size* is still applicable. However, the kernel window only includes adjacent entries in the time series across one dimension.

The FGSM produced tiny perturbations imperceptible to humans to a test image which was subsequently misclassified, was proposed in [23]. As in [24], it was found that the adversarial examples transferred to different models. Since its publication, the FGSM method has seen a lot of research attention. Simple variations, such as adding random perturbations, add robustness to countermeasures [14]. Multi-step methods are more powerful variations, which use projected gradient descent for the negative loss function instead of the sign of the gradient [31, 32]. These have proven to be extremely robust against defences [33].

Causative attack strategies target the classifier itself by manipulating the parameters, the architecture or the training set. Sadeghi et al. [26], claimed that data poisoning is the most common causative attack and defined “Label Flip” as poisoning a dataset by changing the hard class labels. Xiao et al. [34], attacked a Support Vector Machine model with targeted Label Flip, where they changed the labels to an alternative class. Their method required a classifier trained with the uncontaminated dataset. Then, a new model was trained on this manipulated data. This new model was found to be very underperforming.

2.2. *Financial Time Series Prediction*

Time series analysis problems are a common class of problems in domains such as finance, weather, health care and security [28].

Selvin et al. used several neural network models to analyse a financial time series [35]. Each model was given a sliding window of the time series as input. They used minute-wise stock data with a 90 minute sliding window and trained their models to predict 10 minutes into the future. They trained a Recurrent Neural Network (RNN), a Long Short-Term Memory (LSTM) network and a CNN. RNN and LSTM networks are traditionally popular architectures for deep learning with a time series. In an RNN, computational units form a directed circular graph which use internal memory when processing inputs. This is achieved with a recurrent feedback loop. An LSTM is a form of RNN with special cells which allow it to store memory for a longer period of time. The authors found that the RNN and LTSM models were incapable of capturing dynamic trends in the price movement, with the CNN being much more accurate. They hypothesised that this was a result of the dynamic nature of the stock market. As price movements happen for reasons independent of the recent price history, the recent history offered poor predictive value. The CNN model does not use the recent history for any particular sliding window, and was therefore not impacted by this.

Chen et al. [28], used CNNs for financial time series forecasting. They found that CNNs could understand complex patterns with more accuracy than rule based systems. Their work was specific to a 2-Dimensional convolutional network. They applied transformations to the time series to get a 2-Dimensional output. For example, several moving averages were derived from the time series. These were combined together, which resulted in a 2-Dimensional output. Their experiment utilised a sliding window approach, similar to [35].

2.2.1. *Time Series Attacks*

Even though adversarial attacks on 2-Dimensional problems such as image recognition have received a lot of attention in recent years, such attacks against a 1-dimensional time series have not being thoroughly studied.

Fawaz et al. published the first study on adversarial examples against deep learning architectures used for time series classification [30]. They attacked a state of the art deep learning architecture across spectroscopic time series used for food safety, electrical sensor readings from vehicles and a time series of electricity consumption. As in this work, they successfully utilised the FGSM method to produce adversarial examples. However, they had to add noise to investigate how the model would behave, and they did not investigate the financial time series model which, in comparison to their chosen time series models, has larger amounts of noise.

Karim et al. used a fully connected CNN and demonstrated attacks across 42 datasets. They experimented with white-box and black-box attacks. The black-box attacks featured a number of restrictions, such as no access to the dataset labels during an attack. They used a *Gradient Adversarial Transformation Network* model to generate their adversarial examples, as proposed in [36], and used an unsupervised neural network. They found that all datasets were susceptible to attack [37].

Our work differentiates from others in the literature since it contributes an evaluation of several attacks against a 1-Dimensional CNN architecture when used for time series classification. This work also contributes to the 2% of AML research against a deep learning architecture which does not use image or text datasets [26]. Compared to the existing literature, we experimented with: i) A novel Single Value attack; ii) A Label Flip attack; and iii) An FGSM attack using a financial time series model. As there is limited research into using deep learning for financial time series analytics [28], to the best of our knowledge, our work contributes to this research area and demonstrates how financial groups may be vulnerable to a range of attack vectors. It also features a stock trading simulation to assess the financial impact.

3. Methodology

This section describes a 1-Dimensional Convolutional Neural Network model and three attack methodologies, namely FGSM, Single Value and Label Flip.

Each experiment represented a real-world security risk in the financial trading domain. The FGSM and Single Value were evasion attacks, attempting to force a misclassification at testing time. The attacks were performed by intentionally altering the price of a financial instrument across the time series.

The Label Flip attack demonstrated how a poisoned dataset could create a disproportionately inaccurate model. In a real-world scenario, an attacker with access to the training data could poison a small amount of the data and drastically affect performance.

3.1. Model Under Attack

Financial stock data was obtained from [38], and daily stock price data for Google stock was used. The time range was 2006–2018. The training period was 2006–2014 and the remaining data was used for validation. The data was processed with a sliding window. The sliding window size was fixed to 30 days, and the future price prediction offset was fixed to 14 days. These values are chosen to investigate whether a month of data could predict the price in two weeks.

The data was normalised so that each entry in the time series was the price difference from the previous entry. This is known as the price *delta* or Δ. By examining relative price movements, the model would identify certain patterns used for prediction. This is known in financial analytics as *technical analysis* [39].

The Convolutional Neural Network that was used is composed of two convolution layers and two hidden layers. It was trained for 2000 iterations using the PyTorch library. The pseudocode representing the model and its parameters can be seen in Algorithm 1.

Algorithm 1 Pseudocode of 1-D CNN Model

```
Conv1d(inputs=1, outputs=32, kernel_size=3, padding=1, stride=1)
ReLU(inplace=True)
Conv1d(inputs=32, outputs=3, kernel_size=3, padding=1, stride=1)
ReLU(inplace=True)
Linear(84,250)
ReLU(inplace=True)
Linear(250,2)
```

The network output was a vector with two entries. This form of output is known as a *one-hot vector*. The two entries represented the predicted probability of a sell or a buy. The sigmoid function was used on the vector in the final stage of the network to normalise the entries between 0 and 1. The Stochastic Gradient Descent (SGD) optimiser was used for backpropagation. Binary Cross-Entropy (BCE) Loss was used as the loss function. Cross-Entropy Loss functions are common loss functions when evaluating one-hot vector results [37].

3.1.1. Optimisation

The CNN had two convolution layers, each with its own output and kernel sizes. These were hyperparameters to be optimised. For computational performance reasons, a limit was placed on these parameters. A range of 1 to 200 was chosen for tuning the number of convolution layers. A range of 1 to 20 was the constraint for the kernel size. This resulted in 8 million possible hyperparameter combinations.

The Stochastic Gradient Descent algorithm was parameterised with a learning rate and momentum. The learning rate affects how much the gradients are changed during backpropagation. The momentum accelerates the gradient change and leads to faster converging. Both of these parameters were considered hyperparameters. The range for optimisation was between $1e-9$ and $1e-1$ for both parameters.

