SAFE: Spiking Neural Network-based Audio Fidelity Evaluation

Aaditya Khant aaditya.Khant@utsa.edu
Raveen Wijewickrama raveen.Wijewickrama@utsa.edu
Murtuza Jadliwala murtuza.jadliwala@utsa.edu
University of Texas at San Antonio, San Antonio, Texas, USA

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Recent advances in generative AI have enabled the creation of highly realistic synthetic audio, which poses significant challenges in voice authentication, media verification, and fraud detection. While Artificial Neural Networks (ANNs) are frequently used for fake audio detection, they often struggle to generalize to unseen and complex manipulations, particularly partial fake audio, where real and synthetic segments are seamlessly combined. This paper explores the use of Spiking Neural Networks (SNNs) for fake and partial fake audio detection – an unexplored area. Taking advantage of the inherent energy efficiency and temporal processing capabilities of SNNs, we propose novel SNN-based architectures for both tasks. We perform comprehensive evaluations that include hyperparameter tuning, cross-data set generalization, noise robustness, and partial fake audio detection using multiple large-scale public audio datasets. Our results show that SNNs achieve performance comparable to state-of-the-art ANN models while showing better generalization capabilities and robustness to noise. These SNN-based approaches also resulted in additional advantages, such as reduced model sizes and the ability to classify individual segments, making them more suitable for resource-constrained and real-time voice authentication applications. This work lays the foundation for exploring SNNs as countermeasures against audio spoofing in security-critical applications.

Keywords: Fake Audio Detection; Partial Fake Audio Detection; Spiking Neural Networks; Artificial Neural Networks.

1. Introduction

Generative AI continues to evolve at an unprecedented rate, enabling the creation of highly realistic synthetic media in various modalities, including images, video, and audio (Ramdurai and Adhithya, 2023). These technologies are proving to be highly useful in various domains such as entertainment, customer service, education, and healthcare (Ramdurai and Adhithya, 2023). However, the ease of generating convincing artificial content also introduces significant ethical, security, and societal challenges. While synthetic images and deepfake videos dominate media headlines (Rana et al., 2022), *synthetic speech* generated by advanced Text-to-Speech (TTS) (Shen et al., 2018; Ren et al., 2020) and Voice Conversion (VC) (Kameoka et al., 2018; Qian et al., 2019) systems is emerging as an equally disruptive force, particularly in areas where voice authenticity is essential, such as voice authentication systems, customer service, and media. These models can produce speech that is nearly indistinguishable from real voices (Mai et al., 2023), opening the door to malicious activities such as impersonation, fraud, and disinformation (Bleisch, 2024). Even more concerning is the rise of *partial fake audio*, where genuine and fabricated segments are seamlessly combined (Zhang et al., 2021), making detection significantly more challenging.

Current approaches to detect fake audio or synthetic speech were originally based on machine learning (ML) algorithms and have since evolved to incorporate advanced deep learning models (Dixit et al., 2023). The performance of these models also relies heavily on the quality of the training datasets. Among various options, the `ASVSpoof-2019` (Wang et al., 2020b) and `Fake or Real` (Reimao and Tzerpos, 2019) datasets are widely used benchmarks for fake audio detection. Certain models proposed in the literature, such as RawNet2 (Jung et al., 2020) and DeepSonar (Wang et al., 2020a) have shown considerable performance on these datasets, yet recent studies reveal that these models continue to struggle with generalization to audio content generated by newer/previously unseen TTS and VC techniques (Chen et al., 2020). Moreover, detection of *partial* fake audio poses an additional challenge, as existing models typically assume the input audio to be fully fake or real, leaving them ill equipped to handle such complex cases.

Spiking Neural Networks (SNNs), inspired by the brain's efficient, event-driven processing, present a compelling, yet largely untapped paradigm for addressing these limitations. Unlike traditional Artificial Neural Networks (ANNs), SNNs communicate via discrete spikes, offering a potentially more refined and efficient way to capture the critical temporal dynamics inherent in audio signals. Despite these inherent benefits, the application of SNNs to the specific problem of detecting fully and partially fake audio remains largely unaddressed.

Given that the application of SNNs in this area is largely unexplored, coupled with SNNs inherent advantages, a critical question arises: *Can SNNs offer a better approach?* Specifically, we ask: *How do SNN-based architectures perform in detecting fully synthetic speech, in terms of accuracy and robustness under diverse conditions, and how does this compare to state-of-the-art (SOTA) ANN methods?* Furthermore, how well do SNNs generalize to entirely new and unseen forms of synthetic speech, a known weakness of current ANN methods? And perhaps most crucially for real-world scenarios, *are SNNs particularly well-suited to the more challenging task of detecting partial fakes, where subtle temporal cues may be the key to success?* We believe that answering these questions and observing the strengths and limitations of SNNs in these tasks can help guide future work on enhancing voice authentication, media verification, and fraud detection systems, where reliable and efficient detection of audio manipulations is increasingly important. While SNN components are not novel in isolation, their application to fake/partial fake audio detection, especially with frame-level classification (i.e. classify short segments of audio independently, rather than treating the entire clip as a single unit), cross-dataset generalization, and noise robustness experiments is entirely novel and unexplored in existing literature. Our work attempts to fill this critical gap in the intersection of SNNs and their application to detect audio forgery.

