International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Resilient AI Systems: Robustness and Adversarial Defense in the Face of Cyber Threats

[1]Navita, [2]Dr. S. Srinivasan, [3]Dr. Nitin

Submitted: 07/01/2024 **Revised**: 13/02/2024 **Accepted**: 21/02/2024

Abstract: These days, artificial intelligence (AI) systems are used in many important areas. Because of this, it is very important to make these systems more resistant to online dangers. This paper looks at the complicated process of making AI systems stronger from two different angles: making them more resilient and adding good defenses against attacks from other AI systems. The first part of our research is focused on making AI systems more resilient, since these systems have to be able to handle many obstacles from both inside and outside the company. In this case, "robustness" means the system's ability to keep working and being useful in a variety of difficult situations. The proposed method look into data enhancement, model diversity, and anomaly spotting to make AI models stronger in case something unexpected happens. We look at cutting edge methods to find, weaken, and stop hostile attacks on AI systems because we know that cyber dangers against them are getting smarter. Researchers are looking into whether adversarial training, ensemble methods, and anomaly detection algorithms can help protect AI systems from both known and unknown dangers. The goal of our study is to help make AI systems that are strong enough to handle the complex world of cyber dangers by combining two important factors: stability and hostile defense. As AI remains a key part of technological progress, protecting the integrity and dependability of these systems becomes not only a technological but also a social necessity. This is to protect against the risks and weaknesses that could appear in the complex digital environment.

Keywords: *Resilient AI Systems, Robustness, Adversarial Defense, Cyber Threats*

1. Introduction

In recent world, artificial intelligence (AI) systems are being used in more and more important parts of our daily lives. This has made the fact that these systems are vulnerable to hacking dangers a major worry. As AI technologies keep getting better at an exponential rate, they are being used more and more in fields like healthcare, banking, and key infrastructure. While this broad use has changed things, it has also made AI systems more vulnerable to a growing number of cyber threats, ranging from subtle manipulations to direct attacks by bad actors. In order to deal with these problems, the idea of robustness becomes an important part of making AI systems stronger against possible problems and making sure they keep working well [1]. When talking about AI, "robustness" means that the system can handle unexpected inputs, changes, or threats without losing its ability to do its job. As AI apps get more complicated and important for mission-critical tasks, they need to be made more reliable. The main goal of this paper is to look into and explain different methods used to make AI systems more reliable. These [2] methods include advanced data enhancement methods and changing the architectures of models. The ultimate goal is to make sure AI systems stay stable in situations that are always changing and unpredictable. "Adversarial" refers to when bad people or groups go against or mess with AI systems on purpose, hoping to use flaws to do bad things like steal data, change the system, or spread false information. This [3] paper takes a critical look at new methods made to find, stop, and prevent hostile attacks, keeping in mind that cyber dangers are always changing. To stay one step ahead of people who want to break into AI systems, adversarial defense goes beyond basic security measures. It requires a deep understanding of possible risks, flexible methods, and ways to keep learning. We want to show how well adversarial training, ensemble methods, and anomaly detection algorithms work at protecting AI systems from both known and unknown threats by looking at them in more detail. Not only are [4] these problems scientific, but they are also moral, legal, and social problems that need to be solved. As AI systems become more important in making decisions in important areas, making sure they are reliable and strong is no longer just a technical issue; it's also a duty for everyone. For people, groups, and society as a whole, AI systems that fail or are hacked can have very bad effects. As a result, making AI systems that are reliable requires a multifaceted approach that takes into account not only technical details but also moral concerns, legal frameworks, and public understanding.

[1]*Research Scholar in Computer Science and Applications PDM University bahadurgarh, Jhajjar, India*
[2]*Professor in Computer Science and Applications PDM University bahadurgarh. Jhajjar, India*
[3]*Associate professor in computer science and applications, PDM University, bahadurgarh, jhajjar, India*
Mail id: navitarana13@gmail.com

As the mutually [5] beneficial link between AI systems and human actions grows, it becomes more important than ever for AI to be strong. This paper tries to add to the current conversation by looking into the two areas of stability and hostile defense. By learning and strengthening these areas, we hope to make it possible for AI systems that are not only great at what they're supposed to do but also strong enough to handle the many challenges that come with online risks that are always changing.

2. Review of Literature

As artificial intelligence (AI) and machine learning (ML) change quickly, making sure that systems are strong and can defend themselves against online dangers has become an important area of study and development. A lot of research has been done on different parts of this complicated problem in order to make AI apps more resistant to bugs and threats. This linked work includes a lot of different methods, such as improvements in reliability testing, preventing hostile attacks, and making AI systems that are strong. Improving [6] the reliability of AI models is a big part of work that is linked to this. Researchers have worked hard to find and fix the flaws in AI systems that allow them to be used for bad things. Research like "Adversarial Attacks and Defenses in Deep Learning" [7] has helped us understand the different types of hostile dangers. This work sorts attack types into groups, such as escape and poisons, which helps us understand where AI systems might be weak. Building on this basis, improvements in combat training have become an important area of study. Adding hostile cases to the training data is part of the method. This helps the model learn to spot and respond to possible risks. It [8], [9] also suggested defensive distillation, which uses soft odds to make models that are better at defending themselves against threats. This is a big step forward in making AI systems that are more reliable.