The Optuna optimisation framework was used for efficient hyperparameter optimisation [40]. The framework allowed the hyperparameter search space to be dynamically generated in the program at runtime. It then performed sampling to find the optimal parameters given the loss function. It used a Tree-structured Parzen Estimator during the search. This form of Bayesian Optimisation uses a probability model to determine which hyperparameters should be evaluated.

The optimisation framework used BCE Loss as the loss function to mirror the model's loss function. The optimisation ran for 150 rounds, and the accuracy in each round is illustrated in Figure 1.

The optimal parameters for the first convolutional layer were a 32-dimensional output with a kernel size of 3. The optimal parameters for the second layer were a 3-dimensional output with a kernel size of 3. The learning rate was $1.71176e-05$. The momentum was 0.081.

3.2. FGSM Attack

The FGSM [23] is very similar to the methodology used in this experiment. However, it was adapted slightly to function with a time series as an input.

The experiment used validation data from the dataset. This meant that the model was attacked with data which was not used for training. As discussed, the data was processed with a sliding window. This attack created a small perturbation to the original test data. The perturbation created for each sliding window can be seen in Algorithm 2.

Algorithm 2 Pseudocode of Perturbation

```
Evaluate the sliding window with the model.
For each entry in the time series, capture the gradient using backpropagation.
For each gradient, obtain the sign of the gradient as $-1$ or 1.
Multiply the gradients by the parameter $\epsilon$.
```

Formally, the perturbation can be expressed as: $\eta = \epsilon \cdot sign(\nabla_x J(X))$ where X is the model input and $\nabla_x J(X)$ is the gradient. This perturbation was added to the original input data, as in [23] and [30]. This is illustrated in Figure 2.

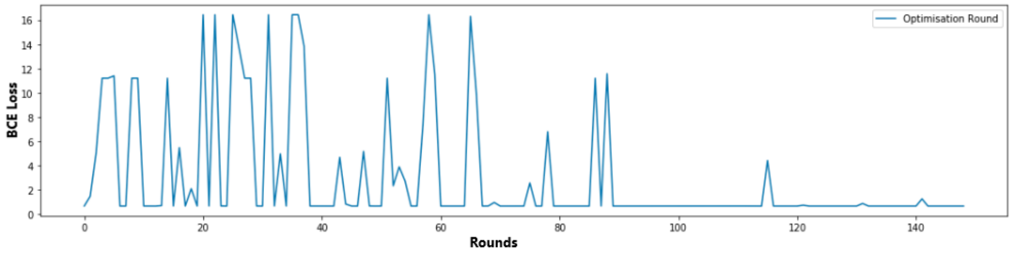


Figure 1: Hyperparameter Optimisation Accuracy Results

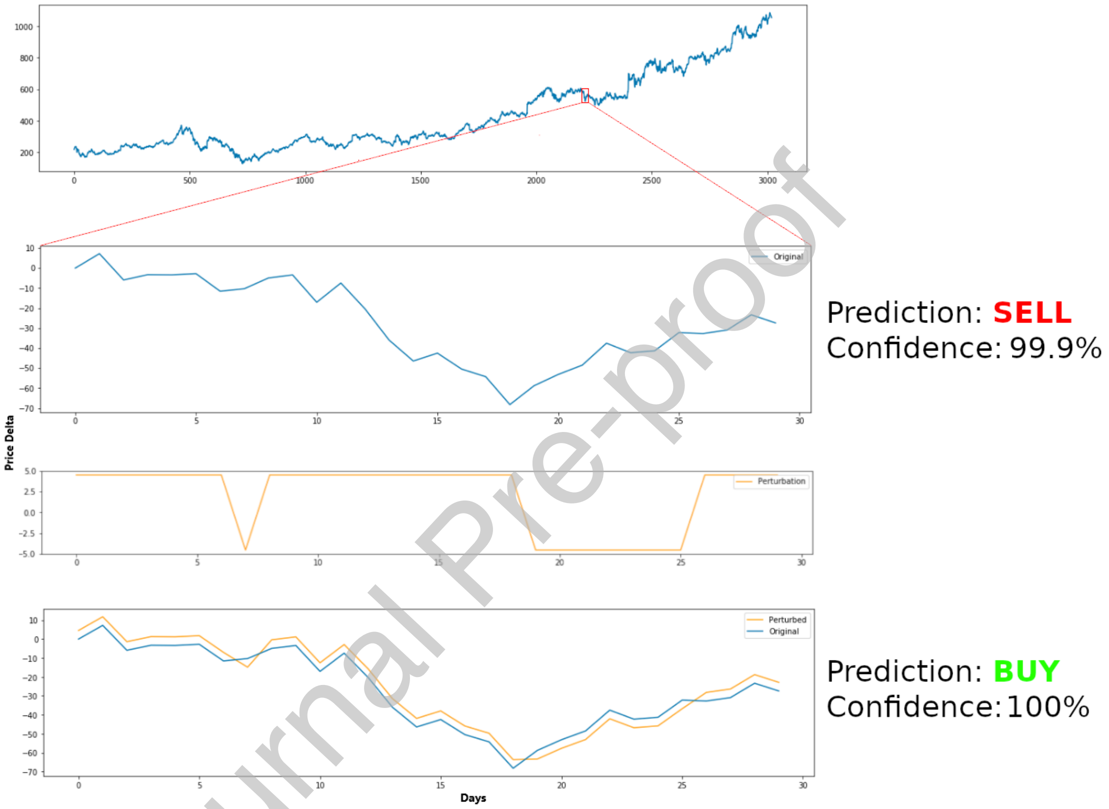


Figure 2: Sliding Window input and FGSM attack

The parameter ϵ drives the magnitude of the attack. In this scenario, it represented how much the price of the stock would be moved to perform the attack. It was important not to set ϵ too high, as this could be identified as an abnormal price movement by human observers.

For these reasons, ϵ was limited to the mean delta across the dataset. The value was calculated as 4.54. This means that the perturbation would modify the time series by the average amount of daily price change. The experiment was repeated for different values of ϵ up to this limit. The initial value for ϵ was 0, where no data was perturbed. In each subsequent iteration, ϵ increased by 10% of the mean delta. The experiment concluded after ϵ was equal to the mean delta.

3.3. Single Value Attack

The Single Value attack is a novel adaptation of the FGSM attack. It attempted to identify the most impactful entry in each sliding window to perturb. It was inspired by the one-pixel attack [14], which caused poor performance by perturbing a single pixel in an image. In our case, each sliding window was passed through the model, and the gradients were retrieved using backpropagation. In the FGSM attack, a single perturbation was created, which perturbed all entries in the sliding window. The Single Value attack methodology differed can be seen in Algorithm 3. This was done for all sliding windows. An example of a chosen perturbation is illustrated in Figure 3.