To this end, we propose and explore three SNN architectural flavors: (i) a minimal feed-forward SNN that offers the lowest latency and parameter count, (ii) a convolutional CSNN that extracts local time-frequency cues, and (iii) a recurrent RSNN that captures longer temporal context. This range of designs lets us study the accuracy–latency trade-off inherent to spiking models. For a thorough evaluation of the proposed approach against current SOTA ANN-based approaches, we also curate a more comprehensive dataset by merging multiple mainstream datasets, ensuring broader coverage of real and synthetic audio. Our results show that the proposed SNN models performed comparably to ANN models for fake audio detection task but showed overall better generalizability and robustness to noise while maintaining a smaller model size. Specifically, our proposed models were able to achieve an Equal Error Rate (EER) as low as 4.79% on the proposed combined dataset for the fake audio detection task and a frame-level accuracy of 71.39% for the partial fake audio task. Specifically, our contributions are as follows.

- **Development and analysis of three specialized SNN architectures** for full and partial fake audio detection problem by leveraging spiking neurons' temporal encoding to capture subtle audio cues at frame-level.[1]
- **Comprehensive hyperparameter exploration** of SNN models, examining different surrogate gradients (*Fast Sigmoid*, *Arctangent*) and loss functions (*CE-count*, *CE-rate*), leading to optimized configurations.
- **Cross-dataset generalization and noise robustness evaluation**, comparing SNN performance to prominent ANN baselines (RawNet2, AASIST, ResNet, CNN, MLP) on known and unseen data, under varying noise levels.
- **Partial fake detection evaluation**, by demonstrating the effectiveness of SNNs in localizing synthetic segments within mostly genuine audio.

2. Related Work

Fake Audio Detection The pipeline for fake audio detection typically involves feature extraction from input audio and classification. Early approaches relied on handcrafted features such as Mel-Frequency Cepstral Coefficients (MFCCs), Constant-Q Cepstral Coefficients (CQCCs), and Linear Prediction Cepstral Coefficients (LPCCs) combined with machine learning classifiers such as Gaussian Mixture Models (GMMs) (Todisco et al., 2016). With the advent of deep learning, end-to-end models such as RawNet2 (Tak et al., 2021) emerged, learning features directly from raw audio and eliminating the need for explicit feature engineering. These deep learning models have achieved considerable success on benchmark datasets such as `ASVspoof-2019` and `Fake or Real`, as summarized in Table 1.

Despite these advancements, there is limited research involving cross-dataset evaluations for fake audio detection. Many SOTA models, while effective on the datasets they were trained on, struggle with generalizing to unseen TTS or VC models (Chen et al., 2020). This generalization problem is particularly concerning as synthetic audio generation technologies continue to evolve, producing increasingly realistic audio that can evade detection. Therefore, enhancing the generalization ability of detection models is critical to improving the security of voice-based systems. Given these challenges, this work investigates the potential of SNNs to address the generalization issue in fake audio detection, particularly in scenarios involving unseen TTS and VC models.

Partial Fake Audio Detection Partial fake audio consists of a mixture of fake and real utterances, making it particularly difficult for deep learning models to detect. Existing models in the literature, typically trained on datasets containing entirely fake and entirely

[1] The code is available at: `https://github.com/aadityakhant/SAFE`

Table 1: A summary of related works on fake and partial fake audio detection.

Type	Model	Dataset	Metric	Result(%)
Fake Audio	TCN (Khochare et al., 2021)	Fake or Real	Accuracy	80.00
	VGG-19 (Reimao and Tzerpos, 2021)			90.72
	STN (Khochare et al., 2021)			92.00
	MobileNet (Reimao and Tzerpos, 2021)			92.00
	AMSDF (Wu et al., 2024b)		EER	4.55
	ASVSpoof B1 (Wang et al., 2020b)	ASVSpoof-19	EER	9.57
	ASVSpoof B2 (Wang et al., 2020b)			8.09
	ASSERT (Lai et al., 2019)			6.70
	RawNet2 (Tak et al., 2021)			1.12
	AMSDF (Wu et al., 2024b)			0.16
	LCNN (Müller et al., 2022)	In-the-Wild	EER	35.14
	RawNet2 (Müller et al., 2022)			33.94
	AMSDF (Wu et al., 2024b)			9.50
Partial Fake Audio	CFPRF (Wu et al., 2024a)	PartialSpoof	EER	7.41
	STFT (Negroni et al., 2024)			6.16
	ResNet-1D (Cai and Li, 2024)			1.16
	ResNet-1D (Cai and Li, 2024)	ADD2023	EER	0.064
	STFT (Negroni et al., 2024)	HAD	EER	7.36
	CFPRF (Wu et al., 2024a)			0.08

real samples, struggle to identify the manipulated portions when genuine audio is present (Rahman et al., 2022). This limitation arises because most current models are designed for binary classification, and they lack the granularity to detect individual fake/real segments within a single audio file. Although time-variant DNN models (see table 1) with variable input/output lengths have shown promise, there remains a lack of open-source datasets featuring diverse partial fake attacks. The `PartialSpoof` dataset (Zhang et al., 2021), based on `ASVspoof-2019`, is currently the only publicly available dataset for this purpose.

Spiking Neural Networks Recently, SNNs have gained attention as a biologically inspired alternative to traditional ANNs due to their temporal dynamics and energy efficiency (Yamazaki et al., 2022). Due to the recurrent nature of spiking neurons, SNNs are well suited for handling temporal data and have been successfully applied to tasks such as sound localization and classification (Baek and Lee, 2024). Further, convolutional and residual SNNs have demonstrated strong performance in image processing, combining the strengths of ANNs and SNNs (Mozafari et al., 2019; Zhou et al., 2020; Kirkland et al., 2020; Hu et al., 2021). Although SNNs have proven effective in sound and image-related tasks, their application to fake or partial fake audio detection remains largely unexplored. This work aims to bridge this gap by systematically investigating their potential in this area.

