# A New Deep Learning Pipeline for Acoustic Attack on Keyboards

Massimo Orazio Spata(✉), Alessandro Ortis, Sebastiano Battiato, and Valerio Maria Russo

Dipartimento di Matematica ed Informatica, Università degli Studi di Catania, Catania, Italy
massimo.spata@unict.it, {ortis,battiato}@dmi.unict.it, valeriomaria.russo@studium.unict.it

**Abstract.** The increasing reliance on services based on recent Artificial Intelligence advancements has elevated concerns about security vulnerabilities, leading to the exploration of novel attack vectors such as keystroke acoustic attacks on keyboards. This research delves into a deep learning approach for such attacks, which exploits acoustic emissions produced during typing to infer sensitive information. Traditional methods of keystroke acoustic attacks have relied on hand-engineered features and shallow classifiers, often failing to capture the intricate patterns within the acoustic data. In contrast, deep learning models have demonstrated remarkable capabilities in learning intricate patterns from complex data sources. We propose the exploitation of a Temporal Convolutional Network (TCN) to process acoustic signals, providing a more sophisticated and adaptive approach for keystroke acoustic attack analysis. The employed deep learning model showcases superior performance in multiple dimensions achieving a peak validation accuracy of 98.3% for keystrokes recorded by phone, and 93.05% for keystrokes recorded via Zoom, obtaining the best performances with respect the related prior art.

**Keywords:** Acoustic side channel attack · Deep learning · User security and privacy · Laptop keystroke attacks · Zoom-based acoustic attacks

## 1 Introduction and Motivation

In the landscape of cybersecurity, the emergence of unconventional attack vectors necessitates innovative defense strategies. Keystroke acoustic attacks, an intriguing avenue, exploit the inadvertent sound produced during typing to decipher sensitive information, posing a considerable threat to digital security. Modern acoustic attacks compromise data security and provide malicious third parties advanced tools for leaking information about passwords, conversations, messages as well as other sensitive information. Moreover, such attacks are now simpler with widespread of high-quality audio microphones which acquire clear and high-quality audio without specific post processing neither rate limitations. ASCAs (Acoustic Side Channel Attack), have received extensive research attention, within the cybersecurity's topic, and they are employed successfully in the literature [1–6].


