ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

FORTIFYING SMART MANUFACTURING: DNN MODELS FOR ADVANCED SECURITY IN SMART SENSING PRODUCTION SYSTEMS

Akula Joshitha, Dr. Arun Elias, Akavaram Swapna
Department of Computer Science Engineering, Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana

ABSTRACT

The use of smart sensing technologies in production systems has grown in popularity in recent years. These systems use sensors to gather data and interpret it in real time, making industrial processes more automated and efficient. However, strong security measures are more important than ever to guard against possible cyber attacks and vulnerabilities due to the increasing complexity and interconnection of these smart sensing production systems. These difficulties include the possibility of unwanted access to private information, falsification of sensor readings, and interference with device-to-device communication. Hence, creating a security architecture that can successfully counteract these new dangers and guarantee the availability, integrity, and confidentiality of the smart sensing production system is central to the issue description. Conventional security systems often depend on intrusion detection systems, firewalls, and encryption methods to secure networks and information. These steps might not be enough, nevertheless, to handle the unique difficulties presented by smart sensing manufacturing systems. Additionally, subtle and sophisticated assaults targeting the networked sensors and communication channels are difficult for traditional systems to detect. Therefore, a more intelligent and adaptable security solution is required, one that can recognize the special traits of smart sensing settings and take proactive measures to counter new threats. Furthermore, any security breech in contemporary industrial systems can have dire repercussions, including possible safety risks, production interruptions, and data leaks. The industrial sector is rapidly embracing Industry 4.0 principles, which emphasize the need for enhanced security measures due to the dependence on networked equipment and data-driven decision-making. Because deep neural networks (DNNs) are excellent at processing complex and high-dimensional data, this research presents a promising method for securing smart sensing production systems. DNNs are particularly well-suited for analyzing the various streams of information generated by sensors in a production environment. DNN models may identify abnormalities suggestive of security vulnerabilities and understand patterns of typical behavior by utilizing artificial intelligence and machine learning. In the context of networked and data-driven production settings, these models offer a more intelligent and adaptable approach to security, providing a better degree of protection against emerging cyber threats.

Keywords: Deep Learning, DNN, Backpropagation, Smart Sensing Technologies, Cyber Threats, Intrusion Detection

1. INTRODUCTION

Sensors are most commonly used in numerous applications ranging from body-parameters' measurement to automated driving. Moreover, sensors play a key role in performing detection- and vision-related tasks in all the modern applications of science, engineering and technology where the computer vision is dominating. An interesting emerging domain that employs the smart sensors is the Internet of Things (IoT) dealing with wireless networks and sensors

distributed to sense data in real time and producing specific outcomes of interest through suitable processing. In IoT-based devices, sensors and artificial intelligence (AI) are the most important elements which make these devices sensible and intelligent. In fact, due to the role of AI, the sensors act as smart sensors and find an efficient usage for a variety of applications, such as general environmental monitoring [1]; monitoring a certain number of environmental factorweather forecasting; satellite imaging and its use; remote sensing based applications; hazard events' monitoring such as landslide detection; self-driving cars; healthcare and so on. In reference to this latter sector, recently the usage of smart devices has been hugely increased in hospitals and diagnostic centers for evaluating and monitoring various health conditions of affected patients, remotely as well as physically [2].

Practically, there is no field of science or research which performs smartly without using the modern sensors. The wide usage and need of sensors; and IoT employed in remote sensing, environment and human health monitoring make the applications as intelligent. In the last decade, the agriculture applications have also included [3] the utilization of many types of sensors for monitoring and controlling various types of environmental parameters such as temperature, humidity, soil quality, pollution, air quality, water contamination, radiation, etc. This paper also aims to highlight the use of the sensors and IoT for remote sensing and agriculture applications in terms of extensive discussion and review. In recent years, SHM of civil structures has been a critical topic for research. SHM helps to detect the damage of a structure, and it also provides early caution of a structure that is not in a safe condition for usage. Civil infrastructure like [4] bridges get damaged with time, and the reason for the damage is heavy vehicles, loading environmental changes, and dynamic forces such as seismic. These types of changes mainly occur at existing structures constructed long ago, and various methods will detect that damage. The strategy of SHM involves observing the structure for a certain period to notice the condition of the structure and the periodic measurements of data will be collected, and the features of data will be extracted from these computation results, and the process of analysis can be done with the help of a featured data to find out the present-day health of the structure. The information collected from the process can be updated periodically to monitor the structure and based on the data collected through monitoring a structure, and the structure can be strengthened and repaired, and rehabilitation and maintenance can be completed [5].