3. Preliminaries

Popular representations of SNNs include the *Leaky Integrate-and-Fire (LIF)* model, the *Hodgkin-Huxley* model, and the *Spike Response* model, each of which captures distinct aspects of neuronal dynamics and behavior. Given its proven effectiveness for power-efficient deep learning (Rozenberg et al., 2019), our work uses the LIF model to implement a SNN.

Leaky Integrate and Fire Neuron The LIF neuron is a simplified model of a biological neuron, widely used in computational neuroscience to simulate the electrical activity of neurons in a network (Dayan and Abbott, 2001). The LIF neuron has a membrane potential $U(t)$ which increases with input $I(t)$ (synaptic current or stimulus) and decays with the membrane potential decay rate β. The neuron "fires" or generates a spike when the membrane potential reaches a certain threshold, following which the membrane potential is reset according to some reset mechanism. The membrane potential of a neuron can be described by the following equation:

$$U(t+1) = \beta \times U(t) + I(t+1) - R(\beta \times U(t) + I(t+1)) \tag{1}$$

where R is the reset mechanism. R is set to 1 when the neuron fires, and 0 otherwise.

Surrogate Gradient Descent Training SNNs through supervised learning is challenging due to the discrete nature of spikes. During the forward pass, the spikes are represented using a *shifted Heaviside step function*. During the backward pass, to calculate the gradients (the partial derivative of the loss with respect to the parameters), the spikes are approximated using a smooth surrogate function such as *Fast Sigmoid* (Zenke and Ganguli, 2018) and *Arctangent* (Fang et al., 2021).

Loss Functions To train SNN models for classification tasks, two commonly used loss functions for backpropagation are *Cross Entropy Spike Count (CE-count)* and *Cross Entropy Rate (CE-rate)*. The `CE-count` loss function first predicts class by accumulating output spikes over all time steps and then calculates loss by calculating the cross-entropy between predicted class and target class. On the other hand, the `CE-rate` loss function processes spike outputs sequentially at each time step. At each time step, the spike output and the corresponding ground-truth values are passed through the cross-entropy (CE) function, with the resulting losses accumulated over time. Both loss functions promote consistent spiking of the correct class and suppresses incorrect spikes. More details on both loss functions are provided in the Appendix A.

4. Methodology

The proposed SNN based approaches begins by segmenting the input audio signal into smaller, overlapping chunks using a sliding window. Each chunk is then converted into a compact feature representation suitable for modeling audio patterns. These features are processed sequentially by a SNN, which classifies each chunk individually while maintaining temporal continuity through its internal membrane potential. This allows the model to capture local patterns as well as broader temporal dependencies across the input. Figure 1 provides a high-level overview of the proposed approach. Below we outline key aspects of our SNN-based methodology, including feature extraction and classification model architecture.

4.1. Feature Extraction

Suppose, input is an audio file sampled at 16 kHz and of 2 seconds in length, resulting in 32,000 floating point values per sample. Feeding these raw values directly into a neural network would dramatically increase the number of input parameters, resulting in a computationally inefficient model with higher memory and processing demands. To mitigate this, we extract a feature vector of 40 MFCCs (Davis and Mermelstein, 1980) and pass that as input to the models. To keep the input length constant, samples shorter than the

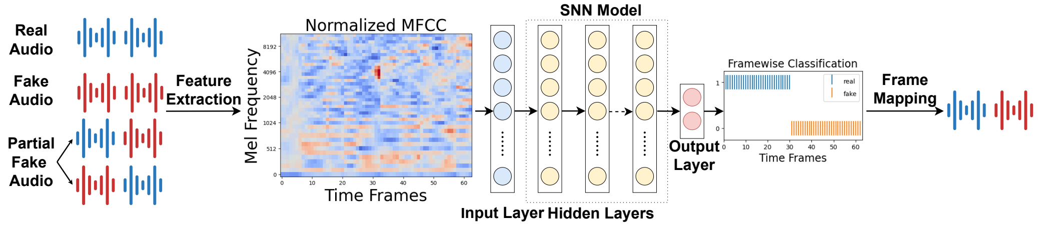

Figure 1: The proposed SNN-based approach for fake and partial fake audio detection.

desired length are padded with zeros (silence) and samples longer than desired length are truncated. For full fake audio detection, we use a window size of 2048 samples with a hop size of 512 samples (25% overlap), resulting in frames of 128 ms. For partial fake detection, we use shorter windows of either 320, 1280 or 5120 samples with no overlap, aligning the chunk size to frame-level ground truth labels. Lastly, MFCCs are normalized using *Lp-norm* normalization, preventing those with larger scales from disproportionately influencing the learning process.

4.2. SNN Models

SNNs are well-suited for sequential data due to their ability to naturally capture temporal dependencies through membrane potential dynamics of LIF neurons, as discussed in section 3. This built-in recurrence allows SNNs to process each input frame in sequence while retaining contextual information from previous frames without requiring explicit recurrent layers or skip connections. In contrast, conventional ANNs typically operate on fixed-length inputs in a single forward pass. We design and evaluate the below three distinct SNN-based architectures to evaluate trade-offs in complexity, efficiency, and performance.

SNN A feed-forward SNN model consisting of an input layer with 40 neurons, followed by four spiking layers containing 256, 126, 10, and 2 (output) neurons, respectively. Each spiking layer comprises a Fully Connected (FC) layer from an ANN model and a corresponding leaky layer. The leaky layer consists of LIF neurons, which are connected one-to-one with the neurons in the preceding FC layer, similar to the ReLU activation function in ANN models. The leaky layer serves as an activation mechanism, outputting either a spike (1) or no spike (0). We set the decay parameter (β) for LIF neurons to 0.9, and the spike threshold is learned during training. The input is processed sequentially, with the 40 MFCCs from one timeframe passed into the network at a time. This sequential input allows the SNN model to capture temporal dependencies, making it independent of input length. For fake audio detection, classification is done based on the spike count of two output neurons. In contrast, for partial fake detection, the model produces a prediction for each time frame by evaluating the spiking activity at that specific step.