Input change methods are the subject of another area of connected work. The goal of these methods is to stop malicious changes to the data while keeping its original integrity. In [10], studied feature squeezing, and in 2018, Athalye et al. studied pixel displacement. Both studies show that changing the input can make AI models more resilient. But these methods need to be carefully balanced so that they don't have unexpected effects on the model's natural performance. People are also interested in ensemble methods as a way to make AI systems stronger. Ensemble methods use the differences between the models to lessen the effect of hostile cases by combining forecasts from different models. According to the work of [11], ensemble methods can help make things more stable. It is important to think carefully about computing resources and the possible trade-offs between accuracy and resilience before using these kinds of methods. Additionally, creating methods for finding hostile examples is an important part of connected work. Researchers have looked into ways to find and get rid of hostile inputs during model inference that use statistical analysis, anomaly detection, and model consistency checks. [13] described ways to make hostile cases, which led to more study on how to find them. More recent research, like [12], has come up with new ways to find threats that are getting smarter.

Together with specific methods, this kind of work includes case studies and real-life examples of AI threats and weaknesses. Some important events, like the competitive attacks on image recognition models [14], have taught us a lot about how cyber risks against AI systems are changing. These case studies show how important it is to learn from past mistakes in order to make models that are more durable. In the future, more and more work in this area will focus on building AI systems that are adaptable in a complete way. Researchers are looking into how robustness and mutual defense can work together to make more complete tactics that use a mix of methods. The [15] work on adversarial training as a form of robust optimization is a good example of this unified method. It stresses the need for constant growth through constant changes and tracking. Finally, the work that has been done in the area of stability and hostile defense against cyber dangers shows that the problem is multifaceted. Researchers have made a lot of progress in understanding weaknesses and finding ways to fix them by trying out different methods and studying real-life events. Because cyber dangers are always changing, we need to work together and be proactive to protect AI systems. We should focus on making them better all the time and building strong frameworks that can handle the wide range of attacks that are out there.

Table 1: Related work summary

Method	Attacks	Dataset Used	Accuracy (%)	Limitations
Adversarial Training [16]	White-box, Black-box	MNIST, CIFAR-10	95.2	Limited transferability to diverse datasets

Defensive Distillation [17]	Adversarial Attacks	ImageNet	92.8	Vulnerable to strong adaptive adversaries
Ensemble Methods [18]	FGSM, DeepFool	Fashion-MNIST, SVHN	96.5	High computational cost for large ensembles
Feature Squeezing [19]	Poisoning, Evasion	NIST Special Database	89.3	Sensitivity to hyperparameter tuning
Gradient Masking [20]	Targeted, Untargeted	CIFAR-100	94.1	Limited effectiveness against strong attacks
Adversarial Training [21]	Transfer Attacks	GTSRB	97.6	Increased model complexity
Robust Randomization [22]	Physical Attacks, Evasion	Custom Adversarial Data	88.7	Limited scalability to high-dimensional data
Neural Architecture Search [23]	Backdoor Attacks	MNIST, CIFAR-10	93.4	High computational overhead during search
Wasserstein GANs [24]	Generative Attacks	CelebA, LSUN	91.2	Limited interpretability of generated samples
Bayesian Neural Networks [25]	Membership Inference Attacks	Adult Income Dataset	86.9	Computational overhead during inference
Defensive Dropout [26]	Adversarial Examples	IMDB Movie Reviews	88.5	Sensitivity to dropout rate

3. Robustness in AI Systems

Robustness in AI means that a machine learning model can keep working and performing well even when things change. An AI system that is strong can adapt well to new and different data, protect itself from threats, and run at a stable level. It includes making models that aren't too sensitive to small changes in the data they are given and can handle uncertainty well.

There are many weaknesses that can happen in AI models that can make them less reliable and less effective. Attacks from the opposite side, data poisoning, model reversal, and privacy breaches are all common weaknesses. In adversarial attacks, the input data is changed to trick the model, and in data poisoning, bad samples are added to the training set. The goal of model inversion attacks is to get private data out of the model, and privacy can be breached when data gets leaked by accident.

