SJER Scientific Journal of Engineering Research

futuristech PT. Teknologi Futuristik Indonesia

Date of publication December 9, 2025, date of current version December 9, 2025.
e-ISSN: 3109-1725
Digital Object Identifier 10.64539/sjer.v1i4.2025.349

Article

Incremental Development of a Framework for Mitigating Adversarial Attacks on CNN Models

Maaz Nisar[1], Nabeel Fayyaz[1], Muhammad Abdullah Ahmed[2], Muhammad Usman Shams[3], Bushra Fareed[3,*]

[1] Department of Computer Science and Information Technology, Khwaja Fareed University of Engineering and Information Technology, Pakistan; maaznisar363@gmail.com; nabeelfayyaz88@gmail.com
[2] Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan; ahmadmuhammadabdullah5@gmail.com
[3] Institute of Computer & Software Engineering, Khwaja Fareed University of Engineering and Information Technology, Pakistan; us.shamsuddeen@gmail.com; bushrafareed00@gmail.com
* Correspondence

This research received no external funding.

ABSTRACT: This work explores the vulnerability of Convolutional Neural Networks (CNNs) to adversarial attacks, particularly focusing on the Fast Gradient Sign Method (FGSM). Adversarial attacks, which subtly manipulate input images to deceive machine learning models, pose significant threats to the security and reliability of CNN-based systems. The research introduces an enhanced methodology for identifying and mitigating these adversarial threats by incorporating an anti-noise predictor to separate adversarial noise and images, thereby improving detection accuracy. The proposed method was evaluated against multiple adversarial attack strategies using the MNIST dataset, demonstrating superior detection performance compared to existing techniques. Additionally, the study highlights the integration of Fourier domain-based noise accommodation, enhancing robustness against attacks. The findings contribute to the development of more resilient CNN models capable of effectively countering adversarial manipulations, emphasizing the importance of continuous adaptation and multi-layered defense strategies in securing machine learning systems.

Keywords: CNN; Adversarial attacks; Anti-noise predictor; Machine learning security; Image classification, Adversarial detection.

1. Introduction

In digital era, the rapid advancement of artificial intelligence (AI) technologies has led to their widespread application across various realms [1], [2]. AI, known for its high presentation, availability, and intelligence, has found utility in numerous areas, including machine translation, speech-activated object identification, image classification, and even more intricate fields, such as medication alignment examination, neural circuit rebuilding, accelerator unit information study, and genetic modification effect analysis. However, the vulnerability of neural networks, as primarily suggested by Zbrzezny et al., [3] has significantly shaped the field of AI argumentative methods. Researchers have been actively exploring and developing novel adversarial attack and defense methods. These attacks can be roughly categorized into three stages: attacks throughout training, attacks in testing, and attacks throughout model arrangement.

Adversarial attacks occur when the model is manipulated during the training phase by altering the training data, adjusting input features, or modifying data labels [4]. For instance, researchers have investigated methods that involve changing or omitting parts of the training data, altering the distribution of the data, or intentionally modifying labels to impact the performance of the classifier. Additionally, adversaries may craft malicious samples and inject them into the training set to distort the model's decision boundaries. This strategy is commonly referred to as manipulating input structures.

Adversarial images present a significant challenge in machine learning, but various strategies have been developed to mitigate their impact [5]. One such strategy is heuristic defense, which, despite lacking formal theoretical guarantees, can effectively counter specific types of attacks. A common heuristic technique is adversarial training, in which models are deliberately trained with adversarial examples to improve their robustness [6], [7]. This approach has demonstrated strong performance on several datasets, including MNIST, CIFAR-10, and ImageNet. In addition to adversarial training, other experimental defense strategies involve transforming the input data or extracted features to weaken or neutralize the impact of adversarial perturbations.

1.1. Research Gaps

Protecting machine learning models from adversarial attacks is a multifaceted challenge that requires a comprehensive approach [8]. This strategy may involve techniques like assessing robustness, enhancing data diversity, and employing adversarial training. Robustness evaluation focuses on testing the model's resilience to adversarial modifications, ensuring its ability to function effectively even when exposed to intentionally altered inputs.

On the other hand, data augmentation involves modifying or distorting the training data to improve the machine learning model's ability to withstand adversarial attacks [9]. In contrast, adversarial training focuses on enhancing the deep learning model by incorporating adversarial examples during training, which helps increase its resistance to such attacks.

Adversarial machine learning attacks pose a significant threat to the security and reliability of machine learning systems. As the complexity of cyberattacks continues to increase, it is essential to develop strong and effective defense mechanisms to mitigate these threats [10]. By integrating comprehensive testing, data augmentation, and adversarial training, machine learning models can be made more resilient to adversarial attacks.

The complexity of attack techniques, computing costs, and the need for on-going surveillance and adaptability to the changing threat environment are all challenges [11]. Aggregation approaches, such as the construction of various models, might improve defense by reducing the effect of particular weaknesses. In general, a diverse strategy is required to defend machine learning models against more sophisticated adversarial assaults.

