Adversarial Attacks on Deep Neural Networks for Time Series Classification

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller
IRIMAS, Université Haute-Alsace, Mulhouse, France
Email: {first-name.last-name@uha.fr}

***Abstract*—Time Series Classification (TSC) problems are encountered in many real life data mining tasks ranging from medicine and security to human activity recognition and food safety. With the recent success of deep neural networks in various domains such as computer vision and natural language processing, researchers started adopting these techniques for solving time series data mining problems. However, to the best of our knowledge, no previous work has considered the vulnerability of deep learning models to adversarial time series examples, which could potentially make them unreliable in situations where the decision taken by the classifier is crucial such as in medicine and security. For computer vision problems, such attacks have been shown to be very easy to perform by altering the image and adding an imperceptible amount of noise to trick the network into wrongly classifying the input image. Following this line of work, we propose to leverage existing adversarial attack mechanisms to add a special noise to the input time series in order to decrease the network's confidence when classifying instances at test time. Our results reveal that current state-of-the-art deep learning time series classifiers are vulnerable to adversarial attacks which can have major consequences in multiple domains such as food safety and quality assurance.**

I. Introduction

Time Series Classification (TSC) problems are encountered in various real world data mining tasks ranging from health care [1]–[3] and security [4], [5] to food safety [6], [7] and power consumption monitoring [8], [9]. As deep learning models have revolutionized many machine learning fields such as computer vision [10] and natural language processing [11], [12], researchers recently started to adopt these models for TSC tasks [13].

Following the advent of deep learning, researchers started to study the vulnerability of deep networks to adversarial attacks [14]. In the context of image recognition, an adversarial attack consists in modifying an original image so that the changes are almost undetectable by a human [14]. The modified image is called an adversarial image, which will be misclassified by the neural network, while the original one is correctly classified. One of the most famous real-life attacks consists in altering a traffic sign image so that it is misinterpreted by an autonomous vehicle [15]. Another application is the alteration of illegal content to make it undetectable by automatic moderation algorithms [14]. The most common attacks are gradient-based methods, where the attacker modifies the image in the direction of the gradient of the loss function with respect to the input image thus increasing the misclassification rate [14], [16], [17].

While these approaches have been intensely studied in the context of image recognition (e.g. NIPS competition on Adversarial Vision Challenge), adversarial attacks haven not been thoroughly explored for TSC. This is surprising as deep learning models are getting more and more popular to classify time series [2], [9], [18]–[21]. Furthermore, potential adversarial attacks are present in many applications where the use of time series data is crucial. For example, Figure 1 shows an original and perturbed time series of coffee beans spectrographs. While a deep neural network correctly classifies the original time series as Robusta beans, adding small perturbations makes it classify it as Arabica. Therefore, since Arabica beans are more valuable than Robusta beans, this attack could be used to deceive food control tests and eventually the consumers.

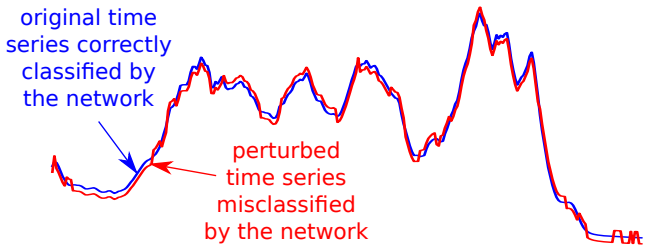

Fig. 1: Example of a perturbed time series that is misclassified by a deep network after applying a small perturbation (time series from the Coffee dataset [22] containing spectrographs of coffee beans).

In this paper, we present, transfer and adapt adversarial attacks that have been shown to work well on images to time series data. We also present an experimental study using the 85 datasets of the UCR archive [22] which reveals that neural networks are prone to adversarial attacks. We highlight specific real-life use cases to stress the importance of such attacks in real-life settings, namely food quality and safety, vehicle sensors and electricity consumption. Our findings show that deep networks for time series data are vulnerable to adversarial attacks like their computer vision counterparts. Therefore, this paper sheds the light on the need to protect against such attacks, especially when deep learning is used for sensitive TSC applications. We also show that adversarial time series learned using one network architecture can be transferred to different architectures. Finally, we discuss some mechanisms to prevent these attacks while making the models more robust to adversarial examples.

To summarize, the main contributions of this paper are:

- The definition and formalization of adversarial attacks for

TSC tasks.
- The transfer and adaptation of image-based attacks to time series data.
- An empirical study of these methods on the UCR archive datasets.
- A set of use-cases that highlight the importance of such attacks in real-life scenarios.
- An open-source framework to generate adversarial time series.
- A set of adversarial time series for each dataset in the UCR archive.
- A list of open issues to be considered in future research on this topic.

II. Background

In this section, we start with the necessary definitions for ease of understanding. We then follow by an overview of critical applications based on deep learning approaches for TSC where adversarial attacks could have serious and dangerous consequences. Finally, we present a brief survey of the current state-of-the-art methods for adversarial attacks which have been mainly proposed and validated on image datasets [14].

Definition 1: A time series $X = [x_1, x_2, \ldots, x_T]$ is an ordered set of real values. The length of X is equal to the number of real values T.