CSNN While deep learning models such as Transformer-encoder are computationally and power intensive, simpler models such as MLP may lack the complexity needed to effectively capture subtle patterns in large audio datasets (Müller et al., 2022). CNNs strike a balance by efficiently extracting complex features through convolutional layers while utilizing smaller fully connected layers for classification. On the other hand, while SNNs are generally energy-

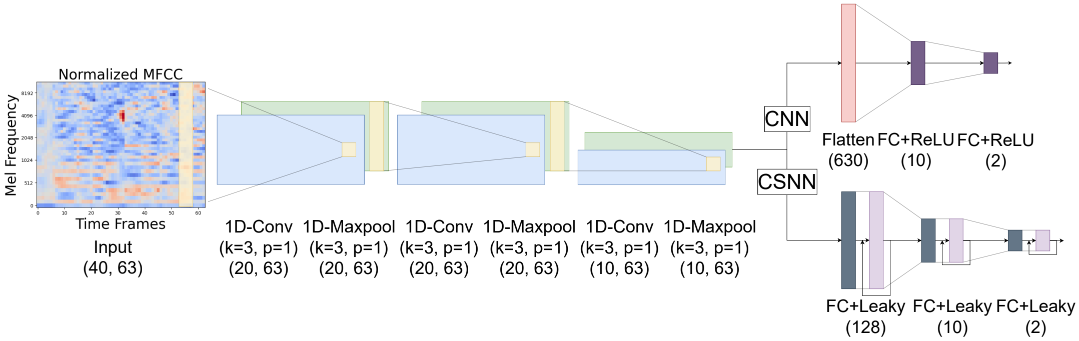

Figure 2: CSNN vs. CNN model architecture.

efficient, they may also lack complexity to fit diverse datasets. To this end, we propose a novel Convolutional Spiking Neural Network (CSNN) based approach that combines the feature extraction power of CNNs with the temporal processing capabilities of SNNs for the task of fake and partial fake audio detection. As shown in fig. 2, CSNN retains the CNN architecture up to the final maxpool layer, where deep features are extracted from the MFCCs. These deep features are then passed through three spiking layers containing 128, 10, and 2 neurons, respectively. Similarly to the previous SNN model, we set the decay parameter (β) to 0.9, and the spike threshold is learned during training. Fake and partial fake audios are classified using a mechanism similar to the SNN model.

RSNN Typically, in a spiking neuron, membrane potentials decay over time, leading to rapid loss of historical information. This limitation is particularly problematic for audio classification tasks because such tasks often involve sequential data in which meaningful patterns are distributed over time. To address this, we draw inspiration from the ResNet18 (He et al., 2016) architecture to implement a Residual SNN (RSNN). As shown in fig. 8 in the Appendix B, the residual block in the RSNN model is visually similar to that of a ResNet18, with three main differences: (1) instead of a standard batch normalization layer, we use Batch Normalization Through Time (BNTT) (Kim and Panda, 2021) to accommodate temporal spiking behavior; (2) we replace the ReLU activation layers with spiking neuron layers. Consequently, the RSNN model represents an advanced deep spiking network architecture for fake and partial fake audio classification; and (3) we substitute 2D convolution with 1D convolution to accommodate the single dimensional audio data.

4.3. Artificial Neural Network Models

Due to the lack of cross-dataset evaluation studies on assessing the generalization ability of ANNs for fake audio detection problem, we implemented five representative ANN models, MLP, CNN, RNN, RawNet2 and AASIST. This aims to establish baseline performance by ANN models that utilize similar resources, including datasets.

Multi Layer Perceptron We implement a 5-layer *Fully-Connected Feed-Forward Neural Network (FC-FFNN)*. The layers contain 2520 (input), 256, 128, 10, and 2 (output) neurons, respectively. A Rectified Linear Unit (ReLU) activation function is used at each hidden layer to introduce non-linearity allowing the model to learn complex patterns. The input to the network consists of flattened MFCCs of size $40 \times 65 = 2520$.

Convolutional Neural Network We then implement a CNN model as demonstrated in fig. 2. The convolution layer in the model uses 1D convolutional and 1D max-pooling, applied over the time domain, to extract deep features from the MFCCs of input audio. 40 MFCCs are treated as 40 input channels. The subsequent convolutional layers have 20, 20, and 10 channels, respectively. In both the convolutional and max-pooling layers, a stride of one and padding (zero-padding) of one is used to preserve the temporal dimensions of the data. The output of the final max-pool layer is flattened into a 630-dimensional vector and fed into an FC-FFNN consisting of three layers with 630, 10, and 2 neurons, respectively, where the ReLU activation function is applied to the intermediate layers.

Residual Neural Network We also implemented the ResNet-18 architecture proposed by He et al. (2016) (illustrated in fig. 8). The model begins with an initial convolutional layer followed by batch normalization, a ReLU activation, and a max pooling layer. This is followed by eight groups of residual blocks. Each residual block contains two consecutive convolutional layers, each followed by batch normalization, with ReLU activation applied after the first batch normalization. A skip connection with down-sampling or identity mapping adds the block's input directly to its output. The last residual block is followed by an average pooling layer and a fully connected layer with two (output) neurons.