1.2. Research Objectives

Machine learning and computer vision encounter significant challenges from adversarial attacks on images. These attacks involve carefully altering input images to mislead the neural networks in machine learning models [12]. Understanding the vulnerabilities of these models is a crucial area of research. Researchers strive to identify why neural networks are prone to such attacks and the underlying mathematical properties that drive adversarial perturbations. This insight is critical for developing more robust and secure image classification systems. Attack strategies in this domain range from white-box attacks, which rely on full knowledge of a model's architecture, to black-box attacks, which operate with limited information.

Targeted attacks attempt to create adversarial examples that force a model to misclassify images into specific, predefined classes, making them particularly difficult to defend against [13]. Physical-world attacks extend these threats beyond digital environments by applying adversarial perturbations to printed images or real-world objects to deceive computer vision systems. Assessing a model's robustness against adversarial attacks is another key area of research. Metrics like accuracy under attack and robustness margins are used to evaluate how well a model can resist adversarial perturbations and maintain its performance when faced with malicious input modifications.

Robustness evaluations are performed on different types of image data, such as natural images, medical scans, and satellite imagery, to identify vulnerabilities across various domains. Defensive strategies have become a critical component of adversarial image research, with studies examining preprocessing techniques, architectural modifications, and other approaches aimed at enhancing model resilience [14]. Achieving an effective balance between accuracy and robustness remains a key challenge. Furthermore, adversarial attacks on images have far-reaching implications beyond academic research, affecting real-world applications such as security, healthcare, and autonomous systems. The main research objectives are:

- The goal of this study is to investigate the vulnerabilities of a CNN model using adversarial assaults based on the FGSM.
- Applying the FGSM approach to create adversarial instances and analyzing its influence on the performance of models are part of the scope.
- The study will assess the CNN's durability in the face of such assaults, providing important insights into the security aspects of machine learning models.

2. Methodology

A methodical approach is used in the proposed approach for executing an adversarial attack on a CNN model using the Fast Gradient Sign Method (FGSM) [15] - [17]. Following that, a relevant dataset is chosen, prepared for testing, and the FGSM approach is used to produce false instances by slightly changing the input data. The effect of these changes on the CNN model's accuracy and robustness is then tested and evaluated. The method comprises fine tuning the assault approach depending on

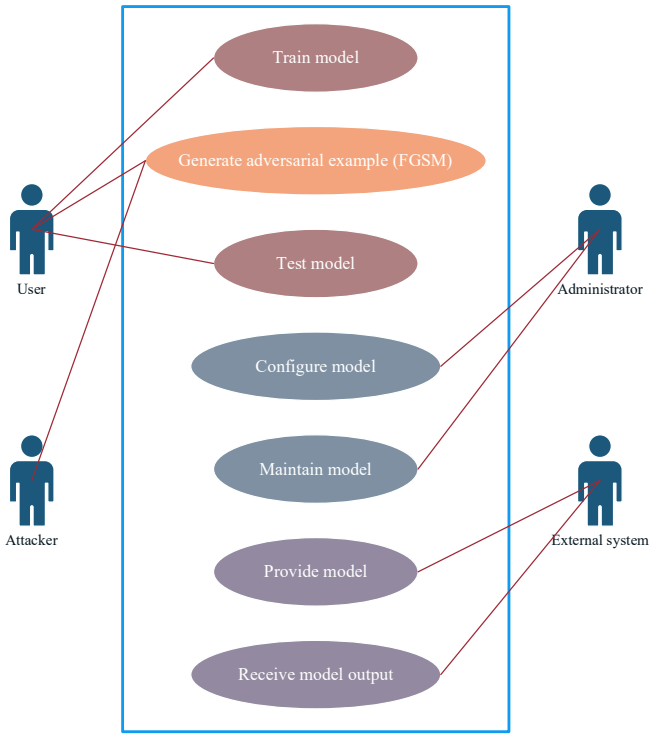

Figure 1. Use case diagram.

the results of testing. Ethical concerns are regularly addressed, with detailed documenting of parameters, observations, and techniques. At last, efforts are made to improve the model's security, and the findings are reported via a thorough report or presentation. Use case diagram of the system is shown in Figure 1.

A dropout and batch normalization regularization approaches, as well as ensemble methods, are used to improve the model's generalization and resilience. The goal of input preliminary processing, which includes controlled noise injections, is to limit susceptibility to adverse effects. A second test set of FGSM-generated adversarial instances is used for assessment, with measures like as correctness and resilience serving as crucial indications. The adjustment of hyper parameters enables appropriate model construction, and documentation is essential for transparent reporting. To remain robust against developing adversarial threats, continuous monitoring and adaptive defenses are applied, establishing a complete and active method to reinforce the CNN model from FGSM-based assaults.

2.1. Software Process Model

The Incremental Model was adopted as the software process model for developing and evaluating the proposed system [18]. This model supports iterative refinement by dividing the system into smaller, functional increments that are developed, tested, and validated in stages. Each increment delivers a subset of the overall functionality, enabling early feedback and progressive enhancement of both the adversarial attack framework and the defense mechanisms.