K. Arai (Ed.): IntelliSys 2024, LNNS 1065, pp. 402–414, 2024.
https://doi.org/10.1007/978-3-031-66329-1_26

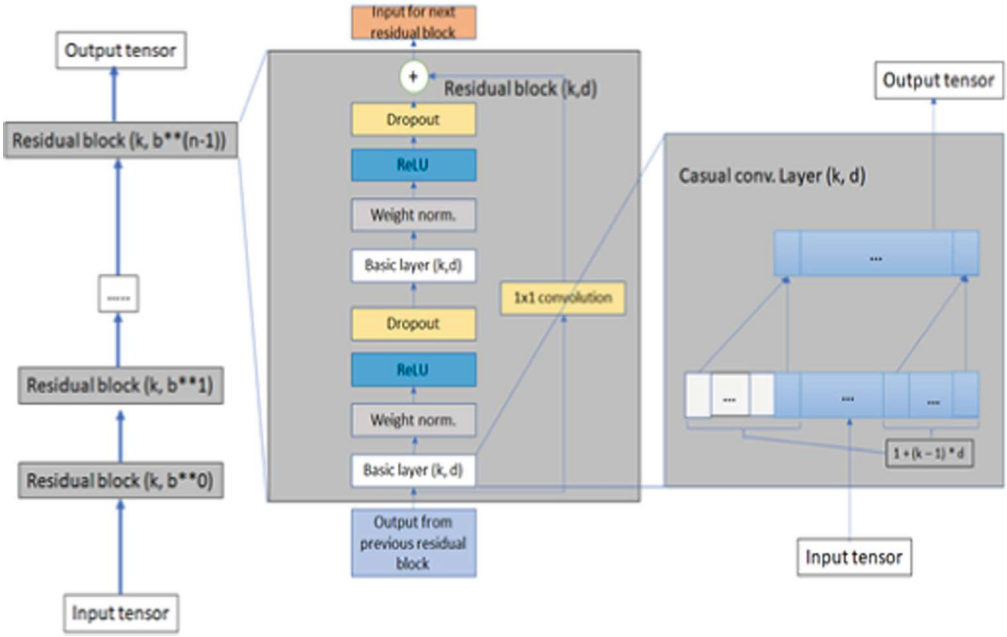


**Fig. 1.** Implemented temporal convolutional network model

Traditional methodologies for keystroke acoustic attacks have often relied on simplistic analysis techniques, limited in their capacity to capture complex temporal dynamics inherent in typing sounds. The addressed task received a significant increasing attention in the scientific literature of the last years as cited in the paper [7–11]. Recently, a deep learning model have been used in order to classify laptop keystrokes, just using a standard smartphone integrated microphone [12]. Experiments over multiple evaluation settings shown as related overall performances outperforms a significant pool of previous works [8, 9, 13–16]. The model in study [12] has been trained on two different datasets [17] created with keystrokes recorded by a nearby phone and the video-conferencing software Zoom, whereas classifier achieved respectively a peak accuracy of 95% and 93%. The authors exploited the CoAtNet model, a recent deep neural architecture based on attention mechanism [18]. For the experiments we have been considered the same dataset splitting used in study [12] and all experimental settings as done by Harrison et al. [12] to conduct a fair comparison. Moreover, we have been executed other experiments with different keyboards, and different smartphone position in order to measure the presented model accuracy in different experimental conditions. To contrast overfitting, we have pursued an in-depth examination of this phenomenon, prioritizing the model's generalization capability predicting keystroke in unseen data. This paper presents a novel pipeline for acoustic attack on keyboards, supported by a comparative evaluation with [12] and with our own specially collected dataset. The novelty is mainly due to exploitation of TCN models, usually applied on different tasks, for acoustic keyboard attack. The TCN facilitates the modeling of complex temporal dependencies, enabling extraction of latent patterns within the acoustic emissions during typing.

Its core strength lies in the capability to capture and process sequential data, dynamically adapting to the variations in typing speed, rhythm, and inter-key intervals. This is a first step of larger research comprising benchmarking of several architectures and

**Table 1.** TCN model setup

| TCN model parameters | Values |
|---|---|
| Number of layers | 4 |
| Number of classes | 36 |
| Number of filters | 128 |
| Batch size | 32 |
| Learning rate | 0.001 |
| Num channels | 7 |
| Kernel size | 3 |
| Dropout | 0.2 |
| Epochs | 500 |
| Input size | 1 |

models. Other related approaches [8, 9, 13, 15, 16] make use of different settings and methods, obtaining overall accuracy on different ranges. Addressing this challenge, this work focuses on the Temporal Convolutional Network (TCN) [19, 20] methodology as a promising approach to counter such keystroke acoustic vulnerabilities. Casual convolution is calculated as:

$$y_i = \sum_{j=0}^{k-1} c_j x_{j-1} \tag{1}$$

where: $x_j$ is an input tensor, $y_i$ is an output tensor, $k$ is the convolution kernel and $c_j$ is a convolution weight. The proposed TCN method, has been implemented the following casual convolution with convolution kernel $k = 3$ and $padding = k-1$ (see Fig. 1). To perform causal convolution, we incorporate padding $(k-1)$ on the left side of the input tensor. To execute causal convolution, we employ classical 1−D convolution with padding and trim elements from the right side. Employing the dilation technique within a causal convolutional layer enhances the coverage of the input time series and substantially reduces computational costs. In the TCN architecture, it is assumed that the sequence of causal convolutional layers has a dilation factor of $2^{i-1}$. The overall configuration of proposed TCN model architecture, is reported in Table 1. Utilizing ReLU as the activation function for TCN is recommended [19]. To address potential gradient propagation issues in the hidden layers, we employ weight normalization for each convolutional layer. Additionally, dropout regularization value 0.2 is applied after every convolutional layer within the central neural network layer of TCN.

The TCN paradigm offers a groundbreaking solution by harnessing the power of deep learning and temporal convolutional architectures. This approach facilitates the modeling of complex temporal dependencies, enabling the extraction of latent patterns within the acoustic emissions during typing. The TCN's core strength lies in its capability to capture and process sequential data, as it can dynamically adapt to the variations in typing speed,