2. LITERATURE SURVEY

Ullo et. al [6] focused on an extensive study of the advances in smart sensors and IoT, employed in remote sensing and agriculture applications such as the assessment of weather conditions and soil quality; the crop monitoring; the use of robots for harvesting and weeding; the employment of drones. The emphasis has been given to specific types of sensors and sensor technologies by presenting an extensive study, review, comparison and recommendation for advancements in IoT that would help researchers, agriculturists, remote sensing scientists and policy makers in their research and implementations.

Sivasuriyan et. al [7] provides a detailed understanding of bridge monitoring, and it focuses on sensors utilized and all kinds of damage detection (strain, displacement, acceleration, and temperature) according to bridge nature (scour, suspender failure, disconnection of bolt and cables, etc.) and environmental degradation under static and dynamic loading. This paper

presents information about various methods, approaches, case studies, advanced technologies, real-time experiments, stimulated models, data acquisition, and predictive analysis. Future scope and research also discussed the implementation of SHM in bridges. The main aim of this research is to assist researchers in better understanding the monitoring mechanism in bridges.

Dazhe Zhao et. al [8] proposed an easy-fabricated and compact untethered triboelectric patch with Polytetrafluoroethylene (PTFE) as triboelectric layer and human body as conductor. We find that the conductive characteristic of human body has negligible influence on the outputs, and the untethered triboelectric patch has good output ability and robustness. The proposed untethered triboelectric patches can work as sensor patches and energy harvester patches. Three typical applications are demonstrated, which are machine learning assisted objects distinguishing with accuracy up to 93.09–94.91 %, wireless communication for sending typical words to a cellphone, and human motions energy harvesting for directly powering electronics or charging an energy storage device.

Bacco et. al [9] described, both analytically and empirically, a real testbed implementing IEEE 802.15.4-based communications between an UAV and fixed ground sensors. In our scenario, we found that aerial mobility limits the actual IEEE 802.15.4 transmission range among the UAV and the ground nodes to approximately 1/3 of the nominal one. We also provide considerations to design the deployment of sensors in precision agriculture scenarios.

Verma et. al [10] discussed the existing state-of-the-art practices of improved intelligent features, controlling parameters and Internet of things (IoT) infrastructure required for smart building. The main focus is on sensing, controlling the IoT infrastructure which enables the cloud clients to use a virtual sensing infrastructure using communication protocols. The following are some of the intelligent features that usually make building smart such as privacy and security, network architecture, health services, sensors for sensing, safety, and overall management in smart buildings. As we know, the Internet of Things (IoT) describes the ability to connect and control the appliances through the network in smart buildings. The development of sensing technology, control techniques, and IoT infrastructure give rise to a smart building more efficient. Therefore, the new and problematic innovation of smart buildings in the context of IoT is to a great extent and scattered. The conducted review organized in a scientific manner for future research direction which presents the existing challenges, and drawbacks.

3.PROPOSED SYSTEM

3.1 Overview

The Python script that uses the Tkinter library to create a graphical user interface (GUI) for a Smart Sensing System in an industrial environment. The GUI provides functionality for uploading and preprocessing datasets, running various machine learning algorithms (Naive Bayes, Random Forest, SVM, Logistic Regression, DNN, KNN), and displaying performance metrics.