The value for ϵ was two times the standard deviation of the delta between each entry. This is represented as $\epsilon = 2\Delta$. This value was chosen as it would have more impact than using the

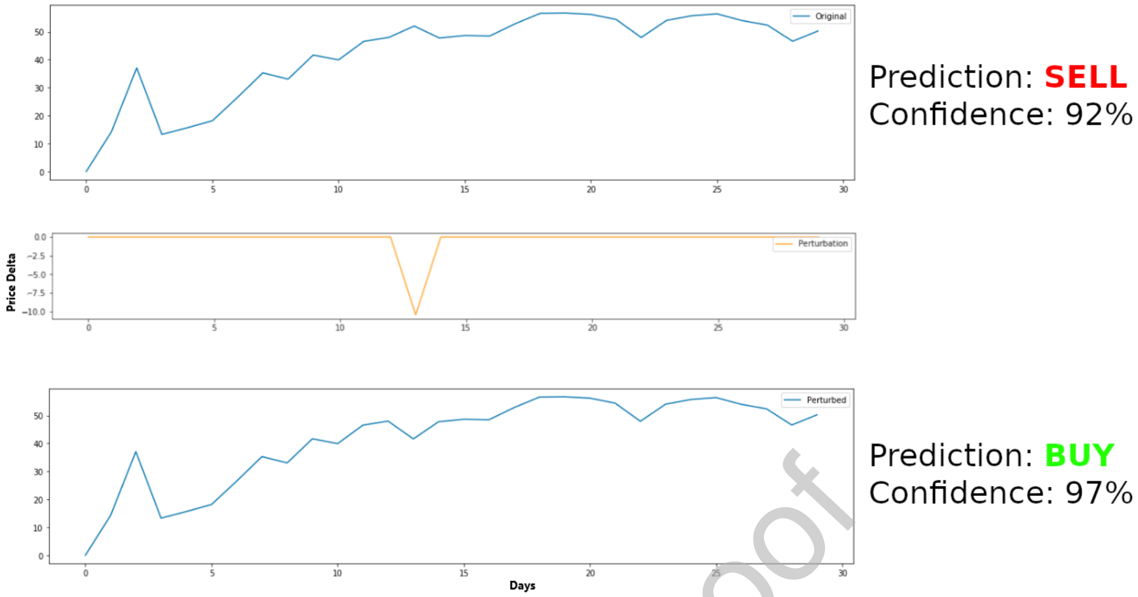


Figure 3: Single Value attack example

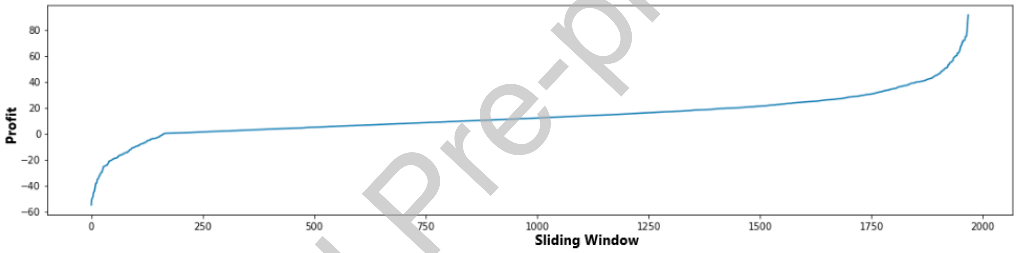


Figure 4: Sliding Windows sorted by profitability

Algorithm 3 Pseudocode of Single Value Perturbations

```
1: For each sliding window, $n$ number of perturbations were temporarily created, where $n$ is the number of entries in the time series. Each of these perturbations affected just a single entry in the sliding window, such that all entries had a corresponding perturbation.
2: Each of the perturbations were applied to the original data.
3: The perturbed data was passed through the model and a loss was obtained.
4: The worst performing perturbation was chosen as the perturbation to use in the attack.
```

mean, but would still be viewed as a normal price movement by human observers. The value was calculated to be 10.4.

3.4. Label Flip Attack

The Label Flip attack involved changing the label of training data so that an inaccurate classifier was trained. The objective was to change a small percentage of the training data and have a relatively large impact on performance. In this experiment, flipping a label meant changing a *Buy* label to a *Sell* and vice versa [34], and this can be seen in Algorithm 4.

This method produced a model which had maximal loss under the original classifier, but minimal loss with the poisoned classifier. This is because it was trained to identify the most costly trades as profitable. This experiment used profitability as the loss function to identify the worst performers. This function was chosen in order to demonstrate the impact in a real-world scenario. For computational performance reasons, the models for the clean and poisoned datasets were trained for 2000 iterations.

The methodology was parameterised by n. This was the percentage of data which would be flipped in the experiment, and a value of $n = 10$ was used. This was chosen by analysing

Algorithm 4 Pseudocode of Label Flip

```
Use a model trained on a clean dataset.
Using a trading simulation, calculate a profit for each sliding window in the training dataset.
Sort the sliding windows by the least profitable.
Extract the worst performing $n$ sliding windows, where $n$ is a parameter representing the percentage of data to be flipped.
Flip the label of the worst performing data.
Re-combine the original and flipped datasets.
Train a new model using this poisoned dataset.
```

the distribution of poorly performing datasets. It was found that some 10% of the sliding windows are disproportionately unprofitable. This is visualised in Figure 4, which shows the profitability for each sliding, sorted by the least profitable first. It can be observed that there is a small percentage of sliding windows which underperform.

4. Results

4.1. FGSM attack

The FGSM attack caused a significant reduction in performance across several metrics, and the BCE loss function was used to compute the loss since this was suitable for a binary classifier. The experiment was repeated over several iterations with the value of ϵ increasing in increments. The minimum value for ϵ was zero, and the maximum was equal to the mean delta in the time series. The increments were in 10% of the mean delta.

When $\epsilon = 0$, the loss was equal to the base loss of the network when using the evaluation data. This was 4.95. As ϵ increased, the loss increased. The final loss for each increment of ϵ is shown in Figure 5.

When ϵ was equal to the mean delta, 4.55, the loss was 18.72. An example of the experiment results for each sliding window in this iteration is shown in Figure 6. This illustrates that the loss was significantly higher when using the FGSM attack.

The classifier used in this experiment was binary, using the labels *Buy* and *Sell*. A simple performance metric for this classifier is calculating its *Accuracy*. Accuracy is the ratio between the correct predictions and the total predictions. The output of a binary classifier can belong to four classes: True Positives, True Negatives, False Positives, and False Negatives [41]. Using these classes, the formula for accuracy can be derived as: Accuracy= $\frac{tp+tn}{tp+tn+fp+fn}$

In this experiment, *Positives* were *Buy* labels and *Negatives* were *Sell* labels. Using these classes, the values of other common performance metrics can be calculated.