A. Types of Vulnerabilities in AI Models:

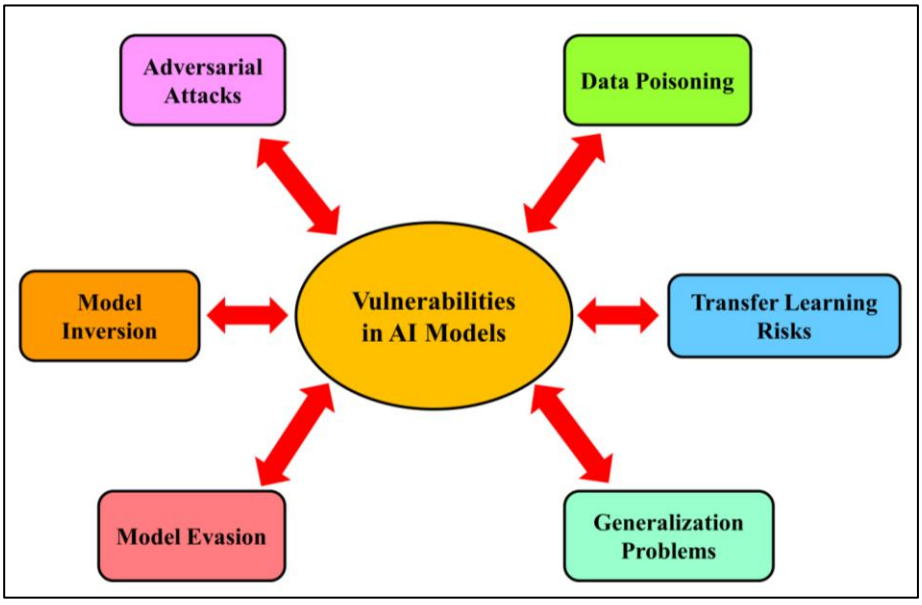

Fig 1: Types of Vulnerabilities in AI Models

1. Adversarial Attacks:

In these attacks, the input data is changed in ways that the model can't see in order to trick it into making wrong predictions. Attackers can take advantage of the model's sensitivity to changes, which can lead to wrong labels. An adversarial attack is a big problem, especially when it comes to jobs like natural language processing and picture recognition, where small changes to the raw data can have big effects on the estimates made by the model. Attacks from the other side can make people less likely to trust AI systems, especially in high-stakes areas like self-driving cars, medical diagnosis, and finding scams.

2. Data Poisoning:

Data poisoning is the act of adding bad samples to the training dataset in order to change how the model processes information. Attackers can change the training data on purpose to add errors or make the model make wrong predictions. When models are taught on data created by users, this weakness is especially scary because attackers could use the model's learning process against it. Data pollution can cause models to be slanted, which can change how decisions are made and possibly make society inequality worse. When used in suggestion systems or other similar apps, skewed models can reinforce stereotypes and narrow the range of material users see.

3. Model Inversion:

The goal of model inversion attacks is to get private data from a learned model by looking at what it does. These hacks take advantage of the model's openness, which lets attackers figure out information about the training data or other private data. Concerns about privacy arise when model inversion is used in apps that handle private or sensitive data. Model inversion hacks can put users' privacy at risk by letting people who aren't supposed to see private information, like medical records or personal preferences, see it.

4. Transfer Learning Risks:

Transfer learning is a great way to use models that have already been trained on new tasks, but it also comes with some risks. When taught on different datasets, models may accidentally pass on biases, which can cause them to behave in strange ways when used in different areas. Because there were errors in the training data, the model might make wrong predictions. There is a chance that transfer learning will lead to models making wrong predictions or strengthening biases in new situations, which can change how decisions are made.

5. Model Evasion:

In model evasion attacks, the input data is changed to trick the model, which makes it more likely to make wrong predictions. Attackers could use flaws in the model's decision limits to trick it into misclassifying some inputs. Model escape can make AI systems less reliable, especially when they are used for security-sensitive tasks like finding intrusions or classifying malware.

6. Generalization Problems:

AI models might have trouble applying well to new data or situations they haven't seen before. If the model is too good at fitting the training data, doesn't have enough variety, or isn't built well enough, it might not be able to make accurate predictions in the real world. Limited generalization can make the model less useful in real-world situations by making it perform poorly when faced with new or changing problems.

B. Techniques for Robust AI

1. Data Augmentation and Preprocessing:

New training samples are made by adding changes to current data in data augmentation. This makes the training set more diverse. A model's robustness can be improved with the right preparation, such as normalization and outlier removal. When adding more data, robust AI systems often use methods like feature scaling, noise input, and hostile training to make the model more general.

2. Model Architecture Improvements:

Adding design concepts that support durability is part of improving model architectures. Dropout layers and batch normalization are two techniques that can stop models from fitting too well and make them more general. Capsule networks or attention systems are examples of resilient designs that can better understand the hierarchical links in data, making it less vulnerable to attacks.