In the context of this study, the initial increments focused on implementing a baseline CNN classifier on the MNIST dataset and verifying its performance under clean (non-adversarial) conditions. Subsequent increments introduced FGSM-based adversarial example generation, integration of the anti-noise predictor, and incorporation of Fourier-domain noise accommodation. At each stage, performance and robustness were re-evaluated, allowing systematic refinement of model architecture, hyperparameters, and defense strategies.

The Incremental Model thus facilitated early vulnerability detection, continuous improvement, and controlled risk management. By isolating changes to specific increments, the impact of each modification on robustness and accuracy could be assessed more clearly, leading to a more reliable and maintainable adversarial attack and defense framework (Figure 2).

This model allows for flexibility, accommodating the dynamic nature of adversarial attacks and defenses, which evolve over time as new techniques and vulnerabilities emerge. The diagram emphasizes four key phases:

1) **Analysis:** In this phase, the project is thoroughly analyzed for potential risks and requirements. For adversarial attack applications, this includes understanding the CNN's vulnerabilities, evaluating the types of adversarial threats, and identifying critical areas where defenses must be built. Each iteration starts with a detailed analysis to guide the direction of the development process.
2) **Design:** Based on the analysis, the design phase refines the system architecture. For CNNs, this involves designing attack strategies such as the FGSM to manipulate the neural network and produce adversarial examples. The design phase also includes integrating various defense mechanisms to improve the model's resilience against these attacks.
3) **Code:** This phase focuses on the implementation of the designed strategies and mechanisms. Code is written to apply adversarial attacks, generate adversarial images, and test the CNN's response to these manipulations. This phase may involve tuning hyperparameters and optimizing the model's architecture for better performance.
4) **Test:** After the system is developed, extensive testing is conducted to estimate its performance under adversarial conditions. This includes testing the CNN against adversarial examples generated using FGSM and other methods, assessing the robustness of the model, and identifying any weaknesses. Testing also informs the next iteration of the process, providing feedback on how to improve the model's defenses.

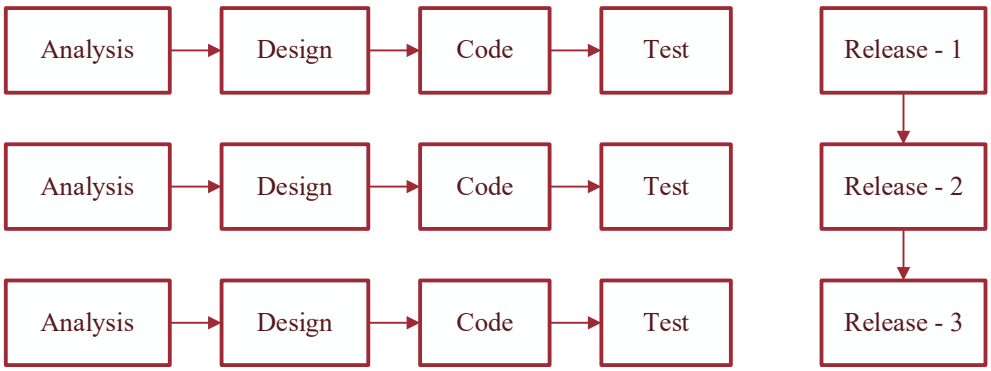

Figure 2. Incremental software process model adopted for the development and evaluation of the CNN-based adversarial attack and defense framework.

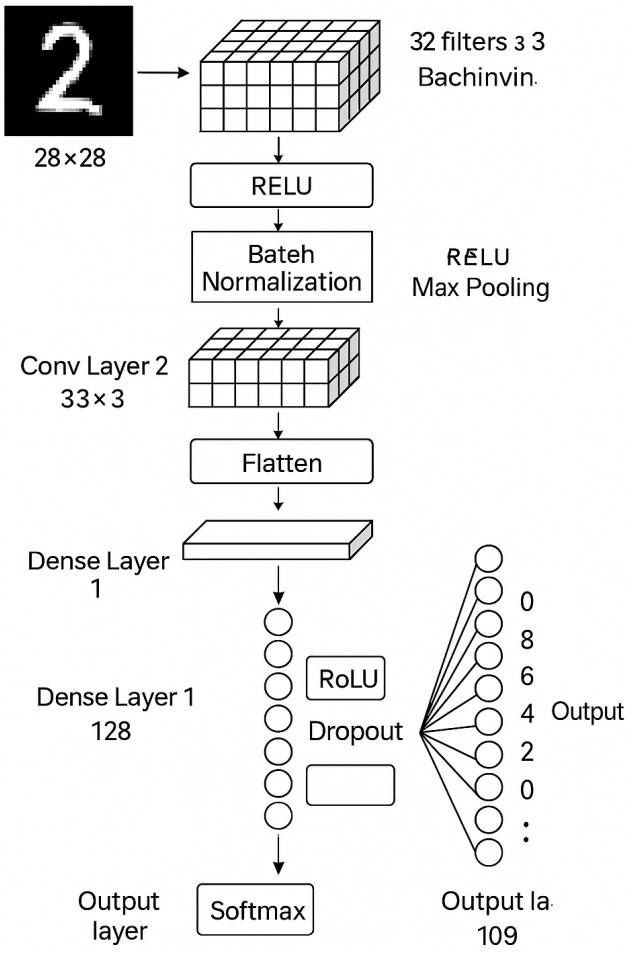

Figure 3. CNN architecture for MNIST classification.