Recall is accuracy in the Positive class. It is an indication of how performant the classifier was at predicting Positive class instances. Recall = $\frac{tp}{tp+fn}$

Precision is the ratio of correctly classified Positive instances, to all instances classified as Positive. It is useful for measuring the level of misclassification as Positive. Precision = $\frac{tp}{tp+fp}$

F-Score is a common metric which is defined as the harmonic mean of Recall and Precision. FScore = 2 × $\frac{precision \times recall}{precision+recall}$

Four accuracy metrics were used for the attacks in this experiment: Standard Accuracy, Recall, Precision and F-Score. The measurements for the FGSM attack showed a dramatic decrease in accuracy across all metrics. These are shown in Figure 7a.

Accuracy and F-Score are among the most popular metrics for measuring binary classification performance. However, [41] found that these metrics display overly optimistic and inflated results due to the imbalance issues. The Matthews Correlation Coefficient (MCC) overcomes this class imbalance issue. The MCC is a special case of the ϕ (phi) coefficient, which is used for binary classification problems. It is computed as MCC = $\frac{(tp \times tn)-(fp \times fn)}{\sqrt{(tp+fp)\times(tp+fn)\times(tn+fp)\times(tn+fn)}}$ and it is claimed to be the only binary classification measurement which generates a high score if a majority of positive and negative instances are correctly classified [41]. The values of MCC for each iteration of the FGSM attack are illustrated in Figure 7b.

The real-world impact was demonstrated in a stock trading simulation of bought or sold stock based on predictions from the model. The accumulated profit when evaluating the test data was recorded for each increment of ϵ, in increments of 10% of the mean delta. The results are depicted in Figure 8. The final loss curve is shown in Figure 9. This shows the final financial profit from the simulation for each increment of ϵ.

4.2. Single Value Attack

A significant reduction in accuracy was observed for the Single Value attack. This was significant as only a single entry in each sliding window was perturbed. This attack was also parameterised with ϵ, representing the amount of price change in the attack. As only one entry was being perturbed, a higher limit was chosen for the value of ϵ. This value was still constrained, so the price movement would be within historical ranges. A value of two times the standard deviation of the delta of the time series was chosen, and this was 10.42 in the dataset.

The loss for each increment of ϵ is shown in Figure 10.

Figure 11 shows the loss for each sliding window when ϵ was at the highest value.

The accuracy measurements were calculated in the same manner as with the FGSM attack. The accuracy, precision, recall and F-Score measurements were calculated for each iteration of ϵ. The results are illustrated in Figure 12a.

The MCC value was calculated, and a significant reduction in accuracy was observed as ϵ was increased. The magnitude of the decrease in the MCC value was half the value seen in the FGSM attack. This is an impressive reduction in accuracy as only one-thirtieth of the entries were perturbed. The result for each iteration is shown in Figure 12b.

The profitability was calculated with the same trading simulations as the FGSM results. However, a large impact on profitability was observed, approximately half the impact of the FGSM results. This correlates with the results seen in the MCC calculation. The final monetary loss for each increment of ϵ is

ϵ (% of mean Δ)	Loss
0 (0 %)	4.95
0.91 (20 %)	8.00
1.82 (40 %)	11.21
2.73 (60 %)	14.14
3.64 (80 %)	16.77
4.55 (100 %)	18.72

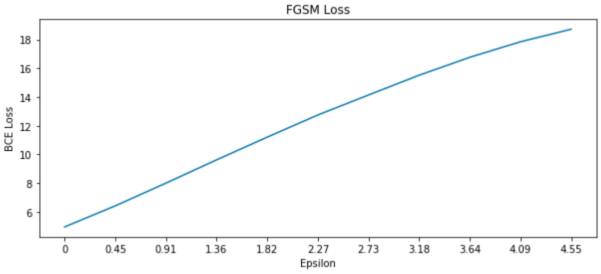


Figure 5: BCE Loss for FGSM attack for each increment of ϵ

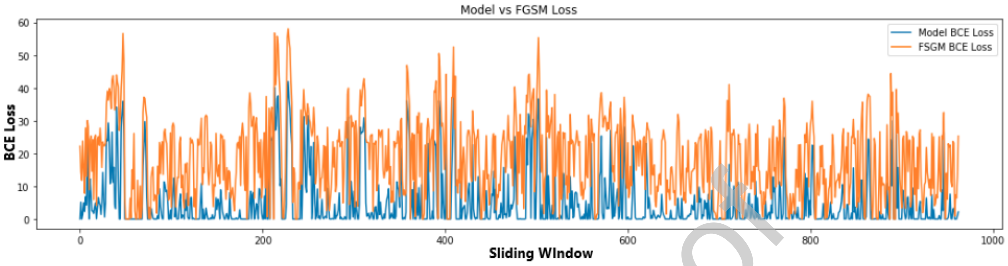
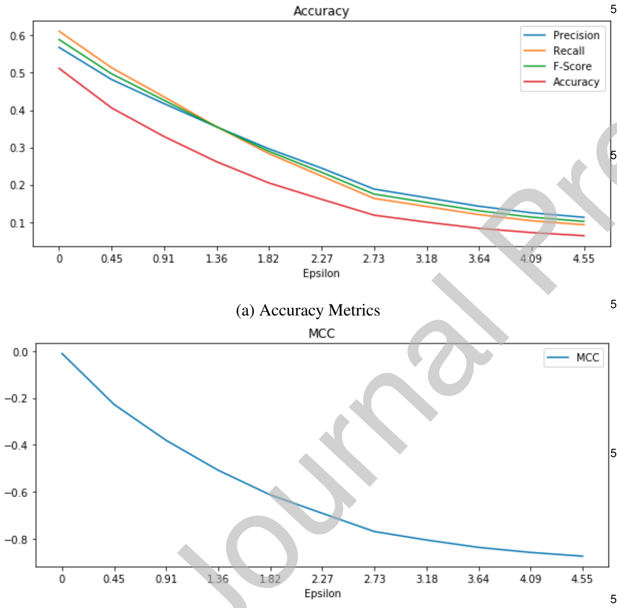


Figure 6: BCE loss for $\epsilon = mean\Delta$ under FGSM attack



(a) Accuracy Metrics

(b) FGSM MCC

Figure 7: Accuracy Metrics & FGSM MCC

shown in Figure 13. The result from the trading simulation for each increment of ϵ is shown in Figure 14.

4.3. Label Flip

The Label Flip attack involved training a classifier with a poisoned dataset. The experiment flipped the label of 10% of the dataset. For evaluation, metrics for a clean dataset and the poisoned dataset were obtained. The attack had a clear impact on profitability in the trading simulation and caused a significant increase in the BCE Loss. Minor decreases were found in accuracy and MCC score.