Definition 2: $D = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ is a dataset of pairs (X_i, Y_i) where X_i is a time series with Y_i as its corresponding one-hot label vector.

Definition 3: Time Series Classification (TSC) task consists of training a classifier on D in order to map from the space of possible inputs to a probability distribution over the class variable values (labels).

Definition 4: $f(\cdot) \in F : \mathbb{R}^T \rightarrow \hat{Y}$ represents a deep learning model for TSC.

Definition 5: $J_f(\cdot, \cdot)$ denotes the loss function (e.g. cross-entropy) of the model f.

Definition 6: $X^{'}$ denotes the adversarial example, a perturbed version of X (the original instance) such that $\hat{Y} \neq \hat{Y}^{'}$ and $\|X - X^{'}\|_p \leq \epsilon$.

A. Deep learning for time series classification

Since AlexNet [10] won the ImageNet competition in 2012, deep learning has seen a lot of successful applications in various domains such as reaching human level performance in image recognition problems [13] as well as different natural language processing tasks [11]. Motivated by this success, researchers and data mining practitioners started adopting these deep machine learning models in various real life TSC problems [2], [5], [9], [13], [19], [23]. In fact, deep Convolutional Neural Networks (CNNs) were shown to achieve state-of-the-art performance for TSC when evaluated over the UCR archive benchmark. Specifically, by sliding one dimensional filters over the input time series, the network is able to learn discriminative, non-linear and time invariant features useful for classification. For a more comprehensive description on how deep CNNs are being adapted for one dimensional temporal data, we refer the interested reader to our recent empirical study of deep learning models for TSC [13].

In this paper, we focus on the application of deep neural networks in crucial and sensitive decision making systems, thus motivating the investigation of neural network's vulnerabilities to adversarial examples. In [2], CNNs were used to mine temporal electronic health data for risk prediction and disease sub-typing. In situations where algorithms are taking the decision for reimbursement of medical treatment, tampering medical records in an imperceptible manner could eventually lead to fraud. Apart from the health care industry, deep CNNs are also being used when monitoring power consumption from houses or factories. For example in [9], time series data from smart grids were analyzed for electricity theft detection, where in such use cases perturbed data can help thieves to avoid being detected. Other crucial decision making systems such as malware detection in smart-phones, leverage temporal data in order to classify if an Android application is malicious or not [5]. Using adversarial attacks, a hacker might generate synthetic data from his/her application allowing it to bypass the security systems and get it installed on the end user's smart-phone. Finally, when deep neural networks are deployed for road anomaly detection [23], perturbing the data recorded by sensors placed on the road could help the entities responsible for such life threatening anomalies, to avoid being captured. We should note that due to space limitations, this list of potential attacks is not exhaustive.

B. Adversarial attacks

Szegedy *et al.* [24] were the first to introduce adversarial examples against deep neural networks for image recognition tasks in 2014. Following these intriguing findings, a huge amount of research has been dedicated to generating, understanding and preventing adversarial attacks on deep neural networks [15]–[17].

Most of these methods have been proposed for image recognition tasks [14]. For example, in [16], a fast gradient-based attack was developed as an alternative to expensive optimization techniques [24], where the authors explained the presence of such adversarial examples with the hypothesis of linearity for deep learning models. This kind of attack was also extended by a more costly iterative procedure [17], where the authors showed for the first time that even printed adversarial images (*i.e.* perceived by a camera) are able to fool a pre-trained network. More recently, it has been shown that perturbing stop signs can trick autonomous vehicles into misclassifying it as a speed limit sign [15].

Other fields such as Natural Language Processing have also been investigated to create adversarial attacks such as adding distracting phrases at the end of a paragraph in order to show that deep learning-based reading comprehension systems were not able to distinguish subtle differences in text [25]. For a review on the different adversarial attacks for deep learning systems, we refer the interested readers to a recent survey in [14].

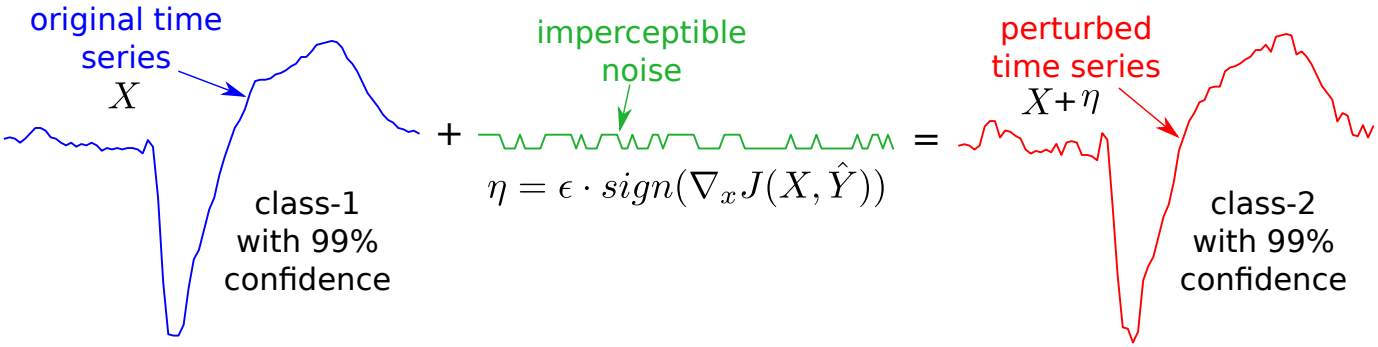

Fig. 2: Example of perturbing the classification of an input time series from the TwoLeadECG dataset by adding an imperceptible noise computed using the Fast Gradient Sign Method (FGSM).