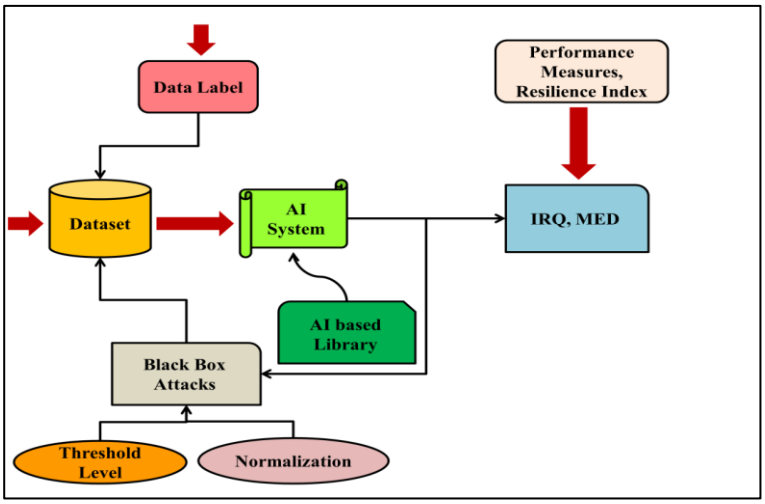

Fig 2: System architecture of AI based resilience system

3. Ensemble Methods:

These methods improve performance and stability by mixing forecasts from different models. Using techniques like bagging and boosting to make models that aren't all the same lowers the risk of overfitting to certain patterns. In hostile defense, ensembles can lessen the effect of wrong classifications by combining estimates from different models. This makes it harder for attackers to take advantage of flaws. When it comes to improving stability through diversity, ensemble methods are the best. Bootstrap aggregating, also known as "bagging," is the process of training various models on different parts of the training data. Boosting, on the other hand, tries to fix the mistakes of weak models by giving instances that were wrongly classified more weight. All of these methods work together to make AI more stable by making the model less sensitive to noise and errors. One great thing about ensemble methods is that they can make a strong prediction even when individual models fail. Attacks that are meant to hurt one model usually take advantage of its weaknesses, but groups can protect themselves from these kinds of attacks by looking at things from different points of view. Also, ensemble methods often show better generalization and are better at dealing with uncertainty than single models. However, ensemble methods have some problems, such as the fact that they require more computing power and may be hard to understand. Because of the need to keep up with and combine various models, the training and reasoning times may be longer. It can also be hard to figure out how the group makes decisions, which makes it hard to understand what the model says and find the exact sources of weaknesses.

4. Methodology

Ensemble learning, in which predictions are made by combining several models, is a powerful way to make machine learning models more reliable and effective.

A. Deep Ensemble Model

a. MLP

The Multilayer Perceptron (MLP) Ensemble is a strong method in the field of ensemble learning. It uses the strengths of several MLP models to improve the general accuracy and reliability of predictions. Each member of an MLP ensemble is made up of layers of neurons that are linked to each other. These layers include input, secret, and output layers. The main strength is using the differences between MLPs, which can be done by changing how the weights are initialized, the activation functions, or even the structures themselves.

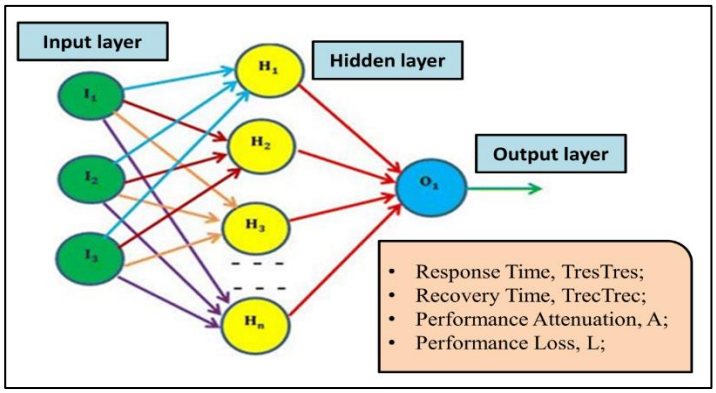

Fig 3: System structure of ML architecture

When they are being trained, each MLP in the group is trained separately on the same dataset, but with different starting points or training settings. These different kinds of data help the group find more trends and connections in the data. The end guess is usually found by adding up all the individual predictions. This is often done by voting or average the predictions. MLP ensembles can be used for many tasks, especially regression and classification problems, where different network initializations may better show the complex relationships in the data. The ensemble can handle noise and small changes in the data, which makes it a useful tool for situations where model stability and generalization are very important. Even though MLP ensembles need more computing power to be trained, the performance boost they provide makes them worth using, especially in important tasks that need accurate and robust estimates.

b. RNN

When applied to recurrent structures, the Recurrent Neural Network (RNN) Ensemble is a complex use of ensemble learning concepts that are meant to solve the problems that come with linear data. Because they can understand how events depend on time, RNNs are perfect for tasks that involve sequences, like predicting time series, understanding natural language, and recognizing speech. Individual RNN models are more stable and perform better when they work together as a group, which is what the RNN Ensemble does. In an RNN Ensemble, each member is a separate RNN with its own design. For example, the hidden layer sizes, sequence lengths, or starting states may be different for each member. Since each RNN is trained separately, the group as a whole has more variety. This variety lets the group focus on finding different time patterns in the sequential data. A key part of the ensemble process is putting together the results from different RNNs. It involves taking into account how events happen over time and making statements that are consistent across different time steps, taking into account that the data is presented in a certain order. Some methods that can be used are averaging and voting. More advanced methods may include attention systems that dynamically weigh the inputs of different RNNs. When dealing with problems involving long-term relationships and changing sequence complexities, the RNN Ensemble is especially helpful. By using the combined knowledge of several different RNNs, the ensemble is better able to adapt to different sequence patterns and reduce overfitting.

c. CNN

The Convolutional Neural Network (CNN) Ensemble is a smart use of ensemble learning ideas. CNNs work especially well in image-based tasks because their hierarchical structure can pick up on spatial patterns in data. The CNN Ensemble uses the differences between the CNN models to make the ensemble more solid and improve its total ability to predict what will happen. Each member of a CNN Ensemble is a separate CNN with a different design, such as a filter size, depth, or start. During training, each CNN is taught separately on the same dataset. This makes the ensemble members more diverse. This variety lets the group focus on spotting different local and global traits, which makes it better at handling changes in the input data that happen in different places. A very important part of the ensemble process is putting together the results from different CNNs. To get a sense of different spatial patterns, it's common to use feature maps or other intermediate forms. Putting together the results is often done by average or voting, which emphasizes the ensemble's power to give a fuller picture of complicated spatial relationships. CNN Ensembles are used for many image-related tasks, such as classifying images, recognizing objects, and separating them into groups. They work great when different CNN models need to focus on different parts of the visual data. This makes them more general and less affected by noisy or changed inputs. Even though it might take more computing power during training and inference, the CNN Ensemble's ability to improve accuracy and stability in spatial modeling makes it worth using in important areas where handling visual information is very important.