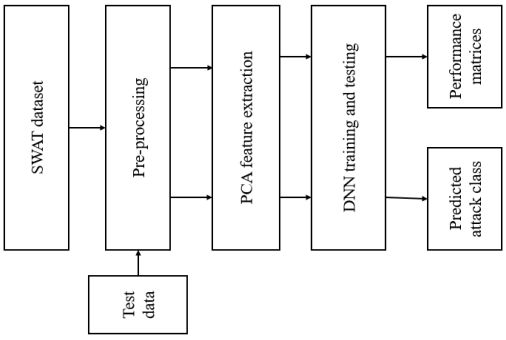


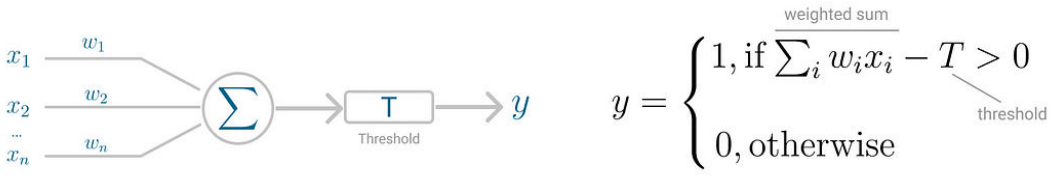
Fig. 1: Block diagram of proposed system.

3.2 DNN

3.2.1 Perceptron

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images.

In particular, interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt's model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.



Perceptron neuron model (left) and threshold logic (right).

Threshold T represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

With this discrete output, controlled by the activation function, the perceptron can be used as a binary classification model, defining a linear decision boundary.

It finds the separating hyperplane that minimizes the distance between misclassified points and the decision boundary. The perceptron loss function is defined as below:

$$\underbrace{D(w,c)}_{\text{distance}} = -\sum_{i \in \mathrm{M}} \overbrace{y_i}^{\text{output}} (x_i w_i + c)$$

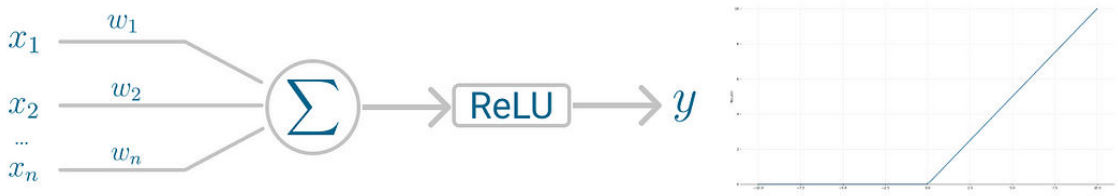
misclassified observations

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

To minimize this distance, perceptron uses stochastic gradient descent (SGD) as the optimization function. If the data is linearly separable, it is guaranteed that SGD will converge in a finite number of steps. The last piece that Perceptron needs is the activation function, the function that determines if the neuron will fire or not. Initial Perceptron models used sigmoid function, and just by looking at its shape, it makes a lot of sense! The sigmoid function maps any real input to a value that is either 0 or 1 and encodes a non-linear function. The neuron can receive negative numbers as input, and it will still be able to produce an output that is either 0 or 1.

But, if you look at Deep Learning papers and algorithms from the last decade, you'll see the most of them use the Rectified Linear Unit (ReLU) as the neuron's activation function. The reason why ReLU became more adopted is that it allows better optimization using SGD, more efficient computation and is scale-invariant, meaning, its characteristics are not affected by the scale of the input.

The neuron receives inputs and picks an initial set of weights random. These are combined in weighted sum and then ReLU, the activation function, determines the value of the output.



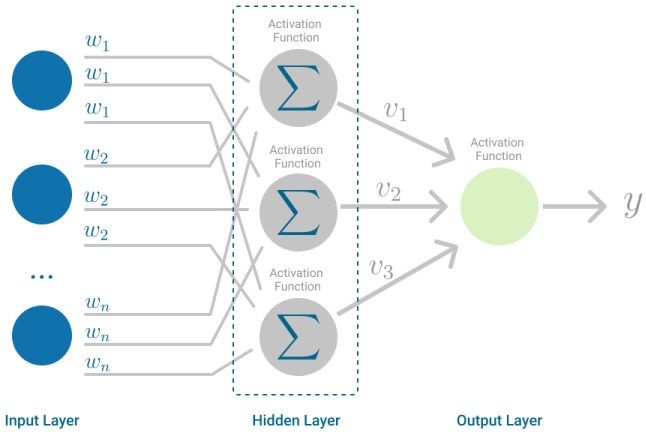
Perceptron neuron model (left) and activation function (right).