For general TSC tasks, it is surprising how adversarial attack approaches have been ignored by the community. The only previous work mentioning attacks for TSC is [26]. By adapting a soft K Nearest Neighbors (KNN) coupled with Dynamic Time Warping (DTW), the authors showed that adversarial examples could fool the proposed nearest neighbors classifier on a single simulated dataset (synthetic_control from the UCR archive [22]). However, the fact that the KNN classifier is no longer considered as the state-of-the-art classifier for time series data [27], we believe that it is important to investigate the generation of adversarial time series examples that deteriorate the accuracy of state-of-the-art classifiers such a Residual Network (ResNet) [13], [19] and to validate it on the whole 85 datasets in the UCR archive. Finally, we formally define an adversarial attack on deep neural networks for TSC.

Definition 7: Given a trained deep learning model f and an original input time series X, generating an adversarial instance $X^{'}$ can be described as a box-constrained optimization problem.

$$\min_{X^{\theta}} \|X^{'} - X\| \ s.t. \\ f(X^{'}) = \hat{Y}', \ f(X) = \hat{Y} \ and \ \hat{Y} \neq \hat{Y}' \quad (1)$$

Let $\eta = X - X^{'}$ be the perturbation added to X, which corresponds to a very low amplitude signal. Figure 2 illustrates this process where the green time series corresponds to the added perturbation η. The optimization problem in (1) minimizes the perturbation while misclassifying the input time series.

III. Adversarial attacks for time series

In this section, we describe the ResNet architecture and present two attack methods that we then use to generate adversarial time series examples for the ResNet model.

A. Residual Network

ResNet was originally proposed for TSC in [19], where it was validated on 44 datasets from the UCR archive [22]. In a recent empirical evaluation [13] on the 85 datasets from the UCR archive, we identified that ResNet achieved state-of-the-art performance for TSC, with results that are not significantly different than the Collective Of Transformation-based Ensembles, the current state-of-the-art classifier, which is an ensemble of 35 classifiers [27]. Note that our adversarial attack methods are independent of the chosen network architecture, and that we chose ResNet for its robustness [13] as well as its use in many critical domains such as malware detection [23]. In addition, adversarial examples are known to be transferable across different neural network architectures which enables the synthetic time series to fool other deep learning models: a technique known as black-box attack [14].

Figure 3 illustrates the adopted architecture in our adversarial attack experiments. The network's input is a time series of length T. The output of the network is a probability distribution over the K possible classes in the dataset. The first nine layers are convolutional layers with the Rectified Linear Unit (ReLU) as activation function [10]. Each convolutional layer is followed by a batch normalization operation [28]. In each convolutional block, the first, second and third convolutions are composed respectively of filters of length 8, 5 and 3. The first, second and last convolutional blocks are comprised respectively of 64, 128 and 128 filters. For all convolutions the stride is set to 1.

Each convolutional layer takes as input a time series and applies a non-linearity to transform it into a multivariate time series whose dimensions are inferred from the number of filters in each layer. The tenth layer is composed of a Global Average Pooling (GAP) operation which takes the output of the third convolutional block and averages the time series over the time dimension. This averaging operation reduces drastically the number of parameters in a deep model while enabling the use of a class activation map which allows an interpretation of the learned features [19]. The output of layer ten is then fed to a softmax classification layer whose number of neurons is equal to the number of classes K in the dataset. The network is trained for 1500 epochs using Adam [29] (with default hyperparameters) to minimize the objective cost function: cross-entropy.

B. Fast Gradient Sign Method

FGSM was first proposed in [16] to generate adversarial images that fooled the famous GoogLeNet model. FGSM is considered "fast" and replaces the expensive linear search