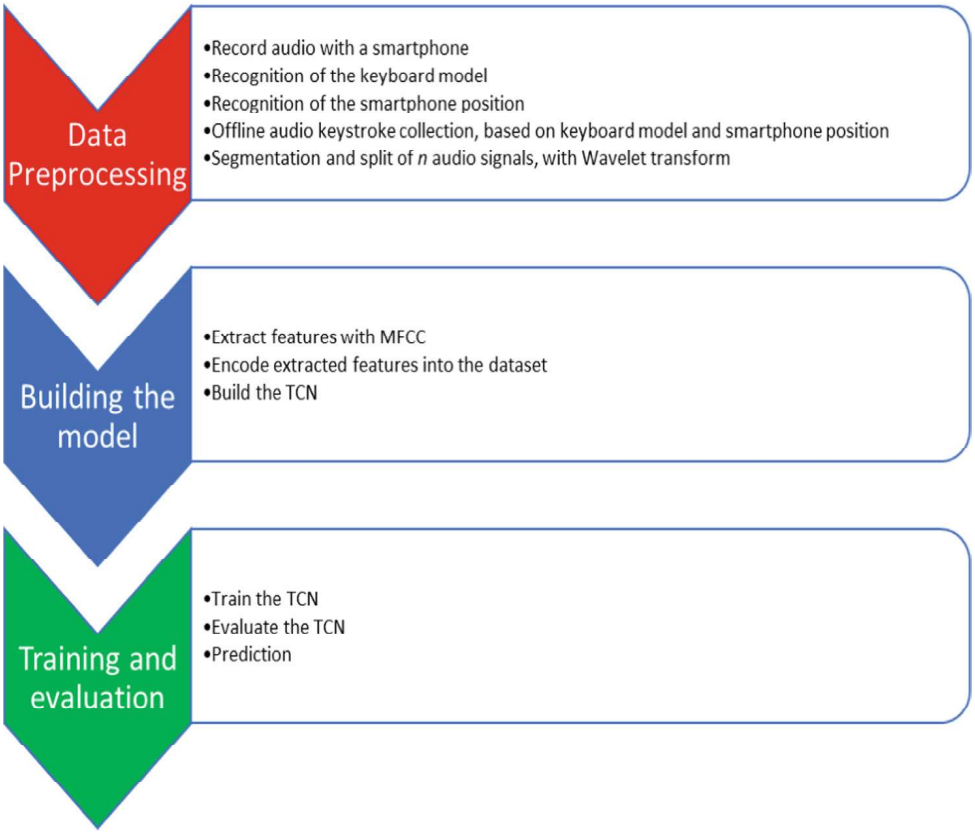


**Fig. 2.** The new adopted pipeline

rhythm, and inter-key intervals. By incorporating dilated convolutions, the TCN model can exponentially expand its receptive field, effectively integrating information from a wide temporal range. This unique feature not only facilitates accurate feature extraction from raw acoustic signals but also enhances the model's resilience against noise and variability. Considering the recent trends and novel keystroke acoustic attacks, this paper aims to highlight the potential of TCNs as a robust and adaptive countermeasure. In order to provide a fair experimental comparison with respect to the state of the art, we employed the same dataset as in [12, 17], as well as the same evaluation metrics. Experiments suggest that the TCN paradigm represents a promising avenue for advancing the field of cybersecurity and allowing reinforcement protection of sensitive information against unconventional threats.

The remainder of this paper is organized as follows: Sect. 2 provides a summary of previous research works related to the topic, the proposed pipeline (see Fig. 2) and the details of the developed deep learning model architecture as well as a description of experimental result is presented in Sect. 3. Section 4 presents the conclusions based on the results, which confirmed the highly promising performance of the designed solution.

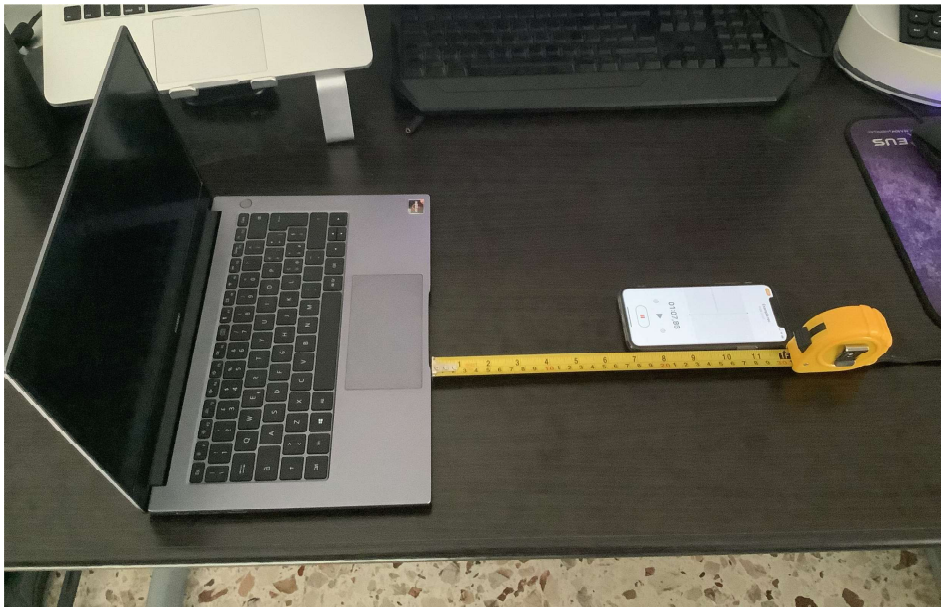

**Fig. 3.** Desk setup for recording keystrokes dataset

## 2 Proposed Method

The proposed work delves into a deep learning approach for attacks which exploit acoustic emissions produced during typing to infer sensitive information like keystrokes, based on the same dataset used in [12]. Differently than [12] our proposed solution is based on a Temporal Convolutional Network (TCN) model. The involved pipeline has been successfully validated in different contexts [16, 21–24]. Determining the position of a smartphone based on audio recordings of keystrokes could be solved with triangulation technique: modern smartphones are equipped with two different microphones at the bottom of the smartphone, so applying triangulation techniques it is possible to estimate the source location based on the differences in arrival times of the keystroke sound at each microphone [16]. By knowing the speed of sound in the medium (e.g., air), you can use these time differences to calculate the distance between the source of the signal and the different receivers. With multiple distance measurements from different pairs of receivers, it is possible triangulate the source's location [16]. We have conducted two different set of experiments to validate our proposed pipeline method. For the first experiment, the dataset has been downloaded from the github repository [17], provided by authors of [12]. This dataset has 36 wav audio files for the keystroke recorded via phone and 36 files recorded via Zoom. Each file has 25 keystroke peaks, which have been properly split in 25 single audio files, to isolate each keystroke peak, for a total of 900 audio files for the phone recording audio and 900 for the zoom recording audio.