The BCE loss for the clean dataset and poisoned dataset were calculated. The loss for the clean dataset was 2.65. This was different to the base loss for the evasion experiments as the model was retrained and initialised with random weights and biases. Using a poisoned dataset increased the loss to 5.8, demonstrating a substantially reduced performance. This is illustrated in Figure 16.

The simulation showed a drastic impact on performance. The profitability during the trading simulation is shown in Figure 15. The trading simulation showed a reduction in final profit of \$2748 for the base model to −\$915 for the poisoned model. This is significant as only 10% of the labels were flipped. Furthermore, this impact was achieved without manipulating the price of the market. These would make the Label Flip attacks much more cost-effective than the evasion attacks.

Further, several accuracy metrics were obtained. These are shown in Figure 17a. A minor decrease in accuracy was observed between the base and poisoned classifiers. This is in contrast to the substantially decreased performance in profitability and BCE Loss. It is interesting that the results are not fully correlated. This may be because profitability and loss metrics incorporate the magnitude of classification and misclassification. They add more weight to *very* costly or *very* profitable predictions. This is in contrast to the accuracy metrics, which are simply concerned with the *correctness* of a prediction.

This effect may be encouraged by the methodology. The methodology flipped the label of the most unprofitable sliding windows instead of the least accurate. This should create a classifier where disproportionality underperforms in profitability metrics. Finally, it is worth noting that the Label Flip attack may perform strongly across all metrics in a highly predictive model. However, producing a model that accurately predicts financial time series data is very challenging. The MCC measurement showed a slight decrease in performance between the base model and the poisoned model. This may be for the same

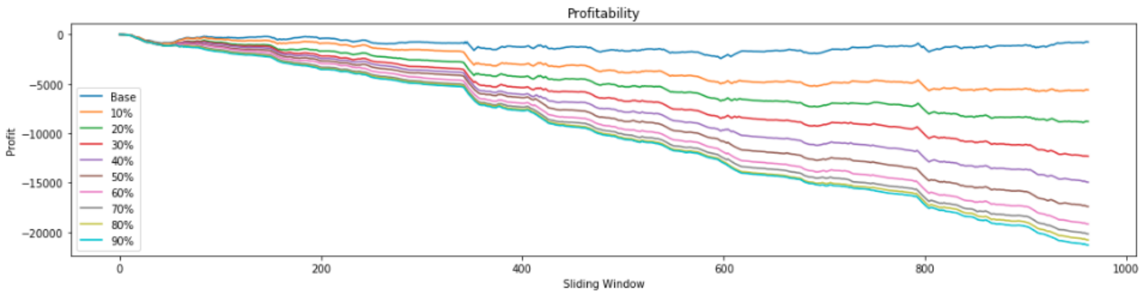


Figure 8: FGSM simulated profit for $\epsilon = mean\Delta$

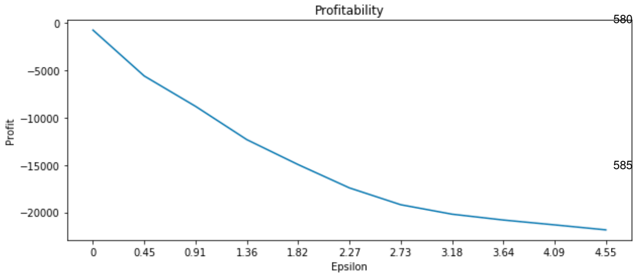


Figure 9: FGSM Total Profit per iteration of ϵ

reason as the other accuracy metrics. The MCC is shown in Figure 17b.

5. Discussion

The results demonstrated a significant performance loss in each attack and how this can lead to financial loss when used in financial trading simulations. As these are white-box attacks, an attacker would need access to the internals of the neural network to perform them. However, the existence of insider threats, malicious employees, network breaches and data theft mean that this is a legitimate concern.

It is worth noting that a more finely tuned model could be even more affected by attacks. A side experiment was performed, where the FGSM method was applied to the training data, instead of the validation data. Naturally, the model would be much more predictive of this data. The initial results showed that the losses were more than double than those of the original model. This is illustrated in Figure 18.

[26] claimed that proactive defences were the most popular approach. They found that most approaches aimed to prevent damage as much as possible. Reactive defence approaches provide protection for a trained neural network, and [14] identified several common approaches in the literature. Adding a *specialised detector* involved including an attack-detector as part of the network. This detector would identify adversarial examples and block them before classification. For example, [42] studied the statistical properties of adversarial examples in order to detect attacks. They demonstrated that many attacks, including the FGSM method, could be detected. For defences which modify the classifier, [26] and [14] identified several approaches. Adversarial example thwarting involved neutralising perturbations in adversarial examples. They found several techniques to achieve this, such as data transformation [43] and noise filtering [44]. Training process modification involves modifying the training data to make the classifier more resilient to adversarial examples. A common approach was incorporating adversarial examples into the training dataset. This approach was tested by [23] when studying the FGSM method. They showed that incorporating adversarial examples into the training set improved the robustness of the classifier. However, [33] found that this approach would add robustness against *one-step* attacks but would not help with *iterative* attacks. ML algorithm modification involves modifying the classifier to draw more accurate class boundaries, such as applying non-linear ML algorithms [45]. Network distillation involved reducing the complexity of the neural network. The technique was originally used to reduce the size of the network by transferring knowledge from a large to a small network. They found that attacks that relied on networks' sensitivity were less successful. However, the improvements were quite modest. For example, the success rate of an attack from [46] against the popular MNIST and CIFAR-10 datasets was reduced by 0.5% and 5%, respectively. Adversarial detecting involves identifying adversarial examples, often using ML architectures. It differs from the proactive specialised detector defence as the system for detecting attacks exists outside of the neural network itself [26].

6. Conclusions

Adversarial examples continue to threaten ML and AI systems. This work explored *FGSM*, a novel *Single Value* and *Label Flip* attacks against a 1-Dimensional Convolutional Neural Network. This work focused on the potential impact to the financial trading domain. A trading simulation was used to assess the impact of the attacks, and we found that all attacks caused a significant reduction in profitability.

The attacks in the experiment demonstrated that the target architecture was susceptible to adversarial examples. Further potential problems arise as the stock market is publicly traded. If the price was modified by buying or selling a stock, other investors could return the stock price to an unwanted value.