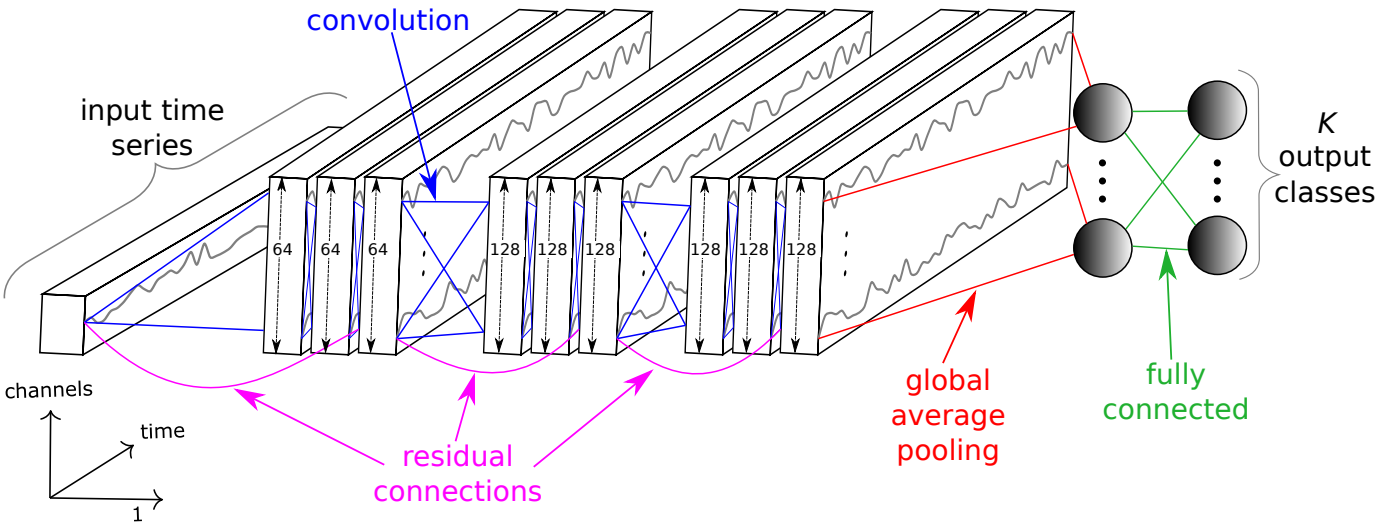

Fig. 3: The deep Residual Network (ResNet) architecture for Time Series Classification (TSC).

method previously proposed in [24]. The attack is based on a one step gradient update along the direction of the gradient's sign at each time stamp. The perturbation process (illustrated in Figure 2) can be expressed as:

$$\eta = \epsilon \cdot sign(\nabla_x J(X, \hat{Y})) \tag{2}$$

where ϵ denotes the magnitude of the perturbation (a hyperparameter). The adversarial time series $X^{'}$ can be easily generated with $X^{'} = X + \eta$. The gradient can be efficiently computed using back-propagation.

C. Basic Iterative Method

BIM extends FGSM by applying it multiple times with a small step size and clip the obtained time series elements after each step to ensure that they are in an ϵ-neighborhood of the original time series [17]. In fact, by adding smaller changes or perturbations in an iterative manner, the method is able to generate adversarial examples that are closer to the original samples and have a better chance of fooling the network. Algorithm 1 shows the different steps of this iterative attack which requires setting three hyperparameters: (1) the number of iterations I; (2) the amount of maximum perturbation ϵ and (3) the per step small perturbation α. In our experiments we have set $\epsilon = 0.1$ heuristically similarly to [16], [17] and the rest of BIM hyperparameters were left at their default value in the Cleverhans API [30].

Algorithm 1 Iterative Adversarial Attack

Parameter: I, ϵ, α
Input: original time series X & its label $\hat{Y}$
1: **Output:** perturbed time series $X^{'}$
2: $X^{'} \leftarrow X$
3: **for** $i = 1$ to I **do**
4: $\quad \eta = \alpha \cdot sign(\nabla_x J(X^{'}, \hat{Y}))$
5: $\quad X^{'} = X^{'} + \eta$
6: $\quad X^{'} = min\{X + \epsilon, max\{X - \epsilon, X^{'}\}\}$
7: **end for**

IV. Results

A. Experimental setup

To train the deep neural network, we leveraged the parallel computation of a cluster of more than 60 GPUs (a mix of GTX 1080 Ti, Tesla K20, K40 and K80). All of our experiments were evaluated on the 85 datasets from the publicly available UCR archive [22]. The model was trained/tested using the original training/testing splits provided in the archive. To perform the attacks, we have adapted the Cleverhans API [30] by extending the well known attacks for time series data and perturbed only the test instances without using the test labels, similarly to the computer vision literature [14].

For reproducibility and to allow the time series community to verify and build on our findings, the source code for generating adversarial time series is publicly available on our GitHub repository[1]. In addition, we provide on our companion web page[2] the raw results, our pre-trained models as well as a set of perturbed time series for each dataset in the UCR archive. This would allow time series data mining practitioners to test the robustness of their machine learning models against adversarial attacks.

B. Results on the whole UCR archive

For all datasets, both attacks managed to reduce ResNet's accuracy. One exception is the DiatomSizeReduction dataset which is the smallest one in the archive with an already low original accuracy equal to 30% due to overfitting [13]. Figure 4 shows the accuracy variation for both attacks with respect to the network's original accuracy on the UCR archive. On average, over the 85 datasets, FGSM and BIM managed to reduce the model's accuracy respectively by 43.2% and

[1] https://github.com/hfawaz/ijcnn19attacks
[2] https://germain-forestier.info/src/ijcnn2019/

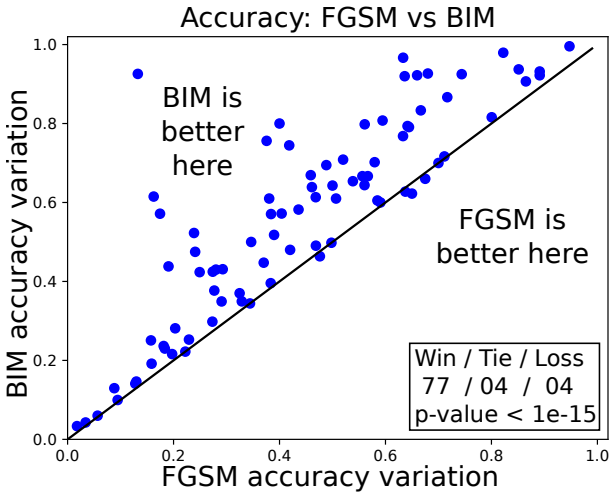

Fig. 4: Accuracy variation for two attacks (FGSM and BIM) with respect to ResNet's original accuracy.