RawNet2 and AASIST We reimplement the RawNet2 architecture (Tak et al., 2021), originally proposed for the ASVSpoof-2019 challenge (Wang et al., 2020b), due to its strong performance on that dataset. We also reimplement the *Additive Angular Softmax-based Integrated Spectral and Temporal (AASIST)* model (Jung et al., 2022), which achieved promising results on the ASVSpoof-2019 dataset. The only modification made to both architectures is adjusting the input dimensions to accommodate the desired input length. This change does not affect the core architecture or feature extraction processes, as both models are designed to seamlessly adapt to varying input sizes.

4.4. Datasets

To conduct a rigorous evaluation, we created the *Consolidated Fake Audio* (`CFA`) dataset by merging the training, validation, and testing sets of the ASVspoof-2019 LA subset (`ASVspoof`) (Wang et al., 2020b), the Fake or Real - Normalized subset (`FoR`) (Reimao and Tzerpos, 2019), and `LibriSpeech` (Panayotov et al., 2015) into corresponding splits for the `CFA` dataset. By combining these sources, we increase both the diversity and quantity of real and synthetic audio samples, capturing a wider range of acoustic characteristics and manipulation techniques. Thus, the `CFA` dataset ensures a varied training environment, which is critical for fair comparisons between the ANN and SNN models in standard fake audio detection scenarios. Table 2 summarizes the real/fake sample distribution.

In addition to full fake audio, we also evaluate the ability of SNNs to detect *partial* fake audio. To this end, we employ the `PartialSpoof` dataset (Zhang et al., 2022), which was generated using real and fake speech samples from the `ASVSpoof` dataset. This dataset provides ground-truth annotations at multiple granularity levels, allowing both coarse- and fine-grained evaluation.

Table 2: Class distribution in the datasets.

Dataset	Train		Validation		Test	
	Fake	Real	Fake	Real	Fake	Real
FoR-normalized (Reimao and Tzerpos, 2019)	26,927	26,941	5,398	5,400	2,370	2,264
ASVspoof-2019 LA subset (Wang et al., 2020b)	22,800	2,580	22,296	2,548	63,882	7,355
LibriSpeech(Panayotov et al., 2015)	-	104,014	-	2,703	-	2,620
In-the-Wild(Müller et al., 2022)	-	-	-	-	11,816	19,963
CFA (proposed)	49,727	49,521	27,694	27,651	66,252	66,239
PartialSpoof(Zhang et al., 2022)	25,380		24,844		71,237	

5. Experiments & Results

This section outlines the conducted experiments and results to evaluate the effectiveness of SNNs in detecting fake and partial fake audio. In terms of performance metrics, we primarily use *EER* (a lower EER indicates better performance). For partial fake audio detection, we use *frame-wise accuracy*, where we compare each frame's prediction to its ground-truth label (instead of classifying the whole audio sample as fake/real). All models were trained for 200 epochs using the Adam optimizer with early stopping (patience of 10 epochs). We used grid search to tune key hyperparameters: learning rate (ranging from 0.0001 to 0.01), L2 regularization (weight decay, from 0 to 0.0001), and batch size (ranging from 32 to 128). All experiments were conducted on an NVIDIA L40S GPU.

5.1. Hyperparameter Tuning

We began by identifying optimal hyperparameters for our proposed SNN models. Specifically, we explored two loss functions (CE-rate and CE-count) and two surrogate gradients (Fast-Sigmoid and Arctangent). For each of these combinations, we trained SNN, CSNN, and RSNN models on the `CFA` dataset using a fixed input length of two seconds. The input length was selected based on the length distribution in the `CFA` dataset. The models were then evaluated on a validation set to select the best performing hyperparameters.

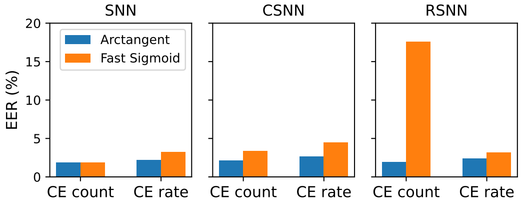

Figure 3: SNN, CSNN & RSNN hyperparameter tuning.

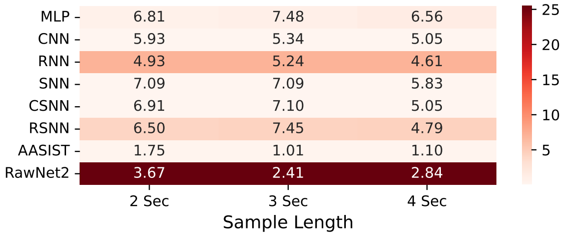

Figure 4: Heatmap of parameter counts (in millions) and sample lengths, annotated with EER(%).

Figure 3 illustrates the impact of these combinations on SNN, CSNN, and RSNN performance. Although changes in EER were relatively small, all three models performed slightly better with a combination of the *Arctangent* surrogate gradient and *CE-count* loss function. We speculate that *CE-count* is more suitable for binary (fake vs. real) classification, whereas *CE-rate* encourages correct classification at each timeframe. Very poor performance of RSNN model when using *CE-count* and *Fast-Sigmoid* is an outlier. For each model (SNN, CSNN, RSNN), the hyperparameter combination (*CE-count* and *Arctangent*) yielding the lowest EER on the validation set was selected for subsequent experiments.

Across all three SNN models, the best-performing configuration consistently used the *Arctangent* surrogate gradient and *CE-count* loss.