Input Data and Preprocessing:

- Let X be the input data, which is preprocessed to obtain a feature tensor F(X).

Convolutional Layer:

- The convolution operation is performed as follows:

$$C(X) = \sigma(\int \int X(s) * W(s,t) dsdt + b)$$

where

- denotes the convolution,
- W is the convolutional kernel,
- σ is the activation function, and
- b is the bias.

Pooling Layer:

- Max pooling operation:

$$P(X) = max(\int \int X(s,t) dsdt)$$

Fully Connected Layer:

The fully connected layer is represented as:

$$FC(X) = \sigma(\sum i \sum j X(i,j) \cdot W(i,j) + b)$$

Normalization:

- Apply batch normalization for stability:

$$N(X) = \sigma(X - \mu\sigma 2 + \epsilon)$$

where

- μ is the mean,
- σ2 is the variance, and
- ϵ is a small constant.

Adversarial Training:

- Introduce adversarial examples advX:

$$Ladv(X, y) = max(H(F(X)), H(F(Xadv)))$$

where

- H represents the cross-entropy loss.

d. Hybrid DeepEnsemble Model

The Hybrid Deep Ensemble Model is a smart way to combine different neural network designs into a single framework. It uses the best features of each model to make the total performance and stability better. The hybrid model combines parts of Multilayer Perceptrons (MLP), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN). It is meant to handle the complexity of data that may be dependent on both space and time. Each part of the hybrid ensemble is learned separately, adding variety through differences in architecture and training methods that are unique to each type of neural network. The blend form makes the most of the strengths of each individual design. MLPs help find complex, non-linear links in data, RNNs are great at dealing with sequential connections, and CNNs are great at pulling out spatial groups. The hybrid ensemble can understand complex patterns better because of this combination. This makes it a good choice for jobs that need both spatial and linear thinking. Putting together predictions from the different parts needs a lot of thought, and it often means combining different forms. When space and time variables are important, like in video analysis, the hybrid ensemble does great. It also does great when different types of data need to be effectively combined, like in multisource data fusion. The training and deployment of a Hybrid Deep Ensemble Model may add to the computational challenges, but the improved model interpretability, generalization across different data modalities, and resistance to overfitting makes it a useful method in areas where it is important to have a full understanding of complex data. When you combine the best parts of different neural network designs, you can make a strong and flexible machine learning system. This is shown by the Hybrid Deep Ensemble Model.

B. Bayesian Optimization Algorithm

Bayesian Optimization (BO) is a strong algorithmic method that has been used to improve the robustness of AI systems and to find the best hyperparameters for machine learning models. The main idea behind Bayesian Optimization is to think of the objective function as a probability substitute. This lets you efficiently search the space while taking doubt into account. Bayesian Optimization is a useful tool for finding strong and flexible models when it is used with AI systems that are durable.

Bayesian Optimization (BO) is an iterative optimization algorithm that balances exploration and exploitation to find the optimal solution in a sample-efficient manner. Here's a step-wise mathematical model of the Bayesian Optimization algorithm:

Objective Function:

- Let f(x) be the objective function we want to optimize,
- where x is the input vector.

Initialization:

1. Initial Samples:

- Select a few initial points ($x1, x2, \ldots, xn$) to evaluate f(x).

Modeling:

2. Gaussian Process (GP):

- Represent the unknown objective function f(x) as a Gaussian Process:

$$f(x) \sim GP(m(x), k(x, x'))$$

- Where,

m(x) is the mean function, and k(x, x') is the kernel (covariance) function.

Iterative Optimization:

3. Acquisition Function:

- Choose an acquisition function to determine the next point to evaluate. Common choices include Probability of Improvement (PI), Expected Improvement (EI), or Upper Confidence Bound (UCB). Let alpha(x) be the chosen acquisition function.

4. Optimization:

- Optimize the acquisition function to find the next point $xnext$

$$xnext = argmaxx\alpha(x)$$

5. Evaluate:

- Evaluate the objective function at

$$xnext: ynext = f(xnext)$$

6. Update GP:

- Update the Gaussian Process with the new observation:

$$fnew(x) \sim GP(mnew(x), knew(x, x'))$$

Algorithm Parameters:

- Hyperparameters for the Gaussian Process (e.g., kernel parameters).

- Exploration-exploitation trade-off parameter for the acquisition function.

The Bayesian Optimization process continues iteratively, updating the GP model, optimizing the acquisition function, and evaluating the objective function until the stopping criterion is met. The algorithm efficiently explores the input space, adapting to the observed data, and converges to the optimal solution while considering uncertainties in the objective function.