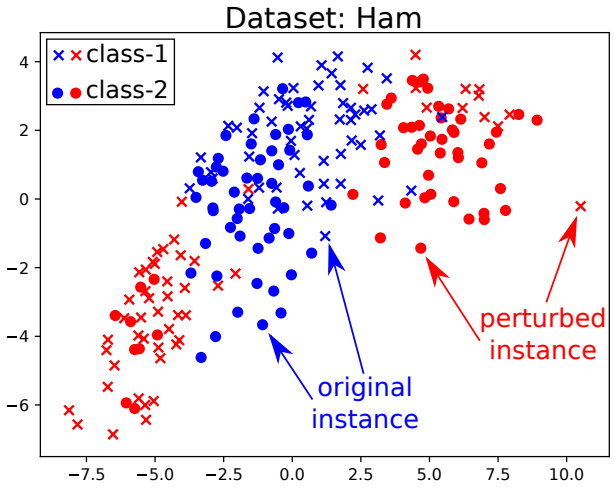

Fig. 5: Multidimensional Scaling (MDS) showing the distribution of perturbed time series on the whole test set of the Ham dataset where the accuracy decreased from 80% to 21% after performing the BIM attack.

56.89%. The Wilcoxon signed-rank test indicates that BIM is *significantly* better than FGSM in decreasing the model's accuracy, with a p-value $\leq 10^{-15}$. However, we should note that FGSM is a fast approach allowing real-time generation of adversarial time series whereas BIM is time-consuming and requires a certain number of iterations I.

By analyzing the use-cases where both attacks failed to fool the classifier, we found out that the corresponding datasets have two interesting characteristics that could explain the classifier's robustness to adversarial examples. The first one is that 50% of the simulated datasets (CBF, Two_patterns and synthetic_control) in the archive are in the top six hardest datasets to attack. Perhaps since these are synthetic datasets generated by humans to serve some human intuition for TSC, small perturbations imperceptible by humans, are not enough to alter the classifier's decision. The second observation is that a network trained on datasets with time series of short length is harder to fool. This is rather expected since the less data points we have, the less amount of perturbation the attacker is allowed to add. For example ItalyPowerDemand contains the shortest sequences ($T = 24$) and is the second most hardest use-case for both attacks.

C. Multi-Dimensional Scaling

We used Multi-Dimensional Scaling (MDS) [31], [32] with the objective to gain some insights on the spatial distribution of the perturbed time series compared to the original ones. MDS uses a pairwise distance matrix as input and aims at placing each object in an N-dimensional space such as the between-object distances are preserved as well as possible. Using the Euclidean Distance (ED) on a set of time series (original and perturbed), it is then possible to create a similarity matrix and apply MDS to display the set into a two dimensional space. The latter straightforward approach supposes that the ED is able to strongly separate the raw time series, which is usually not the case evident by the low accuracy of the nearest neighbor when coupled with the ED [27]. Therefore, we decided to use the *linearly* separable representation of time series from the output of the GAP layer, which is used as input to the softmax *linear* classifier (multinomial logistic regression). More precisely, for each input time series, the last convolution outputs a multivariate time series whose dimensions are equal to the number of filters (128) in the last convolution, then the GAP layer averages the latter 128-dimensional multivariate time series over the time dimension resulting in a vector of 128 real values over which the ED is computed. This enables the MDS projection to be as close as possible to ResNet's latent representation of the time series. As we worked with the ED, we used Metric MDS [31] that minimizes a cost function called *Stress* which is a residual sum of squares:

$$Stress_D(X_1, \ldots, X_N) = \left(\frac{\sum_{i,j} \left(d_{ij} - \|x_i - x_j\|\right)^2}{\sum_{i,j} d_{ij}^2} \right)^{1/2} \quad (3)$$

where d_{ij} is the ED between the GAP vectors of time series X_i and X_j. Obviously, one has to be careful about the interpretation of MDS output, as the data space is highly simplified (each time series X_i is represented as a single data point x_i).

D. Attacks on food quality and safety

The determination of food quality, type and authenticity along with the detection of adulteration are major issues in the food industry [7]. With meat related product, authenticity checking concerns for example the identification of substitution of high value raw materials with cheaper materials like less costly cuts, offal, blood, water, eggs or other types of proteins. These substitutions not only decease the consumers but can also cause severe allergic responses as the

substitute materials are hidden. Discriminating meat that has been frozen-and-thawed from fresh meat is also an important issue, as refreezing food can result in an increased amount of bacteria. Spectroscopic methods have been historically very successful at evaluating the quality of agricultural products, especially food [7]. This technique is routinely used as a quality assurance tool to determine the composition of food ingredients.