Perceptron uses SGD to find, or you might say learn, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

3.2.2 DNN

The DNN was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A DNN has input and output layers, and one or more hidden layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a DNN can use any arbitrary activation function.

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022



Architecture of DNN.

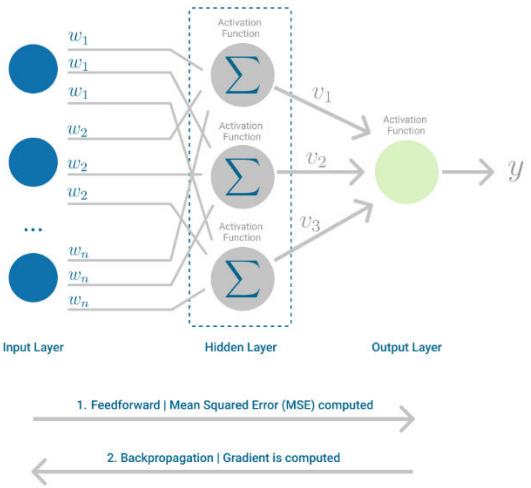
DNN falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer.

If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.

Backpropagation

Backpropagation is the learning mechanism that allows the DNN to iteratively adjust the weights in the network, with the goal of minimizing the cost function. There is one hard requirement for backpropagation to work properly.

The function that combines inputs and weights in a neuron, for instance the weighted sum, and the threshold function, for instance ReLU, must be differentiable. These functions must have a bounded derivative because Gradient Descent is typically the optimization function used in DNN.



ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

DNN, highlighting the Feedforward and Backpropagation steps.

In each iteration, after the weighted sums are forwarded through all layers, the gradient of the Mean Squared Error is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. That's how the weights are propagated back to the starting point of the neural network. One iteration of Gradient Descent is defined as follows:

$$\underbrace{\Delta_w(t)}_{\text{Gradient Current Iteration}} = \underbrace{-\varepsilon}_{\text{Bias}} \frac{\overbrace{dE}^{\text{Error}}}{\underbrace{dw_{(t)}}_{\text{Weight vector}}} + \underbrace{\alpha}_{\text{Learning Rate}} \underbrace{\Delta_{w(t-1)}}_{\text{Gradient Previous Iteration}}$$

This process keeps going until gradient for each input-output pair has converged, meaning the newly computed gradient hasn't changed more than a specified convergence threshold, compared to the previous iteration.

4.RESULTS

Figure 2 is a Sample UI used for smart sensing production system This figure shows a visual representation or screenshot of the user interface (UI) used in the smart sensing production system. Figure 3 is a Dataset used for smart sensing production system This figure displays information or characteristics of the dataset employed in the smart sensing production system. It may include details about features, labels, and data distribution. Figure 4: UI shows the dataset after preprocessing This figure represents the user interface displaying the dataset after undergoing some preprocessing steps. Preprocessing may involve cleaning, transforming, or handling missing data.