2.2. Limitations of Selected Model

- It involves a good development designing.
- Problems may arise due to the system architecture, as not all requirements are gathered upfront for the entire software lifecycle.
- Each iteration phase is rigid and does not overlap each other.
- Fixing an issue in one unit requires adjustments in all related units, which can be time-consuming.

This study describes a methodical approach to gradually strengthen CNN models defenses against malicious assaults. The Incremental Model is a strong architecture that guarantees on going enhancement and modification in reaction to changing adversary threats.

2.3. Data collection and preparation

The dataset was obtained from Kaggle (https://www.kaggle.com/hojjatk/datasets), exactly curated for studying adversarial attacks on image classification models. The dataset's relevance to the research goals ensures it aligns with the study's objectives. It likely contains a wide variety of adversarial attack examples, providing a comprehensive range of scenarios for evaluating the model. While the exact size is not specified, Kaggle datasets are generally large, supporting robust training and evaluation of the model.

2.4. Model architecture design

The CNN architecture was designed to efficiently classify MNIST digit images while providing a meaningful basis for evaluating adversarial robustness. The model consists of two convolutional blocks followed by fully connected layers and an output layer. Each convolutional block comprises a convolutional layer, a non-linear activation function, and a pooling layer to progressively extract hierarchical features and reduce spatial dimensions. Concretely, the network takes a 28×28 grayscale image as input, as shown in Figure 3.

The extracted features are then passed to a fully connected layer with dropout regularization to mitigate overfitting, followed by a softmax output layer for 10-class digit prediction. Batch normalization is applied after the convolutional layers to stabilize training, and dropout is used in the fully connected layer to reduce overfitting. This architecture provides a good balance between representational capacity and computational efficiency, making it suitable for systematic evaluation under FGSM-based adversarial attacks.

2.5. Training procedure

The training procedure involved several steps to optimize the CNN model's performance while mitigating vulnerabilities to adversarial attacks. The dataset was split

into training, validation, and test sets to evaluate model performance effectively. During training, adversarial examples were generated and incorporated into the training procedure to enhance model robustness. Training parameters such as learning rate, batch size, and optimizer choice were tuned through experimentation to achieve optimal performance. Regular monitoring and adjustment of the training process ensured consistent progress towards achieving the research objectives.

2.6. Evaluation and analysis

After training the CNN model, extensive evaluation and analysis were conducted to assess its susceptibility to adversarial attacks. Performance metrics such as accuracy, precision, recall, and F1 score were computed to measure the model's effectiveness in classifying adversarial examples. Adversarial robustness metrics, including robust accuracy and adversarial success rate, were calculated to quantify the model's resilience against attacks. Statistical tests and visualization techniques were employed to analyze the results and draw meaningful conclusions regarding the efficacy of various security mechanisms in enhancing model robustness against adversarial threats.

For a multi-class classification problem, where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives respectively (aggregated or computed per class), the standard evaluation metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$\text{F1-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

To explicitly quantify robustness under adversarial conditions, the following metrics were used:

$$Robust\ Accuracy = \frac{Number\ of\ correctly\ classified\ adversarial\ examples}{Total\ number\ of\ adversarial\ examples} \quad (4)$$

$$Adversarial\ Success\ Rate = 1 - Robust\ Accuracy \quad (5)$$

Robust accuracy measures the proportion of adversarially perturbed samples that remain correctly classified, while adversarial success rate reflects the effectiveness of the attack in causing misclassification. These metrics are reported alongside the standard accuracy and F1-score to provide a comprehensive assessment of the model's performance under both clean and adversarial conditions.

For clean MNIST samples, the CNN achieved high accuracy and strong precision–recall performance, indicating effective feature learning. However, under FGSM perturbations, robust accuracy decreased significantly, and the adversarial success rate increased, confirming the model's vulnerability to gradient-based attacks. The drop in F1-score for adversarial samples further demonstrates the degradation in class-wise performance, particularly among digits whose boundaries are easily distorted by small perturbations. These metric-based observations validate the need for the proposed defense mechanisms, which subsequently improved robust accuracy and reduced adversarial success rate, showing measurable enhancement in resilience.

3. Implementation

The development of the botnet attack detection program involved rigorous steps. Algorithms utilizing machine learning, deep learning, and ensemble techniques were crafted to analyze network traffic for suspicious patterns [19]. After implementation, the program underwent extensive testing, including simulated and real-world scenarios, leading to iterative refinement. Optimization focused on enhancing performance to handle real-time traffic efficiently. Validation ensured effectiveness in live environments. This thorough process yielded a robust botnet detection program capable of accurately identifying malicious activity, reflecting a comprehensive approach to cyber security solution development and refined multiple times to ensure accuracy and efficiency in identifying malware.