After than we have applied a proper data augmentation technique, based on adding noise to the signal as reported in [21] creating other 1800 audio files. Note that in [12] authors augmented variability of the input applying just time-shifted random. Then specifically, MFCCs (Mel-frequency cepstral coefficients) features were used as input features for the deep learning model [12, 25, 26].

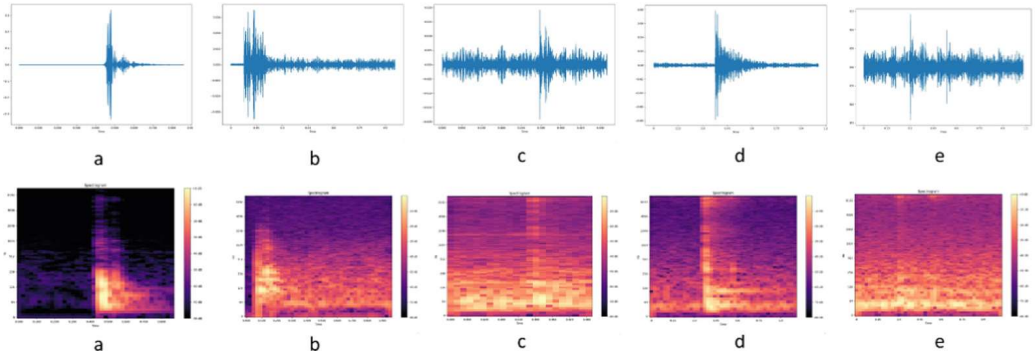


**Fig. 4.** (a) Audio recorded direct with a smartphone (iPhone13) on a Macbook Pro at a distance of 17 cm [12] (b) Audio recorded with a smartphone (iPhone13) via Zoom video conference tool on a Macbook Pro at a distance of 17 cm [12] (c) Audio recorded direct with a smartphone (iPhone X) on a Matebook at a distance of 17 cm on the wild (d) Audio recorded direct with a smartphone (iPhone X) on a Matebook at a distance of 50 cm on the wild (e) Audio recorded direct with a smartphone (iPhone X) on a Matebook at a distance of 100 cm on the wild

A second dataset has been collected using an iPhone X (see Fig. 3). To record audio file, it has been used the native iOS app "Voice Memos", setting the quality to Lossless which creates files in.m4a format. The laptop used is a Huawei Matebook D14 (2020).

The data collection took place in a room of approximately 14 $m^2$ and a height of approximately three meters. The environment is not technically soundproofed but being quite furnished, so it does not suffer from echoes. During the measurements the whole decibels in the room fluctuated between 46 and 50 db. The laptop is placed with the screen facing the smartphone which acts as a microphone positioned at 17, 50, 100 cm away from the computer. The smartphone which has two microphones at the bottom is then placed with the bottom facing the laptop. The measurements generated files in m4a format which were then converted into wav format following the following specifications:

- Audio codec: pcm_s16le.
- Audio bitrate: 320kbps.
- Audio channels: stereo (2.0).
- Sample rate: 48000 Hz.
- 36 audio files were generated for each data collection, each containing a letter or number keypress on the keyboard 25 times.
- Data collections were collected with different distances of the smartphone from the keyboard: 17, 50, 100 cm.

The second dataset has 8100 audio files at different distances. The key core of the implemented above pipeline is the TCN model, a type of convolutional neural network (CNN) architecture designed for processing sequential or time-series data. While traditional CNNs excel at spatial tasks such as image recognition, TCNs are specifically tailored for tasks that involve sequences, such as natural language processing, speech recognition, and time-series forecasting [19, 27, 28]. The key feature of a TCN is its ability to capture long-range dependencies in sequences, which is achieved by dilated convolutions. In a traditional convolutional layer, a small kernel moves across the input

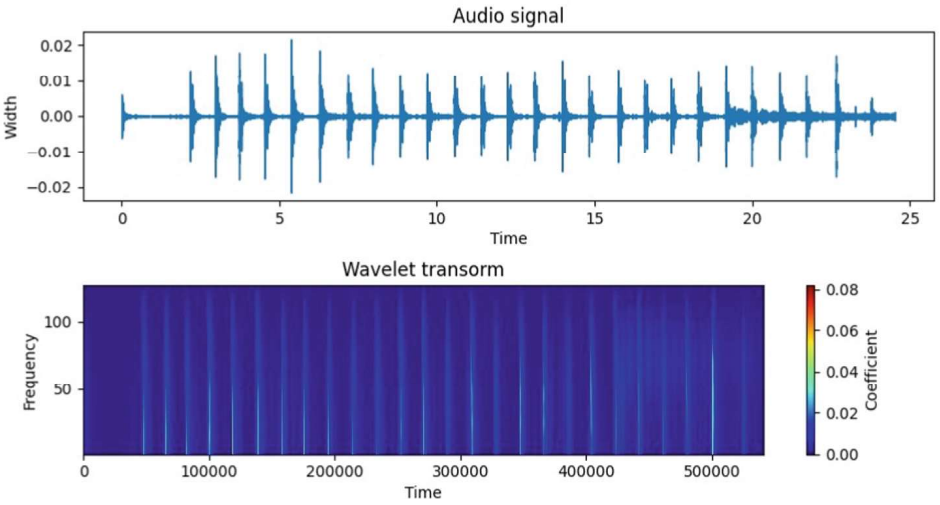


**Fig. 5.** The signal of the audio recorded in [12] direct with iPhone13 and the corresponding wavelet transform used for the segmentation and split of the keystroke's audio signals