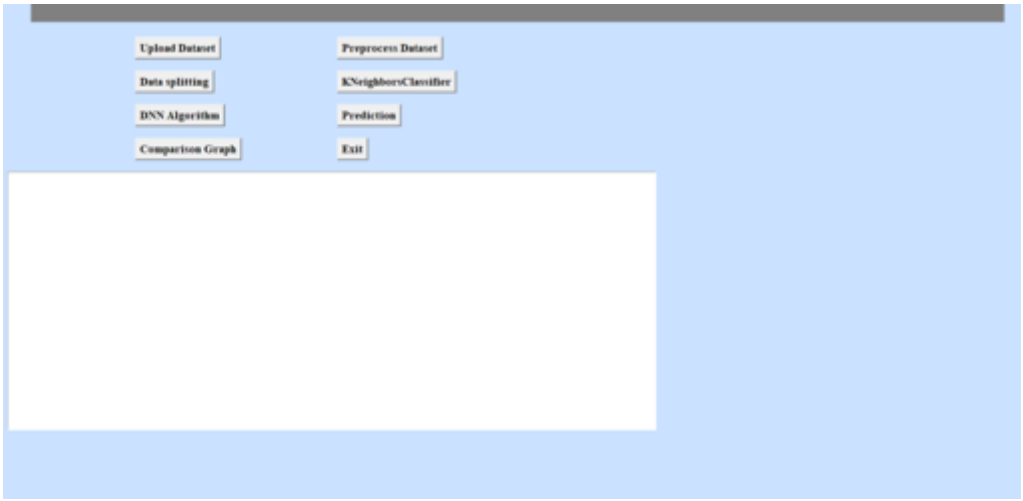


Figure 2: Sample UI used for smart sensing production system

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

Figure 3: Dataset used for smart sensing production system

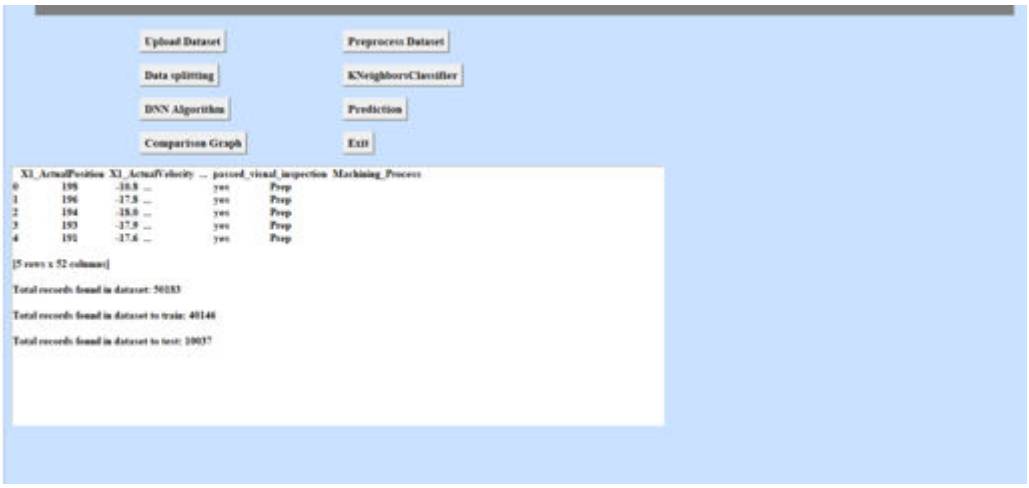


Figure 4:UI shows the dataset after preprocessing

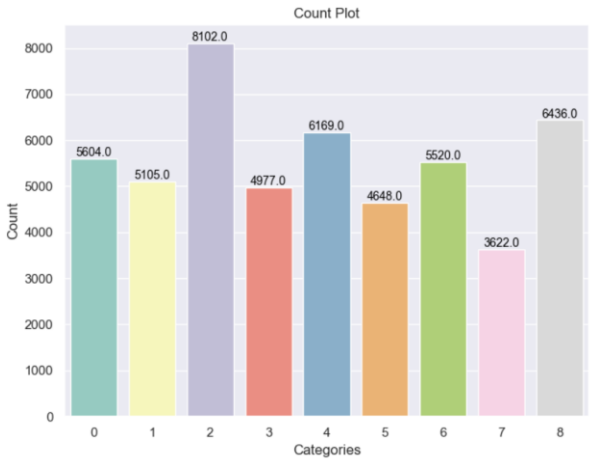


Figure 5: Count plot of target column used for smart sensing production system

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

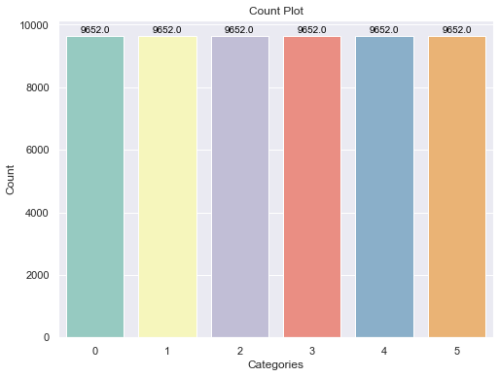


Figure 6: Count plot of target column used for smart sensing production system after preprocessing.