Throughout implementation, a carefully devised strategy was followed. The primary focus was on engineering a user-friendly graphical user interface (GUI) to facilitate effortless interaction with the program. The GUI was designed to offer intuitive controls and provide clear visual feedback, thereby enhancing the overall user experience. Considerations for scalability and performance optimization were also included to ensure that the program can efficiently manage extensive volumes of network traffic.

3.1. Tools Selection

In the development of my machine learning project, Google Colab and Visual Studio Code (VS Code) were employed alongside Python and its scientific libraries. Google Colab, a cloud-based Python environment with a Jupyter Notebook interface, served as a crucial resource for tasks demanding substantial computational resources. Leveraging its access to GPUs and TPUs, I efficiently trained intricate machine learning models on sizable datasets, accelerating the learning process. Additionally, Colab's collaborative features facilitated seamless teamwork, allowing for the sharing of notebooks and fostering collaboration among team members.

Meanwhile, Visual Studio Code emerged as my primary code editor, providing an intuitive and versatile platform for writing, organizing, and debugging code.

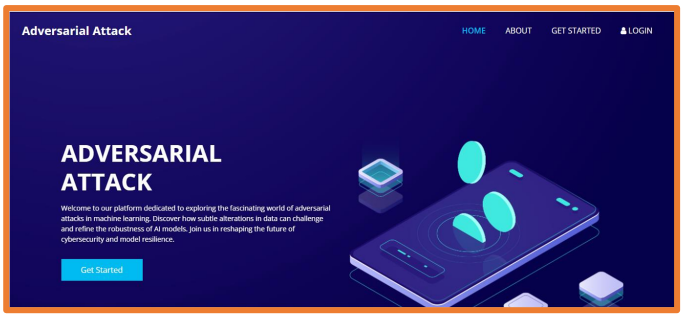

Figure 4. Homepage.

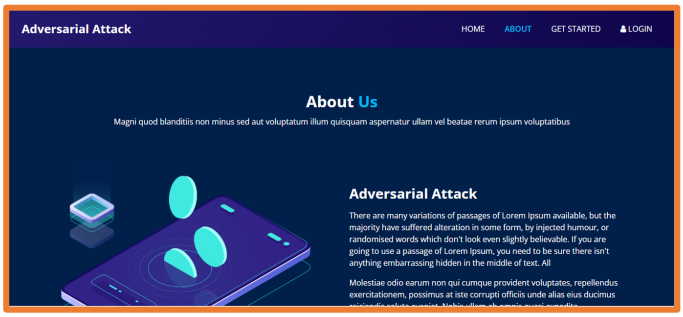

Figure 5. About-page.

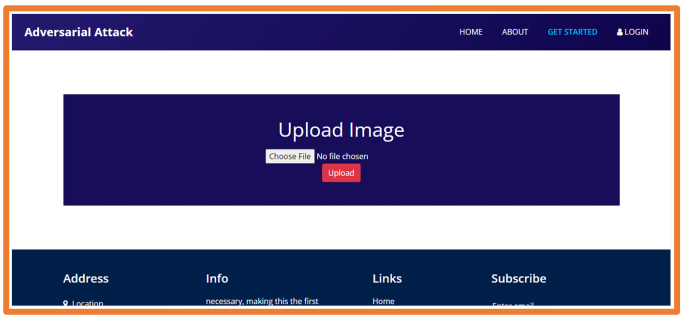

Figure 6. Start-page.

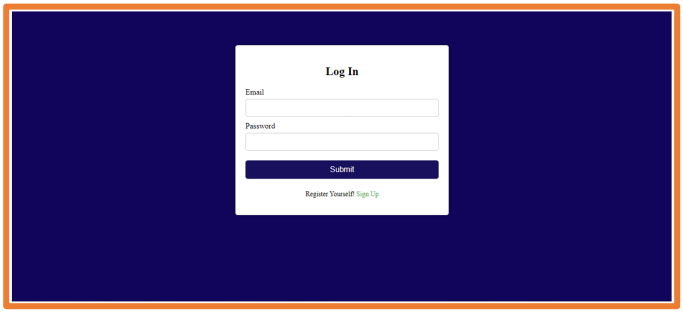

Figure 7. Login page.

Figure 8. signup page.

With its rich set of features including code highlighting, auto-completion, and robust debugging capabilities, VS Code offered a streamlined development experience. Furthermore, the extensive array of extensions and plugins available allowed for customization of the development environment to suit the specific requirements of the project. This adaptability ensured that my workflow remained efficient and tailored to the demands of the machine learning project.

The integration of Google Colab and Visual Studio Code played integral roles in the success of my machine learning endeavor. While Google Colab provided the computational muscle necessary for training complex models, Visual Studio Code offered a robust coding environment for fine-tuning algorithms and debugging code. Together, these tools synergized to optimize the development process, enabling me to efficiently create and deploy sophisticated machine learning solutions. By capitalizing on the strengths of both platforms, I was able to navigate the complexities of machine learning development with confidence and precision, ultimately achieving the desired outcomes for my project.