In this context, an adversarial attack could be used to modify recorded spectrographs (seen as time series) in order to hide the low qualify of the food. The Beef dataset [33] (from the UCR archive) contains four classes of beef spectrograms, from pure and adulterated beef with varying degrees of potential adulterants (heart, tripe, kidney, and liver). An adversarial attack could thus consist in modifying an adulterated beef to make a network classify it as pure beef. For this dataset, FGSM and BIM reduced the model's accuracy respectively by 56.7% and 66.7%.

The Ham dataset [34] contains measurements from 19 Spanish and 18 French dry-cured hams, with the goal to distinguish the provenance of the food. An adversarial attack could consist in perturbing the spectrograms to hide the real provenance of the food. Figure 5 shows the MDS projection of the original and perturbed instances for Ham's test set, where one can see that the adversarial examples are *pushed* toward the other class.

The Coffee dataset [6] is a two class problem to distinguish between Robusta and Arabica coffee beans. Arabica beans are valued most highly by the trade, as they are considered to have a finer flavor than Robusta. An adversarial attack could consist in altering the spectrograms to make Robusta beans look like Arabica beans. Figure 6 shows the MDS representation of the original and perturbed time series from the test set. We can clearly see how the instances are *pushed* toward the class frontiers after performing the FGSM attack.

E. Attacks on vehicle sensors

The increase in the number of sensors and other electrical devices has drastically augmented the amount of data produced in the industry. These data are now routinely used to monitor systems or to perform predictive maintenance and prevent failures [35]. The car industry is not an exception with the increasing number of sensors present in modern vehicles, especially for advanced driver assistance systems and autonomous driving.

Data are also used to perform diagnostic on vehicles in order to detect engine problems or compliance with environmental regulations. In this context, an adversarial attack could consist in altering sensor readings in order to hide a specific problem or to pass a CO_2 emission test. The famous "dieselgate" (or "emissionsgate") [36] made this kind of attack a reality as multiple automakers have been suspected of using emission control systems during laboratory emissions testing.

To illustrate this use case, we used the FordA datasets (from the UCR archive) that was was originally used in a competition in the IEEE World Congress on Computational

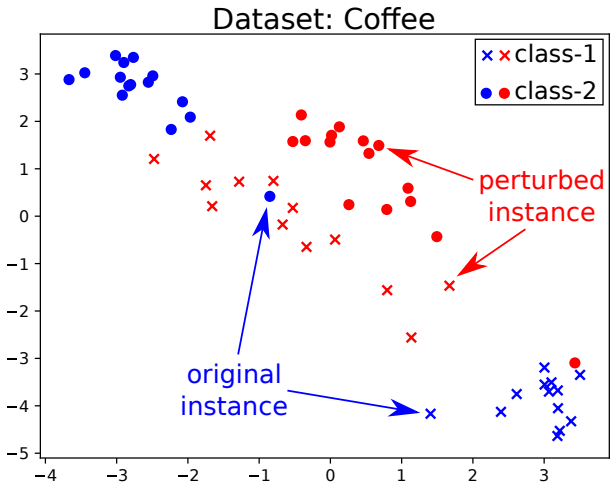

Fig. 6: Multidimensional Scaling (MDS) showing the distribution of perturbed time series on the whole test set of the Coffee dataset where the accuracy decreased from 100% to 50% after performing the FGSM attack.

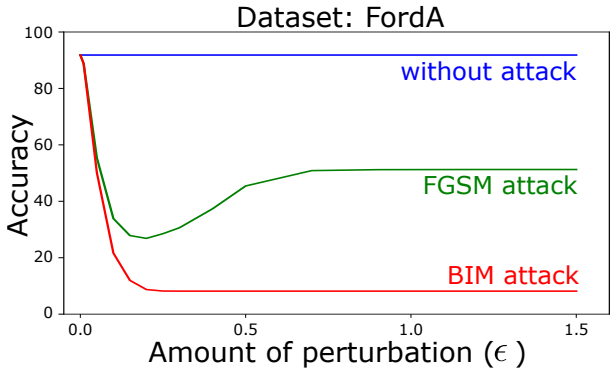

Fig. 7: Accuracy variation with respect to the amount of perturbation for FGSM and BIM attacks on FordA.

Intelligence, 2008. The classification problem is to diagnose whether a certain symptom exists or not in an automotive subsystem. Each case consists of 500 measurements of engine noise and a class label. In this context, an attack could consist in hiding an engine problem. In practice for the FordA dataset, the model's accuracy decreased by 57.9% and 70.2% when applying respectively the FGSM and BIM attacks.

Figure 7 depicts the variations of ResNet's accuracy on FordA with respect to the amount of perturbation ϵ allowed for the FGSM and BIM attacks. As expected [17], we found that FGSM fails to generate adversarial examples that can fool the network for larger values of ϵ, whereas the BIM produces perturbed time series that can reduce a model's accuracy to almost 0.0%. This can be explained by the fact that BIM adds a small amount of perturbation α on each iteration whereas FGSM adds ϵ amount of noise for each data point in the series that may not be useful for misclassifying the test sample.