data with a fixed stride, capturing local patterns. In contrast, dilated convolutions introduce gaps between the kernel's elements, allowing it to access a larger context at each layer. By stacking multiple dilated convolutional layers, TCNs can effectively capture dependencies across different time scales. TCN assumes a completely different approach to the problem of sequential data modeling. TCNs proved that convolutional networks could achieve better performance than RNNs in many tasks while avoiding the common drawbacks of recurrent models [28]. Moreover, using a TCN model instead of a recurrent one can led to performance improvements, as it allows parallel computation as in [29] or parallel CPU as in [30]. TCN processes 1D sequences of data by applying casual convolution filters along the time dimension. It means that the output sequence has the same length as the input one and each element in the output sequence depends on previous elements in the input sequence [28]. The proposed TCN method (see Fig. 1), has been implemented with convolution kernel $k = 3$ and padding $= k-1$. To calculate the casual convolution, we need to add padding from the left of the input tensor. Causal convolution has a simple logical sense: casual convolution collects previous sequence data and patterns. In fact, Deep Learning models that use casual convolution layers can extract dependencies that help predict future values. To implement casual convolution, we need to apply classical 1-D convolution with padding and crop elements from the right. The dilation technique with a casual convolutional layer increases the input time series coverage and reduces the computational costs significantly.

TCN assumes the sequence of casual convolutional layers with has a dilation equaled to $2^i$-1 (where i is the hidden layer number). For the proposed architecture TCN model has been applied the setup on Table 1. ReLU has been used as the activation function for the TCN. To normalize the input of hidden layers (which could propagate gradient problems), weight normalization is applied to every convolutional layer. The dropout regularization method is added after every convolutional layer.

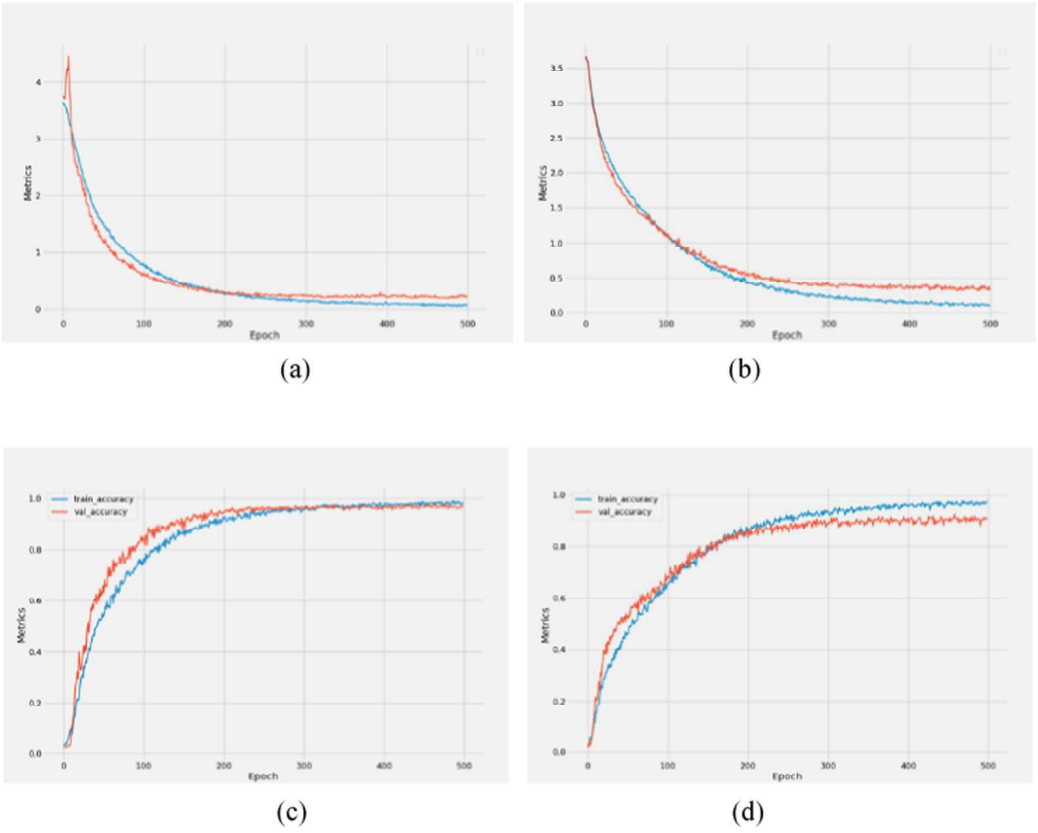


**Fig. 6.** The training and validation loss (a, b) and accuracy (c, d) over 500 epochs for iPhone13 direct recording keystrokes (a, c) and Zoom recording (b, d) [12]

## 3 Experimental Results

We aim to demonstrate the model's effectiveness in various acoustic scenarios to ensure its ability to generalize in real-world applications.