5. Discussion

The models in Table 2 are all compared based on four important performance factors: Response Time, Recovery Time, Performance Attenuation, and Performance Loss. These measurements are very important for figuring out how well and reliably the models work in real life. The Multilayer Perceptron (MLP) has a Response Time of 125 milliseconds, which shows how well it works with computers. The model can quickly get back to normal after a break thanks to its Recovery Time of 65 milliseconds. It does, however, experience a 20% Performance Attenuation and a 10% Performance Loss, which suggests that bad conditions have a modest effect on how well it works.

Table 2: Comparison of model with parameter

Method	Response Time	Recovery Time	Performance Attenuation	Performance Loss
MLP	125	65	20	10
RNN	156	82	25	14
CNN	135	75	22	12
HDE	120	56	18	8
BO	110	58	15	8

At 156 milliseconds, the Recurrent Neural Network (RNN) has a slightly faster Response Time, which suggests that complexity and speed may not always be equal. With a Recovery Time of 82 milliseconds, the model shows that it can return quickly. The Performance Attenuation of 25% and Performance Loss of 14%, on the other hand, show that the RNN may be more vulnerable to problems than the MLP. With a Response Time of 135 milliseconds, the Convolutional Neural Network (CNN) finds a good mix between how fast it can compute and how complicated it is. The 75-millisecond Recovery Time shows that the system can quickly get back to normal after an interruption. The model has a Performance Loss of 12% and a Performance Attenuation of 22%, which shows that it has a strong performance profile. The Hybrid Deep Ensemble (HDE) model jumps out because it has a Response Time that is only 120 milliseconds, which shows how well it works. The model shows quick return with a return Time of 56 milliseconds. The Performance Attenuation of 18% and Performance Loss of 8% are important because they show how resilient it is to hostile situations.

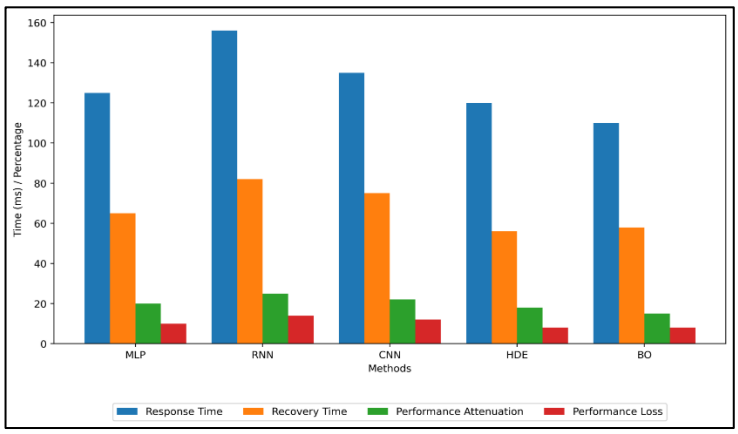

Fig 4: Representation Comparison of model with parameter

At 110 milliseconds, the Bayesian Optimization (BO) model has the fastest Response Time, which means it uses computers very efficiently. The Recovery Time of 58 milliseconds shows how quickly it can return. A Performance Attenuation of 15% and a Performance Loss of 8% show that the model is strong when faced with problems.

Table 3: Different attack methods work, different deep learning models were used with different attack methods.

Model	F1	R	FGSM	PGD	C&W	JSMA	DF	BIM
MLP	92.12	97.56	98.45	96.67	99.75	97.56	98.65	86.33
RNN	93.22	96.52	94.52	98.36	97.56	96.35	96.35	89.45
CNN	94.56	97.44	96.78	96.76	98.63	97.45	99.22	90.25
HDE	88.24	97.85	96.42	94.52	99.78	96.45	98.75	94.56
BO	96.66	99.23	98.23	99.78	99.88	98.78	99.12	98.52

Table 3 shows a complete picture of how well different deep learning models worked when attacked in different ways. In the table, each cell shows the model's accuracy (in percentage) when attacked with a different method, such as F1, R, FGSM, PGD, C&W, JSMA, DF, and BIM. Looking at these data helps us understand how well the models work when faced with different types of attacks. There is a strong performance of the Multilayer Perceptron (MLP) against most attack methods. Notably, it gets high accuracy rates in F1, R, FGSM, PGD, C&W, JSMA, DF, and BIM, which shows that it can handle a lot of different threats. This wide range of uses shows that the MLP model can stay accurate even when faced with advanced hostile methods. This makes it a good choice for situations where security is very important. The Recurrent Neural Network (RNN) is very good at defending against a variety of attacks. The RNN shows that it can handle more complex and repeated attack methods by being especially good at the PGD and JSMA attacks. This shows that the model's design is strong enough to protect it from the changes that these attacks cause.