5.2. Fake Audio Detection

Effect of Sample Length After determining the best hyperparameters, we compared SNN models against standard ANN models (described in section 4.3) at fixed audio sample input lengths of 2, 3, and 4 seconds. These lengths were chosen based on the distribution of audio clip durations in the `CFA` dataset. Figure 4 shows the EER on the `CFA` test set and the parameter count for each model under these three input lengths. Deep neural network models such as RNN, RawNet2 and AASIST, even at 2 seconds input length, have significantly more parameters compared to SNN models. Although performance did not vary significantly with input length, the number of parameters in ANN models increased substantially (except for RawNet2 & AASIST) as input sample length grew (to accommodate higher input dimensions). In contrast, the SNN architectures noticeably remained constant in parameter count, indicating that SNNs can scale more efficiently to longer input clips without a proportional increase in model complexity and still retain competitive accuracy. See Appendix C for detailed model parameter counts.

In summary, SNNs maintain stable parameter counts across different input lengths while preserving accuracy, making them appealing for memory-efficient real-time fake audio detection systems.

Generalization The objective of this experiment was to evaluate the generalization capabilities of SNN in comparison with ANN models for the fake audio detection task. All models (SNN and ANN) were trained on the `CFA` dataset, which included the training sets from `FoR` and `ASVSpoof`. For testing, the models were evaluated on the test sets of these two familiar datasets (`FoR` and `ASVSpoof`) as well as the previously unseen `In-the-Wild` dataset which contains more real-world audio samples (see table 2). To ensure consistency, the input length was fixed at 2 seconds. Figure 5 presents the resulting EER values. As expected,

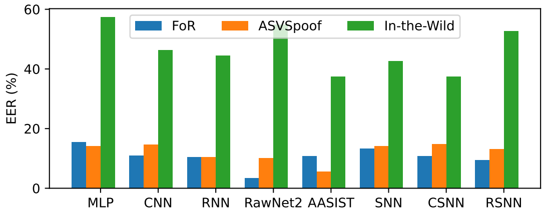

Figure 5: Model performance on known (`FoR` and `ASVSpoof`) vs. unknown (`In-the-Wild`) datasets.

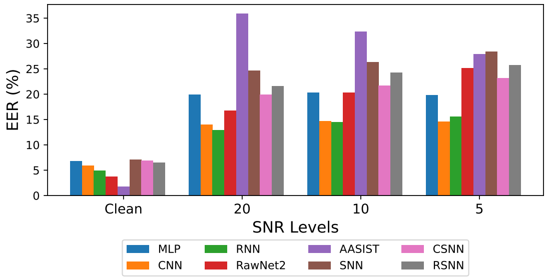

Figure 6: Model performance vs. noise levels.

EER rose significantly for all models when tested on unfamiliar distributions. Nevertheless, SNN, CSNN and RSNN achieved EERs of 42.66%, 37.53% and 52.64%, respectively, on the `In-the-Wild` dataset, which were comparable to or lower than those of the ANN models.

SNN-based detectors generalize at least as well as, and in the case of CSNN markedly better than, heavyweight ANN baselines on unseen generation methods.

Robustness Since random noise is a common adversarial or environmental challenge, we assessed model robustness by adding normalized noise to the audio samples at varying signal-to-noise ratio (SNR) levels of 20, 10, and 5. The SNN and ANN models were then compared at each noise level to quantify performance degradation and assess whether the temporal encoding in SNNs confers greater resilience to noise. Figure 6 shows that advanced architectures such as RawNet2 and AASIST, despite strong performance on clean data, degrade more severely under noise. For instance, AASIST's EER jumps from 1.75% (clean) to 35.90% at 5% noise whereas, CSNN's EER rise from 6.91% to only 19.87%. CNN and RNN models consistently achieve the lowest EER values across all noise levels. The simpler convolutional filters in these models are inherently robust to uniform additive noise, as they focus on local features that remain salient. Similarly, the temporal encoding in SNNs can preserve relevant spiking activity while discarding noisy fluctuations, leading to less pronounced performance drops.

In summary, SNNs show stronger robustness to noise compared to some advanced ANN models, highlighting their potential as a practical and resilient solution for real-world fake audio detection even in noisy and unpredictable conditions.

5.3. Partial Fake Audio Detection

Partial fake audio poses a more significant detection challenge because it blends genuine speech with fabricated segments. To highlight this challenge, we first tested previously trained SNN and ANN models (trained on `CFA` using fully fake/real samples only) against the `PartialSpoof` dataset. For this test, samples in the `PartialSpoof` dataset containing at least one synthetic segment were labeled as fake. Figure 7 shows that while all models struggled to distinguish partially faked samples from real audio, RawNet2 and AASIST showed the worst performance among all models. Among the tested models, MLP and RSNN performed the best, obtaining EER of 22.01% and 23.93%, respectively.

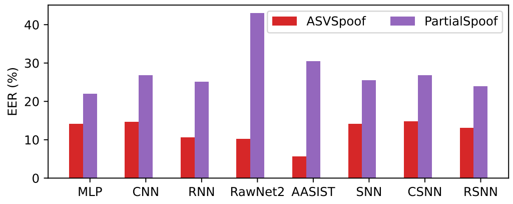

Figure 7: Model performances on full fake dataset (`ASVSpoof`) vs. partial fake dataset (`PartialSpoof`).

Table 3: Model performances on the `PartialSpoof` dataset at different frame lengths.