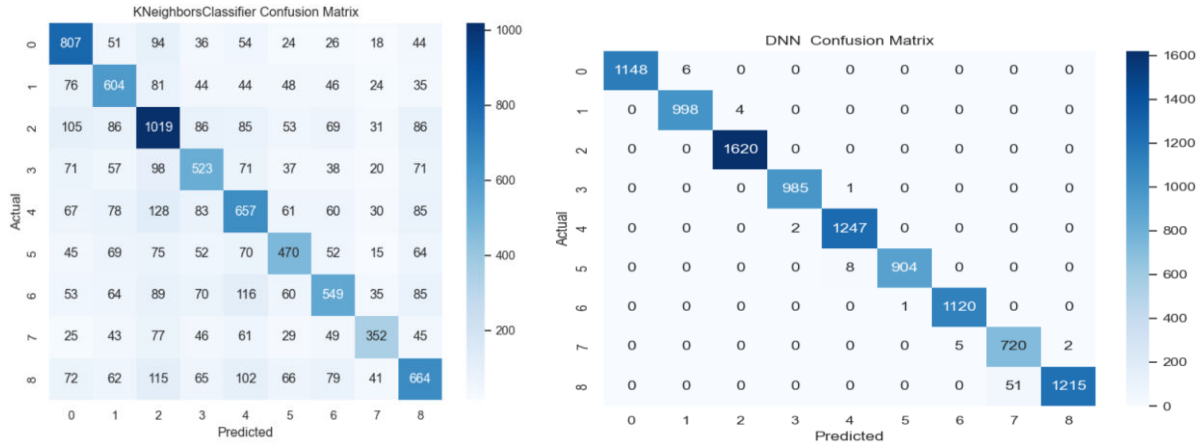


Figure 7: Confusion matrix of all machine learning and deep learning algorithms

Figure 5 is a Count plot of target column used for smart sensing production system This figure presents a count plot visualizing the distribution of the target column in the dataset before any preprocessing.

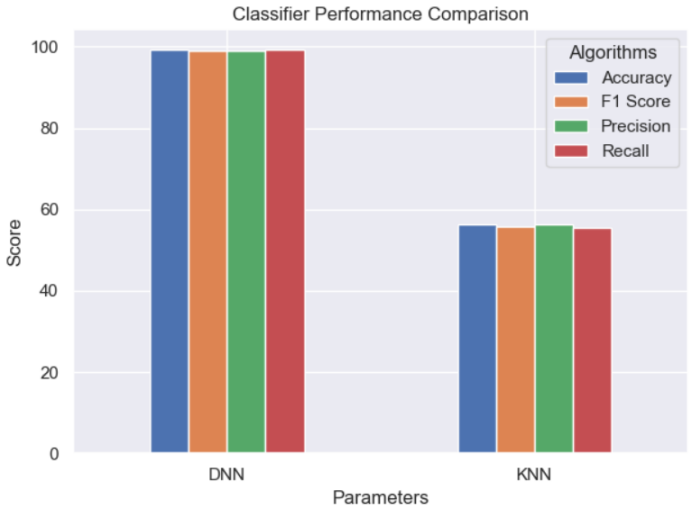


Figure 8: Performance comparison graph of all the ml algorithms

Figure 9: UI shows the prediction results on test data

Figure 6 is a Count plot of target column used for smart sensing production system after preprocessing Similar to the previous figure, this one illustrates the count plot of the target column, but after the dataset has undergone preprocessing steps.

Figure 7 is a Confusion matrix of all machine learning and deep learning algorithms This figure likely displays a confusion matrix that evaluates the performance of various machine learning and deep learning algorithms on the task at hand. It provides insights into the model's ability to correctly classify instances.

Figure 8 is a Performance comparison graph of all the ML algorithms This figure shows a performance comparison graph, possibly depicting metrics such as accuracy, precision, recall, or F1 score for different machine learning algorithms used in the smart sensing production system.

Figure 9 is a UI shows the prediction results on test data This figure demonstrates the user interface presenting the prediction results of the smart sensing production system on test data. It might include visualizations or summaries of the model's predictions.

Table 1 is a Performance comparison of quality metrics obtained using Machine Learning This table likely summarizes the performance metrics (e.g., accuracy, precision, recall) obtained from various machine learning algorithms. It provides a structured comparison of the models' effectiveness.