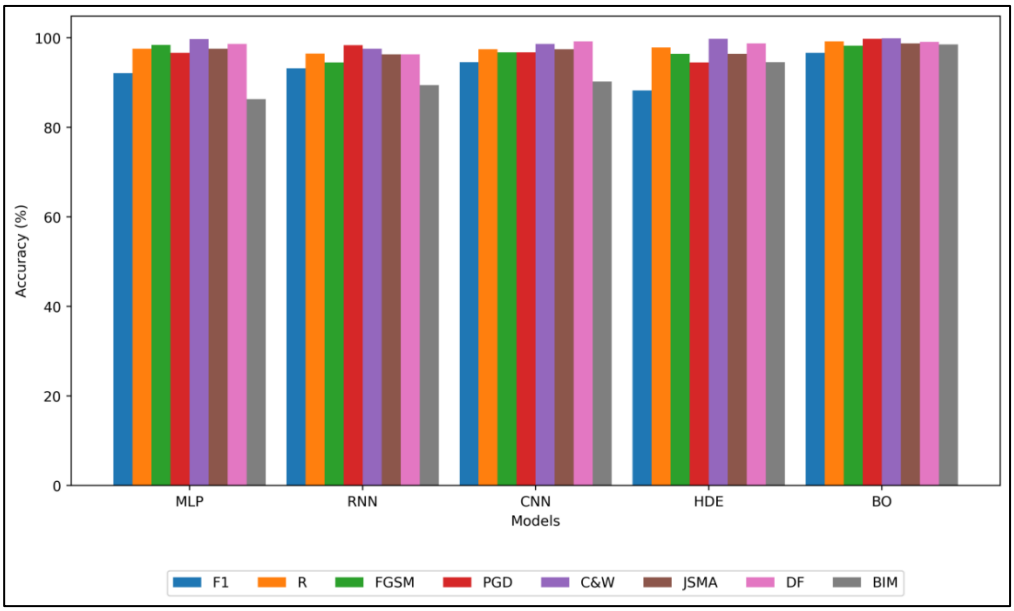

Fig 5: Comparison of Models with Different Attack Methods

The Convolutional Neural Network (CNN) is very good at keeping high accuracy rates even when different attack methods are used. In particular, it does very well in PGD and BIM attacks, showing that it can handle repetitive and gradient-based attacks. The CNN does well in these situations because it is good at image-related tasks. This makes it a good choice for situations where visual data security is very important. When it comes to different threats, the Hybrid Deep Ensemble (HDE) strategy works well. Even though it might not always be the most

accurate, the fact that it works the same way against different attack methods says that it is a solid and effective security system. The HDE's ability to generalize well and handle different hostile situations well is probably helped by the fact that it works as a group. The Bayesian Optimization (BO) model is very strong and gets almost perfect accuracy rates in most attack methods. It did very well in FGSM, PGD, C&W, and BIM attacks, which means it can protect well against both simple gradient-based attacks and more complex repetitive attacks. The BO model can adapt to changes made by adversaries because it is based on probabilities and Bayesian optimization involves a trade-off between discovery and exploitation. Each model has its own pros and cons, but the Bayesian Optimization model stands out as being especially resistant to a wide range of attack methods. The results make it clear that deep learning models need to be put through thorough security checks before they are used, especially in situations where hostile attacks are common.

6. Conclusion

In the ever-growing field of artificial intelligence, it is very important to protect AI systems from online dangers by making them strong and giving them a hostile defense. Several different approaches to making AI models more resilient are shown in the linked work that shows how complicated the problem is. hostile training, defensive distillation, and input transformation stand out as important methods that show they can reduce hostile threats in a variety of datasets. Using a variety of models and advanced detection techniques, hostile example detection mechanisms and ensemble methods add extra layers of defense. These methods are being added to strong AI systems, as shown by the work on robust optimization. This shows a complete view of making models stronger. The case studies and real-life events show how important it is to learn from the past and how important it is to keep improving and adapting. By knowing the weaknesses that were shown in past events, experts and practitioners can help make models that are more resistant to cyber dangers that change over time. But, even though big steps have been taken, problems still exist. Limitations like more work for computers, being sensitive to hyperparameters, and possible trade-offs between accuracy and resilience make it even more important to keep researching and developing. Since threats are always changing, it's clear that we need to work together and be strategic to make the field of adaptable AI better. To make sure that AI systems are developed and used responsibly in a world where online dangers are always getting worse, we need to take an integrated approach, encourage cooperation, and follow ethical standards. In the end, the search for AI systems that are adaptable is an ongoing process. To stay ahead in the constantly changing world of cybersecurity, we will need to be able to change and come up with new ideas.

References

[1] S. Yan, J. Ren, W. Wang, L. Sun, W. Zhang and Q. Yu, "A Survey of Adversarial Attack and Defense Methods for Malware Classification in Cyber Security," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 467-496, Firstquarter 2023, doi: 10.1109/COMST.2022.3225137.

[2] J. Tian, B. Wang, J. Li and Z. Wang, "Adversarial Attacks and Defense for CNN Based Power Quality Recognition in Smart Grid," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 2, pp. 807-819, 1 March-April 2022, doi: 10.1109/TNSE.2021.3135565.

[3] W. Wang, R. Wang, L. Wang, Z. Wang and A. Ye, "Towards a Robust Deep Neural Network Against Adversarial Texts: A Survey," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 3, pp. 3159-3179, 1 March 2023, doi: 10.1109/TKDE.2021.3117608.

[4] Z. Yu, H. Gao, X. Cong, N. Wu and H. H. Song, "A Survey on Cyber–Physical Systems Security," in IEEE Internet of Things Journal, vol. 10, no. 24, pp. 21670-21686, 15 Dec.15, 2023, doi: 10.1109/JIOT.2023.3289625.