3.2. User interface

Users can interact with the application through a standard web browser. The homepage is shown in Figure 4. Figure 4, the homepage, serves as the entry point to the application, providing users with a clean, intuitive layout. One of the key features of the homepage is the display of real-time data or system status updates. For instance, the homepage may show the status of the current attack detection process, including a progress bar for ongoing tests or a notification about the model's current security posture. This helps the user remain informed and engaged with the system. Moreover, the homepage provides direct links to the system's help section, offering resources like tutorials, FAQs, and troubleshooting guides. This helps ensure that users can quickly resolve issues without requiring technical support, further enhancing the usability of the application. The UI is intended to be easy to use and instinctive. Clients can enter the qualities of various elements connected with network traffic or conduct. When the elements are placed, the application examinations the information and decides if the organization is protected from botnet assaults or on the other hand on the off chance that there is a possible danger.

The About-page, shown in Figure 5, provides users with detailed information about the system's objectives, functionality, and underlying technology. This page is crucial for building user trust and transparency, especially when dealing with complex technologies like adversarial machine learning. On this page, users can learn about the significance of adversarial attacks, particularly in the context of CNNs, and the potential risks posed by such attacks

on the security of machine learning models. Additionally, it explains the system's purpose: to detect, classify, and mitigate adversarial perturbations that may lead to incorrect predictions or system failures.

In Figure 5, the About-page also includes sections that describe the core methodologies used in the application, such as the FGSM for generating adversarial examples, and the anti-noise predictor for detecting and mitigating adversarial noise. The Start-page is shown in Figure 6 and it acts as a launching pad for users to begin the adversarial attack detection process. Upon entering this page, users are guided step-by-step through the procedure of setting up their model and starting the adversarial testing. The page includes key features like input fields to upload datasets, configure model parameters, and specify the type of attack to be tested. It may also allow users to adjust settings such as the level of adversarial perturbation they want to introduce, or whether they wish to apply the FGSM attack method or experiment with other attack strategies.

The Start-page's design is intuitive and user-friendly, with a well-structured layout that leads users through the entire process without confusion. It includes helpful tooltips and instructional text, making it accessible to both novice and advanced users. Figure 7, the Login page, is where users can enter their credentials to access the system's advanced features. The page is designed with security and simplicity in mind. Users are prompted to provide their username/email and password to authenticate their session. For added security, the Login page may also offer options like multi-factor authentication (MFA), enhancing the system's security and ensuring that only authorized users can access sensitive information and perform critical actions.

The Login page also features a "Forgot Password" link, allowing users to reset their password if they forget it. This ensures that access to the system remains seamless, even in cases of user error. The page's clean design minimizes distractions, allowing users to focus on the login process without unnecessary elements. It also includes a link to the Signup page (as shown in Figure 8), for new users who wish to create an account and begin using the system.

In Figure 8, the Signup page, is where new users can create an account to start using the system. The design of this page is simple and straightforward, requesting basic information such as the user's name, email address, and password. Additional fields may include user preferences or settings, allowing users to customize their experience from the moment they sign up. The page is built to minimize friction, with clearly labeled fields and validation checks to prevent errors during the registration process.

Upon successful registration, users are directed to the Login page to authenticate and gain access to the platform. The Signup page may also feature links to the Terms and Conditions and Privacy Policy, ensuring that users understand their rights and obligations before they create an account. This page plays an essential role in expanding the user base and providing easy access to the adversarial attack detection system. Together, these UI elements (Figures 4 to 8) ensure a streamlined, user-friendly experience for interacting with the adversarial attack detection system. Each page is designed to facilitate specific tasks while maintaining a cohesive and intuitive interface that guides users through complex processes such as configuring attack parameters, evaluating adversarial attacks, and analyzing results. The thoughtful design of these pages contributes significantly to the effectiveness and accessibility of the system.

4. Discussion

The experimental results show that the baseline CNN achieves high accuracy on clean MNIST images, but its performance degrades noticeably under FGSM-based adversarial attacks. This confirms that even relatively simple CNN architectures are vulnerable to well-designed perturbations. When adversarial training and the proposed anti-noise predictor are incorporated, the model exhibits improved robust accuracy, indicating that the defenses are able to partially counteract the effect of adversarial perturbations.

The integration of Fourier-domain noise accommodation further contributes to robustness by attenuating high-frequency perturbations commonly introduced by FGSM. Although these defense mechanisms slightly reduce clean accuracy, the overall trade-off is favorable from a security perspective, as the model remains substantially more reliable in adversarial scenarios. These findings are consistent with prior work on adversarial training and frequency-domain defenses in image classification models [5] - [7], [15] - [17]. Overall, the choice of algorithm depends on factors such as dataset size, dimensionality, interpretability requirements, and computational resources, and it often involves a trade-off between model complexity, performance, and ease of use, as shown in Table 1.