The aim of this work is to demonstrate the potential of the proposed pipeline for the task of acoustic attack on keyboards. We compared our pipeline with recent work proposed in [12] employing the same dataset as well as the same evaluation settings without any change. We also addressed the problem of overfitting properly, as previously detailed. Moreover, we have been collected a bunch of new experiments in "wild conditions".

Extensive experiments have been carried on validating the proposed TCN model implemented by using Tensorflow [31, 32]. All methods have been run on a PC with Intel(R) Core(TM) i7 CPU, 16 GB memory and NVIDIA RTX 2050 GPU [29, 30].

Every audio dataset file [12, 17] has been split isolating the keystrokes by recognizing audio peaks. Then, noise-based Data Augmentation has been applied. The amplitude of applied noise is in the range between $10^{-4}$ and $10^{-5}$ [10]. Then, for every audio file on the dataset, has been extracted the MFCC (Mel Frequency Cepstrum Coefficients) feature. The dataset has been split in training (80% of the total dataset), and validation (20%) subsets and processed during training and evaluation of the TCN model. We optimized the TCN model with Adam optimizer and cross entropy as loss function. After

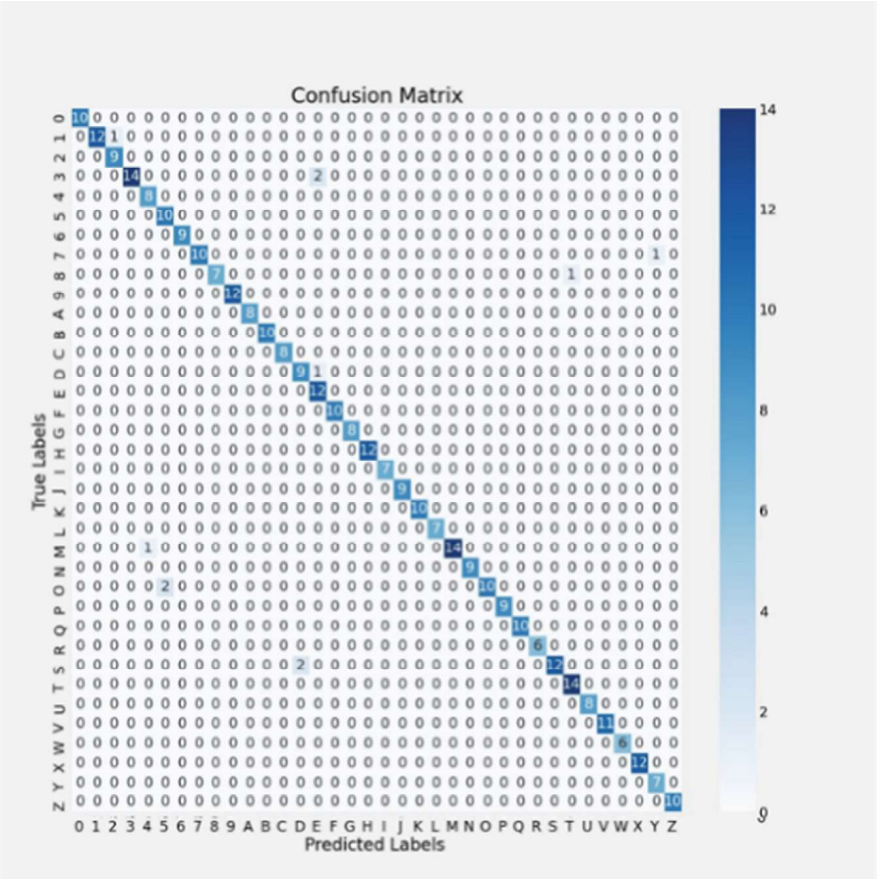


**Fig. 7.** The confusion matrix

an hyperparameter search performed on a subset of the data, the best setting for the model has been defined (see Table 1). Then, we performed training and test considering the above settings, achieving a validation accuracy of 98.3% for phone recording data and 93.05% for Zoom recording data, which is the highest accuracy ever seen with a deep learning model. Figure 4 shows the waveform and spectrogram of the keystroke audio signal, which let us introduce into the acoustic characteristics of typing the waveform displays distinct peaks corresponding to each keystroke, showcasing the temporal nature of the sound. Simultaneously, the spectrogram reveals the frequency composition over time, with varying intensity for different keystrokes. The sharp spikes (see Fig. 5) in both representations underline and the abruptness of individual key presses. This analysis not only captures the essence of typing sounds but also holds potential for applications in security and identification, as each keystroke manifests a unique auditory fingerprint.