F. Attacks on electricity consumption

Smart meters are electronic devices that record electric power consumption while sending information to the electricity supplier for monitoring, billing and data analysis. These meters typically register energy hourly and report back at least once a day to the supplier by leveraging a two-way communication channel between the device and the supplier's central system. These smart meters have raised a set of concerns in public opinion especially because they send detailed information about how much electricity is being used for each time stamp. Precisely, it has been shown that it is possible to know exactly which type of electric device is or has been used from simply analyzing the electricity consumption data [8]. In this context, an attack could consist in modifying the electricity consumption time series of one device to make it recognized as another in order to hide which devices are actually used by a specific user.

To illustrate this use case, we used the SmallKitchenAppliances dataset from the UCR archive that was recorded as part of government sponsored study called Powering the Nation [8]. By collecting and analyzing behavioural data about consumers' daily use of electricity within their homes, the goal is to reduce the UK's carbon footprint. The dataset contains readings from 251 households recorded over a month. Each univariate time series has a length equal to 720 corresponding to 24 hours of readings taken every two minutes. The three classes are: Kettle, Microwave and Toaster. For this dataset, FGSM and BIM managed to reduce the classifier's accuracy respectively by 38.4% and 57%.

The ItalyPowerDemand dataset [37], another dataset from the UCR archive, contains twelve monthly electrical power demand time series from Italy. The task is to differentiate between instances that correspond to winter months (October to March) and summer months (April to September). This dataset contains the shortest time series in the archive ($T = 24$), thus requiring a higher amount of perturbation ϵ in order to be misclassified. Figure 8 shows the variation of the model's accuracy as well as the shape of a time series from the ItalyPowerDemand dataset with respect to the amount of noise that is added. For this example, both attacks needed higher values of perturbation ($\epsilon \geq 0.3$) rather than the default setting ($\epsilon = 0.1$).

G. Transferability of adversarial examples

To evaluate the transferability of perturbed time series, we used the Fully Convolutional Neural Network (FCN) which was originally proposed in [19] and was recently shown to be the second most accurate deep time series classifier [13] when evaluated on the UCR archive [22]. We used the test sets altered with FGSM and BIM using ResNet and try to classify it with FCN (both were originally trained on the same train set). For both FGSM and BIM attacks, FCN's accuracy decreases respectively by 38.2% and 42.8% which shows that adversarial examples are capable of generalizing to a different network architecture. The Wilcoxon signed-rank test also shows that BIM is *significantly* better than FGSM in reducing FCN's accuracy, with a p-value $\leq 10^{-7}$. This type of attacks is known as "black box" where the attackers do not have access to the target model's internal parameters (FCN) yet they are able to generate perturbed time series that fool the classifier.

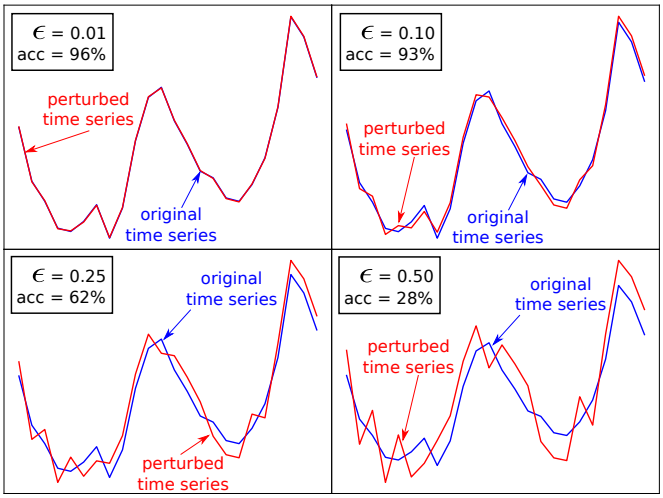

Fig. 8: Accuracy variation for ItalyPowerDemand with respect to the perturbation ϵ where FGSM managed to fool the network with this example for $\epsilon \geq 0.3$.

H. How can we prevent such attacks ?

Countermeasures for adversarial attacks [14] follow two defense strategies: (1) *reactive*: identify the perturbed instance; (2) *proactive*: improve the network's robustness without generating adversarial examples. One of the most straightforward proactive methods is *adversarial training*, which consists of (re)training the classifier with adversarial examples. Other reactive techniques consist of detecting the adversarial examples during testing. However, most of these detectors are still prone to attacks that are designed specifically to fool the detectors [14]. Therefore, we think that the time series community would have much to offer in this area by leveraging the decades of research into non-probabilistic classifiers such as the nearest neighbor coupled with DTW [38]. Running classifiers against the adversarial examples that we provide in this paper is a first step toward identifying vulnerable models and making them more robust to such type of attacks.

V. Conclusion

In this paper, we introduced the concept of adversarial attacks on deep learning models for time series classification. We defined and adapted two attacks, originally proposed for image recognition, for the TSC task. We showed how adversarial perturbations are able to reduce the accuracy for the state-of-the-art deep learning classifier (ResNet) when evaluated on the UCR archive benchmark. With deep neural networks becoming frequently adopted by time series data mining practitioners in real-life critical decision making systems, we shed the light on some crucial use cases where adversarial attacks could have serious and dangerous consequences.

In the future, we would like to investigate countermeasures techniques to defend machine learning models against such attacks while exploring the transferability of adversarial examples to other non deep learning state-of-the-art classifiers. Finally, we would like to further explore the dozens of adversarial attacks that are published each year in order to identify and protect vulnerable deep learning models for TSC.