The table summarizes the architectural configuration and training hyperparameters used in this study. These settings were selected to provide a good trade-off between classification accuracy and robustness to FGSM-based adversarial perturbations. The perturbation magnitude ϵ controls the strength of the adversarial attack, while the learning rate, batch size, and number of epochs govern the optimization dynamics of the CNN.

In this research, leveraging the computational resources offered by a cloud-based platform, eliminating the need for additional hardware infrastructure. This approach allows us to efficiently utilize scalable computing power, storage, and other services provided by the cloud

Table 1. CNN architecture and training hyperparameters used in this study.

Component	Setting	Value
Input	Image size	28 × 28, 1 channel (grayscale)
Conv Layer 1	Number of filters	32
	Kernel size	3 × 3
	Activation function	ReLU
	Pooling	Max pooling, 2 × 2
Conv Layer 2	Number of filters	64
	Kernel size	3 × 3
	Activation function	ReLU
	Pooling	Max pooling, 2 × 2
Normalization	Batch normalization	After each convolutional block
Regularization	Dropout rate	0.5
Dense Layer	Number of units	128
Output Layer	Number of units	10 (softmax)
Optimizer		Adam
Learning rate		0.001
Batch size		64
Number of epochs		20
Adversarial attack	Method	FGSM
	Perturbation magnitude ϵ	0.1
Training dataset split	Train/validation/test	60% / 20% / 20%

Table 2. Classification performance of the CNN model on clean samples (class 0) and FGSM-generated adversarial samples (class 1) across training epochs. Performance metrics are reported using precision, recall, F1-score, and sample support.

	Precision	Recall	F1-score	Support
0	1.00	0.83	0.91	12
1	0.86	1.00	0.92	12
Accuracy			0.92	24
Macro avg	0.93	0.92	0.92	24
Weighted avg	0.93	0.92	0.92	24

platform, reducing the overhead associated with managing physical hardware. By relying on the cloud, the system can focus more on developing and deploying the applications without the constraints and maintenance requirements of traditional. Table 2 presents a visualization of the classification performance of the CNN model under both clean and FGSM-generated adversarial examples. The plot summarizes key performance indicators such as accuracy and loss across training epochs, highlighting the impact of adversarial perturbations on the model's behavior. In particular, Table shows how the adversarially perturbed inputs degrade the baseline CNN's accuracy, and how the incorporation of the anti-noise predictor and Fourier-domain processing improves robustness over time [20].

This performance visualization provides an empirical complement to the methodological description: it shows the evolution of the loss function, the effect of adversarial perturbations on prediction confidence, and the resulting recovery in performance after applying the proposed defense strategy. These observations are central to understanding how the FGSM-based attacks interact with the CNN model and how the proposed defenses mitigate their impact.

5. Conclusion

This study presents a comprehensive approach to addressing the vulnerabilities of CNNs to adversarial attacks, particularly through the use of the FGSM. The research highlights the significance of adversarial perturbations in compromising the integrity and accuracy of machine learning models, stressing the need for robust defense mechanisms. The proposed methodology, incorporating an anti-noise predictor and Fourier domain noise accommodation, demonstrates improved detection accuracy and model resilience against various adversarial strategies, particularly in the context of image classification tasks. By evaluating the CNN's robustness through these innovative techniques, the study contributes to the ongoing efforts to enhance machine learning security. Furthermore, the system's ability to identify, classify, and mitigate adversarial attacks paves the way for more secure and reliable AI applications in real-world scenarios. The results show that integrating noise management techniques and developing adaptive, multi-layered defense strategies can substantially improve model security. Future work should focus on refining these detection mechanisms, exploring more advanced adversarial attack methods, and testing the robustness of the system across a

wider range of datasets and application domains to further enhance its generalization and security performance.

6. Declarations

6.1. Author Contributions

Maaz Nisar: Methodology, Software, Validation, Formal analysis, Writing - Original Draft; **Nabeel Fayyaz:** Methodology, Software, Validation, Formal analysis, Writing - Original Draft; **Muhammad Abdullah Ahmed:** Data Curation, Writing - Review & Editing, Visualization; **Muhammad Usman Shams:** Validation, Formal analysis, Methodology, Writing - Review & Editing; **Bushra Fareed:** Conceptualization, Supervision, Project administration, Funding acquisition.

6.2. Institutional Review Board Statement

Not applicable.

6.3. Informed Consent Statement

Not applicable.

6.4. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.5. Acknowledgment

We deeply appreciate the reviewers for their valuable feedback and Dr. Muhammad Usman Shoukat (School of Mechanical and Vehicle Engineering, Beijing Institute of Technology, China) for his exceptional guidance and support throughout this research. His expertise and encouragement have played a crucial role in shaping this work, and we are truly thankful for his ongoing mentorship.

6.6. Conflicts of Interest

The authors declare no conflicts of interest.

7. References

[1] Y. Lu, "Artificial intelligence: a survey on evolution, models, applications and future trends," *Journal of Management Analytics*, vol. 6, no. 1, pp. 1–29, Jan. 2019, https://doi.org/10.1080/23270012.2019.1570365.