Training cross entropy convergence to zero for both phone and Zoom recording are reported on the first row of Fig. 6, while the plots in second row of Fig. 6 show the converging training and validation accuracies, achieving a peak validation accuracy of 98.3% for phone recording data and 93.05% for Zoom recording data, which is the higher validation accuracy peak in state of the art for the problem of keystroke acoustic attack on keyboards (see Table 2).


**Fig. 8.** Misclassified keystroke proximity. Arrows connect the true keystroke with the corresponding misclassified one predicted by the model

**Table 2.** Overview of existing research papers and comparison with our work

| Method | Accuracy validation peak (%) |
|---|---|
| TCN (our) | 98.3 |
| Harrison et al. [12] | 95 |
| Anindya et al. [13] | 93.7 |
| Compagno et al. [15] | 91.7 |
| Bai et al. [8] | 91.2 |
| Abhishek et [9] | 74.3 |
| Zhu et al. [14] | 72.2 |

While the model demonstrates remarkable performance on the training data, the perfect convergence may indicate overfitting on the training set. In order to contrast overfitting, we have diligently pursued an in-depth examination of this phenomenon, prioritizing the model's generalization capability for predicting keystroke classes in unseen data, with minimal instances of misclassification. Indeed, off-diagonal elements are notably rare. This suggests the model's efficacy in distinguishing between different keystroke acoustic classes. Such precision is a testament to the model's capacity to capture subtle distinctions within the acoustic data, underscoring its potential in bolstering security measures. Through a series of comprehensive tests (100 audio files of unseen keystroke data extracted randomly), we consistently observed that in 99% of cases, the model accurately predicted the correct keystroke class, demonstrating its robustness. This encouraging result underscores the potential effectiveness of our approach, while ongoing efforts will focus on scaling the model's applicability across diverse datasets and real-world scenarios to ensure its reliability in practical cybersecurity contexts.

This striking congruence showcases the model's remarkable accuracy in capturing intricate patterns and nuances, suggesting a robust understanding of the data. Such coherence between predictions and reality substantiates the effectiveness of the methodology and lays a strong foundation for its potential applications in keystroke acoustic attack mitigation and broader security contexts. Figure 7 shows the confusion matrix which is

**Table 3.** Classification report

| Keystroke | Precision | Recall | F1-score | Misclassified keystroke |
|---|---|---|---|---|
| 1 | 1.00 | 0.92 | 0.96 | 2 |
| 3 | 1.00 | 0.88 | 0.93 | E |
| D | 0.82 | 0.90 | 0.86 | E |
| M | 1.00 | 0.93 | 0.97 | 4 |
| O | 1.00 | 0.83 | 0.91 | 5 |
| S | 1.00 | 0.86 | 0.92 | D |

**Table 4.** Experimental results report in wild conditions with a new different dataset, where audio has been recorded with an iPhone X direct to a matebook laptop at three different smartphone distances from the keyboard: 17, 50, 100 cm

| Distance (cm) | Peak accuracy (%) | Peak loss |
|---|---|---|
| 17 | 99.44 | 0.0426 |
| 50 | 98.88 | 0.0562 |
| 100 | 97.02 | 0.0633 |

truly exceptional in its clarity and insights. It vividly expresses the model's misclassification cases. Table 3 reports only the misclassified keystroke classes, specifying the wrongly predicted keystroke (rightmost column). We indeed observed that most of the misclassified keystrokes are in proximity with the true classes, as shown in Fig. 8. This suggest that the latent representation of the input signal defined by the model considers the physical distance between the source (i.e., keystroke) and the microphone, opening room for further investigations.

In Table 4 it is reported experimental results in wild conditions with the new dataset, where audio has been recorded at three different distances: 17, 50, 100 cm.

## 4 Conclusions and Future Works

The presented investigation of the Temporal Convolutional Network (TCN) methodology for keystroke acoustic attacks on keyboards has showcased promising strides in bolstering cybersecurity. The TCN's adeptness at capturing temporal intricacies within typing sounds has yielded remarkable results, with training and validation accuracy converging effectively. The model's ability to accurately predict keystroke classes on unseen data, as evident from the exhaustive tests, underscores its robustness and potential practical application. However, such study would benefit for a more accurate investigation and ablation studies focusing on specific samples for model explanation.

Thus, future works should focus on introducing regularization techniques to enhance the model's generalization capability. Moreover, expanding the dataset's diversity and

scale could further assess the model's reliability in real-world scenarios [16]. Future research should also address the model's response to varying noise levels and nuanced typing behaviors, ensuring its effectiveness in practical settings. Exploring ensemble methods or hybrid architectures could potentially enhance classification accuracy further. Exhaustive tests, underscores its robustness and potential practical application. This work highlighted how TCN are a promising approach for countering keystroke acoustic vulnerabilities, representing substantial potential for safeguarding sensitive information from emerging threats, as the cybersecurity attacks landscape continues to evolve leaning on novel AI-based approaches.