Our experiment was performed using twelve years of daily stock price movements for Google stock. Our future plans include the use of much more granular data while considering 2-Dimensional CNN models. Additionally, there are many

ϵ (% of $2\sigma\Delta$)	Loss
0 (0 %)	4.95
2.08 (20 %)	5.62
4.17 (40 %)	6.29
6.25 (60 %)	7.05
8.34 (80 %)	7.82
10.42 (100 %)	8.59

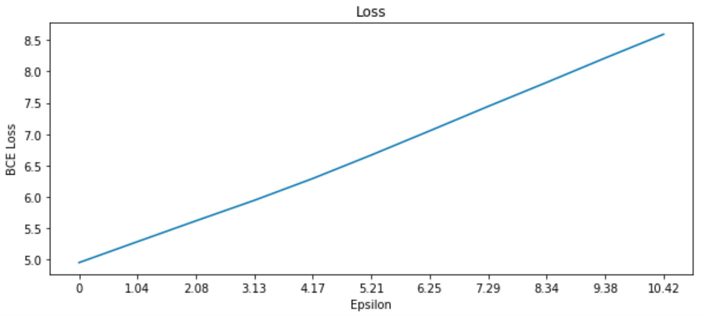


Figure 10: BCE Loss for Single Value attack for each increment of ϵ

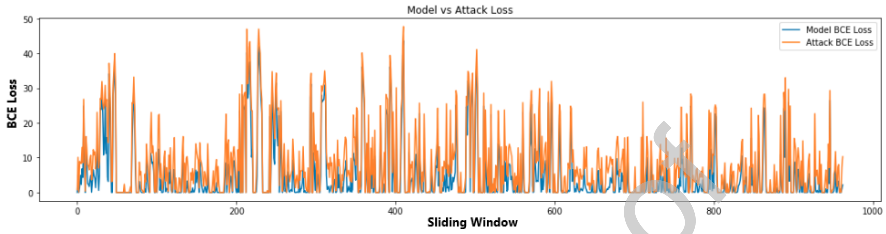
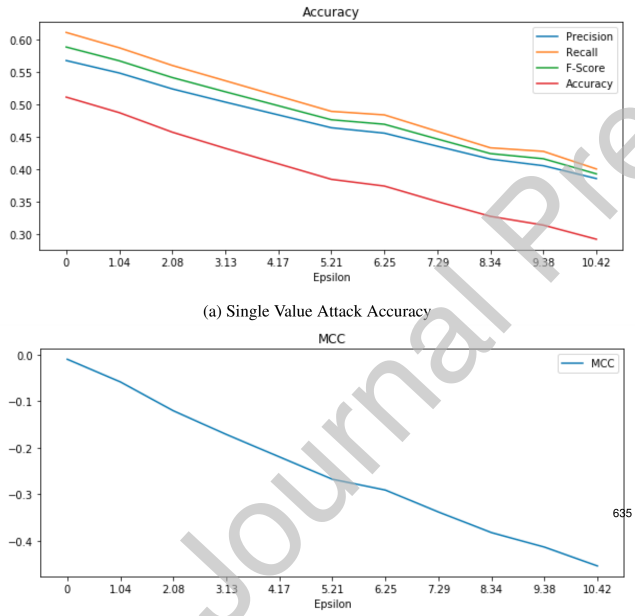


Figure 11: BCE loss for $\epsilon = 2\omega\Delta$ under Single Value attack



(a) Single Value Attack Accuracy

(b) Single Value Attack MCC

Figure 12: Single Value Attacks for each increment of ϵ

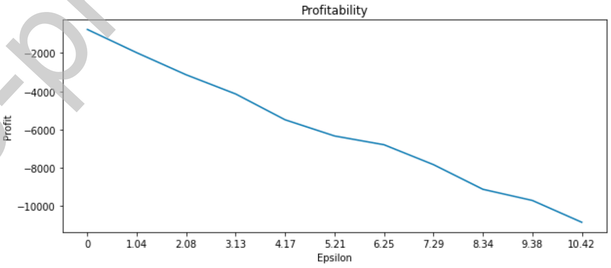


Figure 13: Single Value Total Profit per iteration of ϵ

other grey-box and black-box attacks described in the literature which could be effective against a 1-Dimensional CNN.

In our future work, we plan to study the effects of such attacks against different Deep Neural Network architectures. This would be interesting from an adversarial perspective as the adversaries would have to produce adversarial examples across all the channels. Additionally, we aim to explore defensive countermeasures further, such as adversarial training techniques [47–49]and calculate the impact of these attacks again.

References

[1] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, et al., Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer, Jama 318 (22) (2017) 2199–2210.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, nature 542 (7639) (2017) 115–118.

[3] H. J. Vishnukumar, B. Butting, C. Müller, E. Sax, Machine learning and deep neural network—artificial intelligence core for lab and real-world test and validation for adas and autonomous vehicles: Ai for efficient and quality test and validation, in: 2017 Intelligent Systems Conference (IntelliSys), IEEE, 2017, pp. 714–721.

[4] N. Dhieb, H. Ghazzai, H. Besbes, Y. Massoud, A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement, IEEE Access 8 (2020) 58546–58558.

[5] Y. Nikoloudakis, I. Kefaloukos, S. Klados, S. Panagiotakis, E. Pallis, C. Skianis, E. K. Markakis, Towards a machine learning based situational awareness framework for cybersecurity: An sdn implementation, Sensors 21 (14) (2021) 4939.

[6] A. Parkin, O. Grinchuk, Recognizing multi-modal face spoofing with face recognition networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[7] M. Arntz, T. Gregory, U. Zierahn, The risk of automation for jobs in oecd countries (2016).

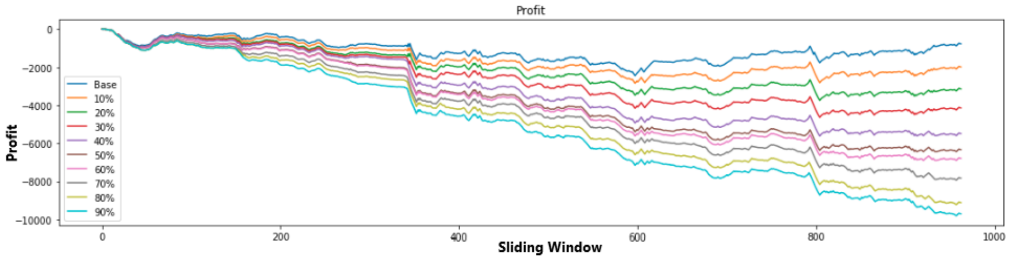


Figure 14: Single Value Attack simulated profit for 10% increments of $\epsilon = 2\omega\delta$