ACKNOWLEDGEMENTS

We would like to thank the providers of the UEA/UCR datasets, as well as NVIDIA Corporation for the Quadro P6000 grant and the Mésocentre of Strasbourg for providing access to the cluster.

REFERENCES

[1] S. M. Abdelfattah, G. M. Abdelrahman, and M. Wang, "Augmenting the size of EEG datasets using generative adversarial networks," in *International Joint Conference on Neural Networks*, 2018, pp. 1–6.

[2] T. Ma, C. Xiao, and F. Wang, "Health-ATM: A deep architecture for multifaceted patient health record representation and risk prediction," in *SIAM International Conference on Data Mining*, 2018, pp. 261–269.

[3] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Evaluating surgical skills from kinematic data using convolutional neural networks," in *International Conference On Medical Image Computing and Computer Assisted Intervention*, 2018, pp. 214–221.

[4] C. W. Tan, G. I. Webb, and F. Petitjean, "Indexing and classifying gigabytes of time series under time warping," in *SIAM International Conference on Data Mining*, 2017, pp. 282–290.

[5] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in *IEEE Annual Computer Software and Applications Conference*, 2016, pp. 577–582.

[6] R. Briandet, E. K. Kemsley, and R. H. Wilson, "Discrimination of Arabica and Robusta in instant coffee by Fourier transform infrared spectroscopy and chemometrics," *Journal of agricultural and food chemistry*, vol. 44, no. 1, pp. 170–174, 1996.

[7] A. Nawrocka and J. Lamorska, "Determination of food quality by using spectroscopic methods," in *Advances in agrophysical research*, 2013.

[8] P. Owen and R. Foreman, "Powering the nation: Household electricity using habits revealed," *Energy Saving Trust/DECC/DEFRA*, 2012.

[9] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[11] M. Yang, Q. Qu, K. Lei, J. Zhu, Z. Zhao, X. Chen, and J. Z. Huang, "Investigating deep reinforcement learning techniques in personalized dialogue generation," in *SIAM International Conference on Data Mining*, 2018, pp. 630–638.

[12] X. Wang, C. Li, and B. Xu, "Hierarchical tree long short-term memory for sentence representations," in *International Joint Conference on Neural Networks*, 2018, pp. 1–6.

[13] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, 2019.

[14] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *ArXiv*, 2017.

[15] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations - Workshop track*, 2017.

[18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep neural network ensembles for time series classification," in *IEEE International Joint Conference on Neural Networks*, 2019.

[19] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *International Joint Conference on Neural Networks*, 2017, pp. 1578–1585.

[20] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," in *International Workshop on Advanced Analytics and Learning on Temporal Data, ECML PKDD*, 2018.

[21] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," in *IEEE International Conference on Big Data*, 2018, pp. 1367–1376.

[22] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR time series classification archive," 2015.

[23] F. S. Cabral, M. Pinto, F. A. L. N. Mouzinho, H. Fukai, and S. Tamura, "An automatic survey system for paved and unpaved road classification and road anomaly detection using smartphone sensor," in *IEEE International Conference on Service Operations and Logistics, and Informatics*, 2018, pp. 65–70.

[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ArXiv*, 2014.

[25] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[26] I. Oregi, J. Del Ser, A. Perez, and J. A. Lozano, "Adversarial sample crafting for time series classification with elastic similarity measures," in *International Symposium on Intelligent and Distributed Computing*, 2018, pp. 26–39.

[27] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[30] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *ArXiv*, 2018.

[31] J. B. Kruskal and M. Wish, "Multidimensional scaling. number 07–011 in sage university paper series on quantitative applications in the social sciences," 1978.

[32] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *IEEE International Conference on Data Mining*, 2017, pp. 865–870.

[33] O. Al-Jowder, E. Kemsley, and R. H. Wilson, "Detection of adulteration in cooked meat products by mid-infrared spectroscopy," *Journal of agricultural and food chemistry*, vol. 50, no. 6, pp. 1325–1329, 2002.

[34] R. Olias, B. Maldonado, P. Radreau, G. Le Gall, F. Mulholland, I. J. Colquhoun, and E. K. Kemsley, "Sodium dodecyl sulphate-polyacrylamide gel electrophoresis of proteins in dry-cured hams: Data registration and multivariate analysis across multiple gels," *Electrophoresis*, vol. 27, no. 7, pp. 1288–1299, 2006.

[35] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2015.

[36] C. Brand, "Beyond dieselgate: Implications of unaccounted and future air pollutant emissions and energy use for cars in the United Kingdom," *Energy Policy*, vol. 97, pp. 1–12, 2016.

[37] E. Keogh, L. Wei, X. Xi, S. Lonardi, J. Shieh, and S. Sirowy, "Intelligent icons: Integrating lite-weight data mining and visualization into GUI operating systems," in *IEEE International Conference on Data Mining*, 2006, pp. 912–916.

[38] C. W. Tan, M. Herrmann, G. Forestier, G. I. Webb, and F. Petitjean, "Efficient search of the best warping window for dynamic time warping," in *SIAM International Conference on Data Mining*, 2018, pp. 225–233.