## References


1. Friedman, J.: Tempest: a signal problem. NSA Cryptologic Spectrum **35**, 76 (1972)
2. Halevi, T., Saxena, N.: Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios. Int. J. Inf. Secur. **14**(5), 443–456 (2015)
3. NSA NACSIM. "5000: Tempest Fundamentals". In: National Security Agency (1982)
4. Toreini, E., Randell, B., Hao, F.: An acoustic side channel attack on enigma. In: School of Computing Science Technical Report Series (2015)
5. Peter Wright. Spycatcher: the candid autobiography of a senior intelligence officer. New York: Viking (1987)
6. Zhuang, L., Zhou, F., Doug Tygar, J.: Keyboard acoustic emanations revisited. ACM Trans. Inf. Syst. Secur. (TISSEC) **13**(1), 1–26 (2009)
7. Asonov, D., Agrawal, R.: Keyboard acoustic emanations. In: IEEE Symposium on Security and Privacy, 2004. Proceedings, pp. 3–11. IEEE (2004)
8. Bai, J.X., Liu, B., Song, L.: I know your keyboard input: a robust keystroke eavesdropper based-on acoustic signals. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1239–1247 (2021)
9. Abhishek Anand, S., Saxena, N.: Keyboard emanations in remote voice calls: password leakage and noise (less) masking defenses. In: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, pp. 103–110 (2018)
10. Kim, G., Han, D.K., Ko, H.: SpecMix: a mixed sample data augmentation method for training with time-frequency domain features. Interspeech (2021)
11. Taheritajar, A., Harris, Z.M., Rahaeimehr, R.: A survey on acoustic side channel attacks on keyboards. ArXiv, abs/2309.11012 (2023)
12. Harrison, J., Toreini, E., Mehrnezhad,M.: A practical deep learning-based acoustic side channel attack on keyboards. In: 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, Netherlands, pp. 270–280 (2023)
13. Maiti, A., Armbruster, O., Jadliwala, M., He, J.: Smartwatch-based keystroke inference attacks and context-aware protection mechanisms. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (2016)
14. Zhu, T., Ma, Q., Zhang, S., Liu, Y.: Context-free attacks using keyboard acoustic emanations. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (2014)
15. Compagno, A., Conti, M., Lain, D., Tsudik, G.: Don't Skype & Type! acoustic eavesdropping in voice-over-IP. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (2016)
16. Bai, J., Liu, B., Song, L.: I know your keyboard input: a robust keystroke eavesdropper based-on acoustic signals. In: Proceedings of the 29th ACM International Conference on Multimedia (2021)

17. Harrison, J.: Keystroke-Datasets. https://github.com/JBFH-Dev/Keystroke-Datasets (2023)
18. Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: marrying convolution and attention for all data sizes. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
19. Dudukcu, H.V., Taskiran, M., Taskiran, Z.G., Yildirim, T.: Temporal convolutional networks with RNN approach for chaotic time series prediction. Appl. Soft Comput. **133**, 109945 (2023)
20. Spata, M.O., Battiato, S., Ortis, A., Rundo, F., Calabretta, M., Pino, C., Messina, A.A.: Deep learning algorithm for advanced level-3 inverse-modeling of silicon-carbide power MOSFET devices. Workshop on Electronics Communication Engineering (2023)
21. Park, D.S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E.D., Le, Q.V.: SpecAugment: a simple data augmentation method for automatic speech recognition. Interspeech (2019)
22. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
23. Huq, S., Xi, P., Goubran, R., Valdés, J.J., Knoefel, F., Green, J.: Data augmentation using reverb and noise in deep learning implementation of cough classification. In: IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6 (2023)
24. Siriwardena, Y.M., Attia, A.A., Sivaraman, G., Espy-Wilson, C.: Audio data augmentation for acoustic-to-articulatory speech inversion using bidirectional gated RNNs. arXiv:2205.13086
25. Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., Sporleder, C.: Acoustic side-channel attacks on printers. USENIX Security Symposium (2010)
26. Berger, Y., Wool, A., Yeredor, A.: Dictionary attacks using keyboard acoustic emanations. In: Proceedings of the 13th ACM conference on Computer and communications security, pp. 245–254 (2006)
27. Gridin, I.: Time series forecasting using deep learning: combining PyTorch, RNN, TCN, and deep neural network models to provide production-ready prediction solutions (2021)
28. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. ArXiv, abs/1803.01271 (2018)
29. Pelletier, C., Webb, G.I., Petitjean, F.: Temporal convolutional neural network for the classification of satellite image time series. ArXiv, abs/1811.10166 (2018)
30. Spata, M.O., Rinaudo, S.: Virtual machine migration through an intelligent mobile agents system for a cloud grid. J. Converg. Inf. Technol. **6**, 351–360 (2011)
31. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P.A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zhang, X.: TensorFlow: a system for large-scale machine learning. In: USENIX Symposium on Operating Systems Design and Implementation (2016)
32. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I.J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D.G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P.A., Vanhoucke, V., Vasudevan, V., Viégas, F.B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous distributed systems. ArXiv, abs/1603.04467 (2016)