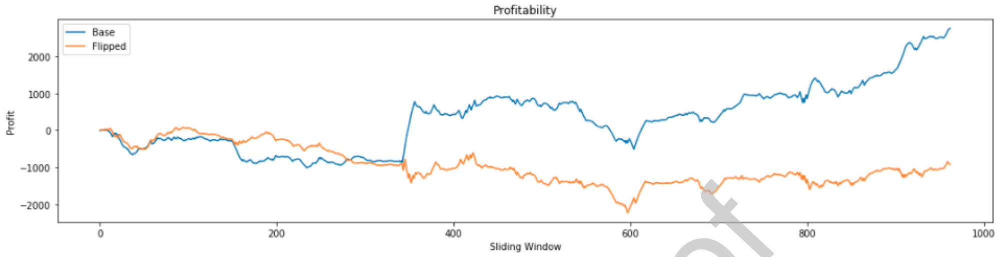


Figure 15: Label Flip simulated trading performance

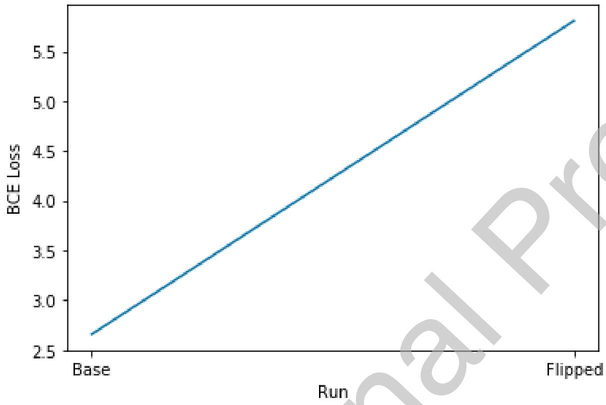
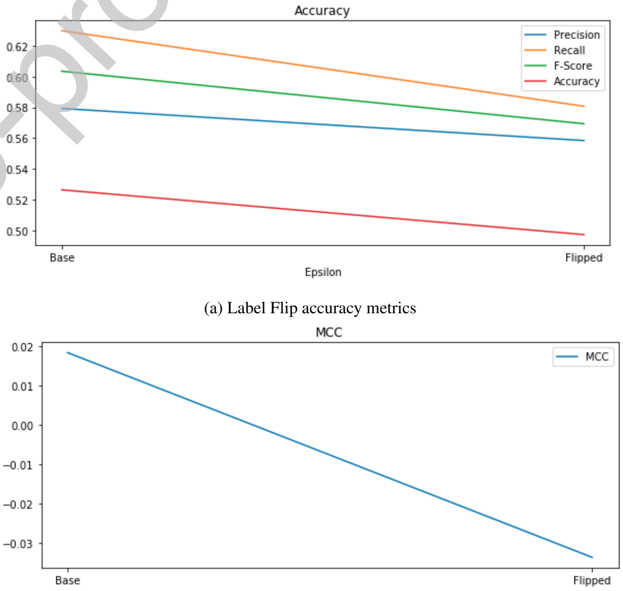


Figure 16: Label Flip BCE loss



(a) Label Flip accuracy metrics

(b) Label Flip MCC measurement

Figure 17: Label Flip accuracy and MCC

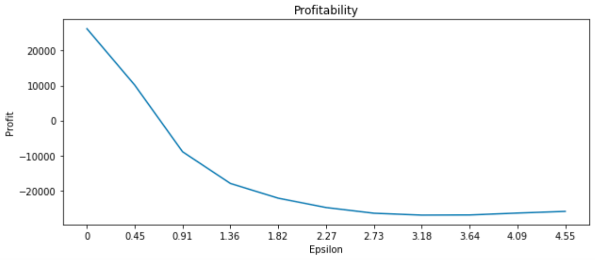


Figure 18: Profit for FGSM attack when evaluated with training data

[8] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160.

[9] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, E. K. Markakis, A survey on the internet of things (iot) forensics: challenges, approaches, and open issues, IEEE Communications Surveys & Tutorials 22 (2) (2020) 1191–1221.

[10] M. Sharif, S. Bhagavatula, L. Bauer, M. K. Reiter, 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, 2016, pp. 1528–1540.

[11] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain, arXiv preprint arXiv:1708.06733 (2017).

[12] J. Steinhardt, P. W. W. Koh, P. S. Liang, Certified defenses for data poisoning attacks, Advances in neural information processing systems 30 (2017).

[13] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, A taxonomy and survey of attacks against machine learning, Computer Science Review 34 (2019) 100199.

[14] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: Attacks and defenses for deep learning, IEEE transactions on neural networks and learning systems 30 (9) (2019) 2805–2824.

[15] M. Kearns, M. Li, Learning in the presence of malicious errors, SIAM Journal on Computing 22 (4) (1993) 807–837.

[16] D. Lowd, C. Meek, Adversarial learning, in: Proceedings of the Eleventh

ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05, Association for Computing Machinery, New York, NY, USA, 2005, p. 641–647. doi:10.1145/1081870.1081950.

[17] A. Globerson, S. Roweis, Nightmare at test time: Robust learning by feature deletion, in: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 353–360. doi:10.1145/1143844.1143889.

[18] J. Newsome, B. Karp, D. Song, Paragraph: Thwarting signature learning by training maliciously, in: D. Zamboni, C. Kruegel (Eds.), Recent Advances in Intrusion Detection, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 81–105.

[19] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, J. D. Tygar, Antidote: Understanding and defending against poisoning of anomaly detectors, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, IMC '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 1–14. doi:10.1145/1644893.1644895.

[20] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, J. D. Tygar, Can machine learning be secure?, in: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 16–25. doi:10.1145/1128817.1128824.

[21] M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning, Machine Learning 81 (2) (2010) 121–148.

[22] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, J. D. Tygar, Adversarial machine learning, in: Proceedings of the 4th ACM workshop on Security and artificial intelligence, 2011, pp. 43–58.

[23] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).

[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[25] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[26] K. Sadeghi, A. Banerjee, S. K. S. Gupta, A system-driven taxonomy of attacks and defenses in adversarial machine learning, IEEE Transactions on Emerging Topics in Computational Intelligence 4 (4) (2020) 450–467. doi:10.1109/TETCI.2020.2968933.

[27] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, Face recognition: A convolutional neural-network approach, IEEE transactions on neural networks 8 (1) (1997) 98–113.

[28] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, A.-P. Chen, Financial time-series data analysis using deep convolutional neural networks, in: 2016 7th International Conference on Cloud Computing and Big Data (CCBD), IEEE, 2016, pp. 87–92.

[29] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, Y. Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Transactions on Industrial Informatics 14 (4) (2017) 1606–1615.

[30] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Adversarial attacks on deep neural networks for time series classification, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).

[32] T. Huang, V. Menkovski, Y. Pei, M. Pechenizkiy, Bridging the performance gap between fgsm and pgd adversarial training, arXiv preprint arXiv:2011.05157 (2020).

[33] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv preprint arXiv:1611.01236 (2016).

[34] H. Xiao, H. Xiao, C. Eckert, Adversarial label flips attack on support vector machines., in: ECAI, 2012, pp. 870–875.

[35] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, K. Soman, Stock price prediction using lstm, rnn and cnn-sliding window model, in: 2017 international conference on advances in computing, communications and informatics (icacci), IEEE, 2017, pp. 1643–1647.

[36] S. Baluja, I. Fischer, Adversarial transformation networks: Learning to generate adversarial examples, arXiv preprint arXiv:1703.09387 (2017).