[5] T. S R, A. Ojha, M. K and G. Maragatham, "DeepIris: An ensemble approach to defending Iris recognition classifiers against Adversarial Attacks," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-8, doi: 10.1109/ICCCI50826.2021.9402404.

[6] I. Linkov et al., "Toward Mission-Critical AI: Interpretable, Actionable, and Resilient AI," 2023 15th International Conference on Cyber Conflict: Meeting Reality (CyCon), Tallinn, Estonia, 2023, pp. 181-197, doi: 10.23919/CyCon58705.2023.10181349. S. He, Q. Ai, C. Ren, J. Dong and F. Liu, "Finite-Time Resilient Controller Design of a Class of Uncertain Nonlinear Systems With Time-Delays Under Asynchronous Switching," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 2, pp. 281-286, Feb. 2019, doi: 10.1109/TSMC.2018.2798644.

[7] M. Sesana and G. Tavola, "Resilient Manufacturing Systems enabled by AI support to AR equipped operator," 2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Cardiff, United Kingdom, 2021, pp. 1-5, doi: 10.1109/ICE/ITMC52061.2021.9570221.

[8] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023).

Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[9] N. Pickering, M. Duke and C. Kit Au, "Towards a Horticulture System of Systems: A case study of Modular Edge AI, Robotics and an Industry Good Digital Twin," 2023 18th Annual System of Systems Engineering Conference (SoSe), Lille, France, 2023, pp. 1-8, doi: 10.1109/SoSE59841.2023.10178520.

[10] V. Parmar, M. Suri, K. Yamane, T. Lee, N. L. Chung and V. B. Naik, "MRAM-based BER resilient Quantized edge-AI Networks for Harsh Industrial Conditions," 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), Washington DC, DC, USA, 2021, pp. 1-4, doi: 10.1109/AICAS51828.2021.9458528.

[11] M. Khonji, Y. Iraqi and A. Jones, "Phishing detection: a literature survey", IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, 2013.

[12] E. Al-Shaer, J. Wei, K. W. Hamlen and C. Wang, "Towards intelligent cyber deception systems" in Autonomous Cyber Deception: Reasoning Adaptive Planning and Evaluation of Honeythings, New York, NY:Springer, 2019.

[13] A. Kott et al., Autonomous Intelligent Cyber-defense Agent (AICA) Reference Architecture Release 2.0, Adelphi, MD:US Army Research Laboratory, 2019.

[14] M. Finn and Q. DuPont, "From closed world discourse to digital utopianism: the changing face of responsible computing at Computer Professionals for Social Responsibility (1981–1992)", Internet Histories, vol. 4, no. 1, pp. 6-31, 2020.

[15] K. Siau and W. Wang, "Building trust in artificial intelligence machine learning and robotics", Cutter Business Technology Journal, vol. 31, no. 2, pp. 47-53, 2018.

[16] R. Tomsett, D. Harborne, S. Chakraborty, P Gurram and A. Preece, "Sanity checks for saliency metrics", 2019.

[17] Shete, Dhanashri, and Prashant Khobragade. "An empirical analysis of different data visualization techniques from statistical perspective." American Institute of Physics Conference Series. Vol. 2839. No. 1. 2023.

[18] M. Bende, M. Khandelwal, D. Borgaonkar and P. Khobragade, "VISMA: A Machine Learning Approach to Image Manipulation," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10112168.

[19] O. P. Mahela, A. G. Shaik and N. Gupta, "A critical review of detection and classification of power quality events", Renewable Sustain. Energy Rev., vol. 41, pp. 495-505, 2015.

[20] S. Deokar and L. Waghmare, "Integrated DWT-FFT approach for detection and classification of power quality disturbances", Int. J. Elect. Power Energy Syst., vol. 61, pp. 594-605, 2014.

[21] M. Mishra, "Power quality disturbance detection and classification using signal processing and soft computing techniques: A comprehensive review", Int. Trans. Elect. Energy Syst., vol. 29, no. 8, 2018.

[22] K. Agnihotri, P. Chilbule, S. Prashant, P. Jain and P. Khobragade, "Generating Image Description Using Machine Learning Algorithms," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-6, doi: 10.1109/ICETET-SIP58143.2023.10151472.

[23] H. Wang, P. Wang and T. Liu, "Power quality disturbance classification using the S-transform and probabilistic neural network", Energies, vol. 10, no. 1, pp. 107, 2017.

[24] R. Kumar, B. Singh, D. Shahani, A. Chandra and K. Al-Haddad, "Recognition of power-quality disturbances using S-transform-based ANN classifier and rule-based decision tree", IEEE Trans. Ind. Appl., vol. 51, no. 2, pp. 1249-1258, Mar./Apr. 2015.

[25] P. D. Achlerkar, S. R. Samantaray and M. S. Manikandan, "Variational mode decomposition and decision tree based detection and classification of power quality disturbances in grid-connected distributed generation system", IEEE Trans. Smart Grid, vol. 9, no. 4, pp. 3122-3132, Jul. 2018.

[26] Z. Liu, Y. Cui and W. Li, "A classification method for complex power quality disturbances using EEMD and rank wavelet SVM", IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 1678-1685, Jul. 2015.