Model	Frame Length (ms)		
	20	**80**	**320**
SNN	68.99%	64.13%	65.86%
CSNN	71.45%	69.06%	69.41%
RSNN	71.39%	68.74%	70.32%

To further explore the temporal and recurrent properties of SNNs for detecting such fine-grained manipulations, we then performed frame-level classification by training the proposed SNN, CSNN, and RSNN models on the `PartialSpoof` dataset with the CE-rate loss. To further improve SNNs' performance, we incorporated population coding (Eshraghian et al., 2023) by changing the number of output neurons from two to ten, where each class is represented by five output neurons instead of one. For each timeframe, spikes from five output neurons are accumulated for classification. ANN models were excluded from this experiment, as they do not provide classification at the individual timeframe level. Table 3 compares frame-level accuracy for these models trained on the `PartialSpoof` dataset with non-overlapping sliding windows of length 20 ms, 80 ms, 320 ms. The slight increase in accuracy across all three models with shorter frame lengths is attributed to more granular frames, which offer a better distinction between real and fake segments.

All three SNN variants saw substantial gains when trained directly on partial fakes, with CSNN yielding the highest frame-level accuracy at 71.45%.

6. Discussion & Conclusion

Table 4: Key practical advantages of CSNN over a representative ANN baseline (RawNet2).

Aspect	**ANN Baseline**	**SNN**	**p-value**	**Benefit**
Unseen-attack degradation ΔEER	+51.53	+**26.8**	< 0.001	1.9× better generalisation
Robustness to noise ΔEER	+21.5	**+16.24**	< 0.0015	0.75× more robust
Partial-fake detection EER	42.96%	**26.79**%	< 0.001	0.62×better partial fake detection
Parameter count (millions)	25	**0.006**	−−	416× lighter, scalable
Frame-level classification	✗	✓	−−	partial-fake localization

In this work, we explored the novel application of SNNs to the increasingly critical task of detecting both fully and partially fake audio. As shown in table 4, our comprehensive comparative evaluations revealed that SNN-based architectures are not only viable but also effective competitors to SOTA ANNs in this domain. Specifically, SNNs achieved comparable performance in detecting fully synthetic audio and, in several scenarios, demonstrated superior robustness and efficiency. Moreover, when trained on partially fake audio, SNNs significantly improved their ability to detect small, localized manipulations. This highlights the practical benefit of their inherent temporal encoding and capacity for frame-level classification of audio segments. Our results indicate that SNNs offer a distinctive combination of robustness, efficiency, and temporal resolution, making them well-suited for real-world fake audio detection systems that require adaptability and low computational overhead.

Although we did not implement the proposed models on neuromorphic hardware or measure hardware-level energy usage, the sparse activation and the inherently event-driven nature of SNNs present an attractive avenue for energy-efficient deployments, for instance on Intel Loihi (Davies et al., 2018). Such hardware platforms may further amplify SNNs' advantages for large-scale or continuous fake audio monitoring. Despite these promising findings, our study underscores two pressing limitations in the fake audio detection domain. First, the limited diversity in existing training datasets impedes broader generalization to novel or unseen voice synthesis algorithms. Second, the shortage of sophisticated partial fake audio datasets with frame level labels and realistic manipulations restricts progress in partial fake audio detection. This work lays the foundation for future research aimed at enhancing the robustness and generalization of SNN models, particularly for security-critical audio manipulation detection.

Acknowledgments

Research reported in this publication was supported by the US National Science Foundation (NSF) under award number 2332744.

References

Suwhan Baek and Jaewon Lee. Snn and sound: a comprehensive review of spiking neural networks in sound. *Biomedical Engineering Letters*, 2024.

N. David Bleisch. Deepfakes and American Elections. `https://www.americanbar.org/groups/public_interest/election_law/american-democracy/resources/deepfakes-american-elections`, 2024. [Online; accessed 15-Aug-2024].

Zexin Cai and Ming Li. Integrating frame-level boundary detection and deepfake detection for locating manipulated regions in partially spoofed audio forgery attacks. *Computer Speech & Language*, 2024.

Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Odyssey*, 2020.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 2018.

Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans Acoust*, 1980.

Peter Dayan and L.F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, 2001.

Abhishek Dixit, Nirmal Kaur, and Staffy Kingra. Review of audio deepfake detection techniques: Issues and prospects. *Expert Systems*, 2023.

Jason K Eshraghian, Max Ward, Emre O Neftci, , et al. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 2023.

Wei Fang, Zhaofei Yu, Yanqi Chen, et al. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *IEEE/CVF*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF*, 2016.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE TNNLS*, 2021.

Jee-weon Jung, Seung-bin Kim, Hye-jin Shim, et al. Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms. *arXiv:2004.00526*, 2020.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, et al. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *IEEE ICASSP*, 2022.

Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, et al. Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *2018 SLT*, 2018.

Janavi Khochare, Chaitali Joshi, Bakul Yenarkar, Shraddha Suratkar, and Faruk Kazi. A deep learning framework for audio deepfake detection. *AJSE*, 2021.

Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in neuroscience*, 2021.

Paul Kirkland, Gaetano Di Caterina, John Soraghan, and George Matich. Spikeseg: Spiking segmentation via stdp saliency mapping. In *IEEE IJCNN*, 2020.

Cheng-I Lai, Nanxin Chen, Jesús Villalba, and Najim Dehak. Assert: Anti-spoofing with squeeze-excitation and residual networks. *arXiv:1904.01120*, 2019.

Kimberly T Mai, Sergi Bray, Toby Davies, and Lewis D Griffin. Warning: Humans cannot reliably detect speech deepfakes. *Plos one*, 2023.

Milad Mozafari, Mohammad Ganjtabesh, Abbas Nowzari-Dalini, et al. Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks. *Pattern recognition*, 2019.

Nicolas M Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? *arXiv:2203.16263*, 2022.

Viola Negroni, Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Analyzing the impact of splicing artifacts in partially fake speech signals. *arXiv:2408.13784*, 2024.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *IEEE ICASSP*, 2015.