[37] F. Karim, S. Majumdar, H. Darabi, Adversarial attacks on time series, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

[38] Kaggle, available at: https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231 (Accessed on 11-Oct-2020) (2017).

[39] J. L. Treynor, R. Ferguson, In defense of technical analysis, The Journal of Finance 40 (3) (1985) 757–773. doi:10.1111/j.1540-6261.1985.tb05000.x.

[40] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.

[41] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, BMC genomics 21 (1) (2020) 6.

[42] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. McDaniel, On the (statistical) detection of adversarial examples, arXiv preprint arXiv:1702.06280 (2017).

[43] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples, in: International conference on machine learning, PMLR, 2018, pp. 284–293.

[44] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, D. Pérez-Cabo, No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation, IEEE Transactions on Information Forensics and Security 12 (11) (2017) 2640–2653.

[45] A. Fawzi, O. Fawzi, P. Frossard, Analysis of classifiers' robustness to adversarial perturbations, Machine Learning 107 (3) (2018) 481–508.

[46] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2016, pp. 372–387.

[47] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, T. Goldstein, Adversarial training for free!, Advances in Neural Information Processing Systems 32 (2019).

[48] S. Grierson, C. Thomson, P. Papadopoulos, B. Buchanan, Min-max training: Adversarially robust learning models for network intrusion detection systems, in: 2021 14th International Conference on Security of Information and Networks (SIN), Vol. 1, IEEE, 2021, pp. 1–8.

[49] P. Papadopoulos, O. Thornewill von Essen, N. Pitropakis, C. Chrysoulas, A. Mylonas, W. J. Buchanan, Launching adversarial attacks against network intrusion detection systems for iot, Journal of Cybersecurity and Privacy 1 (2) (2021) 252–273.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit Author Statment

Michael Gallagher: Conceptualization, Methodology, Software, Validation, Data Curation, Writing - original draft, Writing - review & editing

Nikolaos Pitropakis: Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision

Christos Chrysoulas: Conceptualization, Writing - original draft

Pavlos Papadopoulos: Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing

Alexios Mylonas: Conceptualization, Writing - review & editing

Sokratis Katsikas: Conceptualization, Writing - review & editing, Supervision

Mike Gallagher received his bachelor's in Computer Science form the Trinity College (Dublin) in 2014. He received his MSc in Advanced Security and Digital Forensics from Edinburgh Napier University in 2020. During his studies his focus was on financial technology, and advanced security architectures. Mike is having great experience on the banking eco-system, and he worked as technology expert in the Bank Of America (2013-2017). Currently he is the Technical Lead at Depop (2021 till now).

Nikolaos Pitropakis is an Associate Professor of Cyber Security at Edinburgh Napier University and the Director of Cyber Academy. He holds a degree in Informatics and Telecommunications from the National and Kapodistrian University of Athens (GR). He received his MSc in Advanced Information Systems from Athens University of Economics and Business (GR) and his Ph.D. from the University of Piraeus, Department of Digital Systems. He has worked as the Director of Cyber Security at Eight Bells LTD, as a lecturer for London South Bank University and as a postdoctoral Researcher for Georgia Institute of Technology, where he was involved in a U.S. Department of Defense project. He is a member of the Blockpass Identity Lab and his current research interests include trust and privacy using distributed ledger technology, adversarial machine learning, advanced cyber-attack attribution, data science applied on cyber security and IoT devices security. Dr Pitropakis has published 40 quality scholarly articles and has more than 550 citations.

Christos Chrysoulas received his Diploma and his Phd in Electrical Engineering from the University of Patras in 2003 and 2009 respectively. During his Phd (2004-2009) and PostDoc studies (2010-2015) his research was focused on Smart Grids, IoT, Industrial Automation, Machine Learning, Big Data, E-Learning systems, Computer Networks, High Performance Communication Subsystems Architecture and Implementation, Wireless Networks, Service Oriented Architectures (SOA), Resource Management and Dynamic Service Deployment in New Generation Networks and Communication Networks, Grid Architectures, Semantics, and Semantic Grid. He joined CISTER Research Center, Porto, as an invited Researched in 2013. He joined University of Porto as Post-Doc Research fellow in 2014 and from July 2015 he was with the University of Essex, holding a Senior Officer Researcher position. Currently he is holding a Lecturer's position in Software Engineering in the Edinburgh Napier University, UK. The outcome of this effort was properly announced in more than 40 technical papers in these areas. Dr. Christos Chrysoulas also participated as Senior Research/Engineer in both European and National Research Projects.

Pavlos Papadopoulos received his bachelor's degree from the department of Digital Systems in 2016 at the University of Piraeus, Greece. He completed an MSc in Advanced Security and Digital Forensics in 2019 at Edinburgh Napier University and is currently a PhD Student in Privacy-Preserving Systems around Security, Trust and Identity in the School of Computing at Edinburgh Napier University. Pavlos is a member of the Blockpass Identity Lab, and he is also serving as an Associate Lecturer for Edinburgh Napier University. Pavlos participated in Diffusion Berlin hackathon with a team composed of Blockpass Identity Lab's students winning the "Identity and beyond with Hyperledger - Best Business Impact of Digital Credentials" and "Machine Learning in the Decentralised World" awards. His current research interests include cybersecurity, distributed ledger technology and privacy-preserving machine learning. Pavlos is leading the Edinburgh Napier's latest venture, TrueDeploy venture, which has received funding from Scottish Enterprise and Innovate UK, to develop the project's innovative technology.

Alexios Mylonas is a Senior Lecturer at the School of Computing. He holds a PhD in Information and Communication Security and a BSc (Hons) in Computer Science from the Athens University of Economics and Business, as well as an MSc in Information Security from Royal Holloway, University of London. He has a record of excellence in working in multiple security domains and he currently leads the OWASP Dorset Chapter. Currently, his research interests focus on adversarial machine learning, APT detection, incident response and web security. He has published more than 40 papers in esteemed scientific venues and his work is well cited (> 1800 citations, h-index: 20).

Sokratis K. Katsikas was born in Athens, Greece, in 1960. He is the Director of the Norwegian Centre for Cybersecurity in Critical Sectors and Professor with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Norway. He is also Professor Emeritus of the Department of Digital Systems, University of Piraeus, Greece. In 2019 we was awarded a Doctorate Honoris Causa from the Department of Production and Management Engineering, Democritus University of Thrace, Greece. In 2021 he was ranked 7th in the security professionals category of the IFSEC Global influencers in security and fire list. He has authored or co-authored more than 300 journal papers, book chapters and conference proceedings papers. He is serving on the editorial board of several scientific journals, he has co-authored/ edited 46 books and has served on/chaired the technical programme committee of more than 800 international scientific conferences. He chairs the Steering Committee of the ESORICS Conference and he is the Editor-in-Chief of the International Journal of Information Security.