[2] S. A. Nawaz, J. Li, U. A. Bhatti, M. U. Shoukat, and R. M. Ahmad, "AI-based object detection latest trends in remote sensing, multimedia and agriculture applications," *Front Plant Sci*, vol. 13, Nov. 2022, https://doi.org/10.3389/fpls.2022.1041514.

[3] A. M. Zbrzezny and A. E. Grzybowski, "Deceptive Tricks in Artificial Intelligence: Adversarial Attacks in Ophthalmology," *J Clin Med*, vol. 12, no. 9, p. 3266, May 2023, https://doi.org/10.3390/jcm12093266.

[4] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019, https://doi.org/10.1109/TNNLS.2018.2886017.

[5] A. Bajaj and D. K. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," *Multimed Tools Appl*, vol. 83, no. 3, pp. 9351–9416, Jan. 2024, https://doi.org/10.1007/s11042-023-15883-z.

[6] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent Advances in Adversarial Training for Adversarial Robustness," *arXiv preprint arXiv:2102.01356*, 2021, https://doi.org/10.48550/arXiv.2102.01356.

[7] M. Ivgi and J. Berant, "Achieving Model Robustness through Discrete Adversarial Training," *arXiv preprint arXiv:2104.05062*, 2021, https://doi.org/10.48550/arXiv.2104.05062.

[8] K. Zhao, L. Wang, F. Yu, B. Zeng, and Z. Pang, "FedMP: A multi-pronged defense algorithm against Byzantine poisoning attacks in federated learning," *Computer Networks*, vol. 257, p. 110990, Feb. 2025, https://doi.org/10.1016/j.comnet.2024.110990.

[9] C.-Y. Hsu, P.-Y. Chen, S. Lu, S. Liu, and C.-M. Yu, "Adversarial Examples Can Be Effective Data Augmentation for Unsupervised Machine Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, pp. 6926–6934, Jun. 2022, https://doi.org/10.1609/aaai.v36i6.20650.

[10] I. H. Sarker, "Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview," *Security and Privacy*, vol. 6, no. 5, Sep. 2023, https://doi.org/10.1002/spy2.295.

[11] M. Repetto, "Adaptive monitoring, detection, and response for agile digital service chains," *Comput Secur*, vol. 132, p. 103343, Sep. 2023, https://doi.org/10.1016/j.cose.2023.103343.

[12] J. Fang, Y. Jiang, C. Jiang, Z. L. Jiang, C. Liu, and S.-M. Yiu, "State-of-the-art optical-based physical adversarial attacks for deep learning computer vision systems," *Expert Syst Appl*, vol. 250, p. 123761, Sep. 2024, https://doi.org/10.1016/j.eswa.2024.123761.

[13] D. J. Miller, Z. Xiang, and G. Kesidis, "Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks," *Proceedings of the IEEE*, vol. 108, no. 3, pp. 402–433, Mar. 2020, https://doi.org/10.1109/JPROC.2020.2970615.

[14] A. Abomakhelb, K. A. Jalil, A. G. Buja, A. Alhammadi, and A. M. Alenezi, "A Comprehensive Review of Adversarial Attacks and Defense Strategies in Deep Neural Networks," *Technologies (Basel)*, vol. 13, no. 5, p. 202, May 2025, https://doi.org/10.3390/technologies13050202.

[15] M. Hassan, S. Younis, A. Rasheed, and M. Bilal, "Integrating single-shot Fast Gradient Sign Method (FGSM) with classical image processing techniques for generating adversarial attacks on deep learning classifiers," in *Fourteenth International Conference on Machine Vision (ICMV 2021)*, W. Osten, D. Nikolaev, and J. Zhou, Eds., SPIE, Mar. 2022, p. 48. https://doi.org/10.1117/12.2623585.

[16] A. Agarwal, R. Singh, and M. Vatsa, "The Role of 'Sign' and 'Direction' of Gradient on the Performance of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2020, pp. 2748–2756. https://doi.org/10.1109/CVPRW50498.2020.00331.

[17] S. M. A. Naqvi, M. Shabaz, M. A. Khan, and S. I. Hassan, "Adversarial Attacks on Visual Objects Using the Fast Gradient Sign Method," *J Grid Comput*, vol. 21, no. 4, p. 52, Dec. 2023, https://doi.org/10.1007/s10723-023-09684-9.

[18] W. Pedrycz and K.-C. Kwak, "The Development of Incremental Models," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 507–518, Jun. 2007, https://doi.org/10.1109/TFUZZ.2006.889967.

[19] C. J. Trammell, M. G. Pleszkoch, R. C. Linger, and A. R. Hevner, "The incremental development process in Cleanroom software engineering," *Decis Support Syst*, vol. 17, no. 1, pp. 55–71, Apr. 1996, https://doi.org/10.1016/0167-9236(95)00022-4.

[20] H. Ben Braiek and F. Khomh, "On testing machine learning programs," *Journal of Systems and Software*, vol. 164, p. 110542, Jun. 2020, https://doi.org/10.1016/j.jss.2020.110542.