Kaizhi Qian, Yang Zhang, Shiyu Chang, et al. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *ICML*, 2019.

Md Hafizur Rahman, Martin Graciarena, Diego Castan, et al. Detecting synthetic speech manipulation in real audio recordings. In *IEEE WIFS*, 2022.

Balagopal Ramdurai and Prasanna Adhithya. The impact, advancements and applications of generative ai. *SSRG-IJCSE*, 2023.

Md Shohel Rana, Mohammad Nur Nobi, Beddhu Murali, and Andrew H Sung. Deepfake detection: A systematic literature review. *IEEE Access*, 2022.

Ricardo Reimao and Vassilios Tzerpos. For: A dataset for synthetic speech detection. In *IEEE SpeD*, 2019.

Ricardo Reimao and Vassilios Tzerpos. Synthetic speech detection using neural networks. In *IEEE SpeD*, 2021.

Yi Ren, Chenxu Hu, Xu Tan, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv:2006.04558*, 2020.

MJ Rozenberg, O Schneegans, and P Stoliar. An ultra-compact leaky-integrate-and-fire model for building spiking neural networks. *Scientific reports*, 2019.

Jonathan Shen, Ruoming Pang, Ron J Weiss, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *IEEE ICASSP*, 2018.

Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2. In *IEEE ICASSP*, 2021.

Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients. In *Odyssey*, 2016.

Run Wang, Felix Juefei-Xu, Yihao Huang, et al. Deepsonar: Towards effective and robust detection of ai-synthesized fake voices. In *ACM MM*, 2020a.

Xin Wang, Junichi Yamagishi, Massimiliano Todisco, et al. Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 2020b.

Junyan Wu, Wei Lu, Xiangyang Luo, et al. Coarse-to-fine proposal refinement framework for audio temporal forgery detection and localization. In *ACM Multimedia*, 2024a.

Junyan Wu, Qilin Yin, Ziqi Sheng, et al. Audio multi-view spoofing detection framework based on audio-text-emotion correlations. *IEEE TIFS*, 2024b.

Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain Sciences*, 2022.

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 2018.

Lin Zhang, Xin Wang, Erica Cooper, Junichi Yamagishi, Jose Patino, and Nicholas Evans. An initial investigation for detecting partially spoofed audio. *arXiv:2104.02518*, 2021.

Lin Zhang, Xin Wang, Erica Cooper, et al. The partialspoof database and countermeasures for the detection of short fake speech segments embedded in an utterance. *IEEE/ACM TASLP*, 2022.

Qian Zhou, Yan Shi, Zhenghua Xu, et al. Classifying melanoma skin lesions using convolutional spiking neural networks with unsupervised stdp learning rule. *IEEE Access*, 2020.

Appendix A. Loss Functions

Given an audio sample with n number of time frames and target class y, the model predicts class $\hat{y}$ using $\hat{y} = \arg\max_{c \in \{0,1\}} \sum_{i=1}^{n} \hat{y}_i^{(c)}$, where $\hat{y}_i^{(c)}$ is the predicted probability of class c for the i^{th} time frame. Then $\mathcal{L}_{\text{CE-count}}$ is calculated using eq. (2).

$$\mathcal{L}_{\text{CE-count}} = -\left[y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right] \tag{2}$$

For the same sample, $\mathcal{L}_{\text{CE-rate}}$ is calculated using eq. (3), where $y_i = y$ for $i = 1, 2, ..., n$ and class prediction for the i^{th} frame $\hat{y}_i = \arg\max_{c \in \{0,1\}} \hat{y}_i^{(c)}$. Equation (3) can also be utilized to calculate loss for partial fake audio, where $y_i \in \{0,1\}$ is the target class for the i^{th} frame.

$$\mathcal{L}_{\text{CE-rate}} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{3}$$

Appendix B. RSNN vs RNN Model Architecture

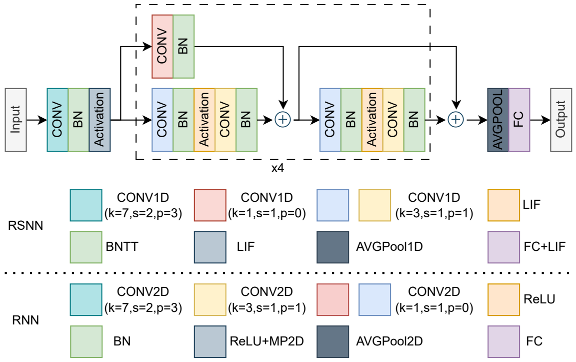

Figure 8: RSNN vs. RNN model architecture.

Appendix C. Input Length vs Parameter Counts

Table 5 reports the parameter counts for each model with input lengths 2-4 s, and floating point operations (FLOPs) for 2 s. The FLOPs include both the cost of converting raw audio into normalized MFCC features (feature extraction) and the cost of classification. For RawNet2 and AASIST, the feature extraction FLOPs are zero since they directly process raw audio. For all other models, the feature extraction cost is fixed at 17.31 millions FLOPs.

Table 5: Parameter counts for models at input lengths of 2-4 s. FLOPs reported for 2 s.

	MLP	CNN	RNN	RawNet2	AASIST	SNN	CSNN	RSNN
2 Sec	679,584	9,672	6,976,962	25,433,602	297,866	44,708	6,063	4,137,740
3 Sec	997,024	12,772	6,976,962	25,433,602	297,866	44,708	6,063	4,286,540
4 Sec	1,324,704	15,972	6,976,962	25,433,602	297,866	44,708	6,063	4,440,410
FLOPs (in millions)	18.67	17.32	2249.86	490.60	8877.44	22.97	17.66	27.36