Table 1: Performance comparison of quality metrics obtained using Machine Learning

Algorithm	Precision	Recall	F1-Score	Accuracy
KNN	56.39	55.57	55.77	56.24
DNN	98.99	99.21	99.08	99.20

Precision: Precision measures the accuracy of positive predictions made by the model. For the KNN algorithm, the precision is 56.39%, indicating that out of all the instances predicted as p ositive, 56.39% were actually positive. In contrast, the DNN algorithm achieved a much highe r precision of 98.99%, indicating a higher accuracy in positive predictions.

Recall: Recall measures the ability of the model to identify all positive instances. The KNN al gorithm achieved a recall of 55.57%, meaning that it correctly identified 55.57% of all actual positive instances. On the other hand, the DNN algorithm had a recall of 99.21%, indicating it s superior ability to capture positive instances.

F1-Score: The F1-Score is the harmonic mean of precision and recall, providing a balance bet ween the two metrics. For the KNN algorithm, the F1-Score is 55.77%, reflecting the balance between precision and recall in its predictions. Conversely, the DNN algorithm achieved a mu ch higher F1-Score of 99.08%, indicating a strong balance between precision and recall.

Accuracy: Accuracy measures the overall correctness of the model's predictions. The KNN al gorithm achieved an accuracy of 56.24%, meaning that it correctly classified 56.24% of all in stances. In comparison, the DNN algorithm achieved a significantly higher accuracy of 99.20 %, indicating its overall superior performance in classification tasks.

Overall, the DNN algorithm outperformed the KNN algorithm across all metrics, demonstrati ng its effectiveness in accurately classifying instances and making predictions.

REFERENCES

[1] Kayad, A.; Paraforos, D.; Marinello, F.; Fountas, S. Latest advances in sensor applications in agriculture. Agriculture 2020, 10, 362.
[2] Elahi, H.; Munir, K.; Eugeni, M.; Atek, S.; Gaudenzi, P. Energy harvesting towards self-powered IoT devices. Energies 2020, 13, 5528.
[3] Ullo, S.L.; Sinha, G.R. Advances in smart environment monitoring systems using IoT and sensors. Sensors 2020, 20, 3113.
[4] Carminati, M.; Sinha, G.R.; Mohdiwale, S.; Ullo, S.L. Miniaturized pervasive sensors for indoor health monitoring in smart cities. Smart Cities 2021, 4, 146–155.
[5] Ullo, S.L.; Addabbo, P.; Di Martire, D.; Sica, S.; Fiscante, N.; Cicala, L.; Angelino, C.V. Application of DInSAR technique to high coherence Sentinel-1 images for dam monitoring and result validation through in situ measurements. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 2019, 12, 875–890.
[6] Ullo, S.L. and Sinha, G.R., 2021. Advances in IoT and smart sensors for remote sensing and agriculture applications. Remote Sensing, 13(13), p.2585.
[7] Sivasuriyan, A., Vijayan, D.S., LeemaRose, A., Revathy, J., Gayathri Monicka, S., Adithya, U.R. and Jebasingh Daniel, J., 2021. Development of smart sensing technology approaches in structural health monitoring of bridge structures. Advances in Materials Science and Engineering, 2021.
[8] Dazhe Zhao, Kaijun Zhang, Yan Meng, Zhaoyang Li, Yucong Pi, Yujun Shi, Jiacheng You, Renkun Wang, Ziyi Dai, Bingpu Zhou, Junwen Zhong, Untethered triboelectric patch for wearable smart sensing and energy harvesting, Nano Energy, Volume 100, 2022, 107500, ISSN 2211-2855, https://doi.org/10.1016/j.nanoen.2022.107500.
[9] M. Bacco, A. Berton, A. Gotta and L. Caviglione, "IEEE 802.15.4 Air-Ground UAV Communications in Smart Farming Scenarios," in IEEE Communications Letters, vol. 22, no. 9, pp. 1910-1913, Sept. 2018, doi: 10.1109/LCOMM.2018.2855211.
[10] A. Verma, S. Prakash, V. Srivastava, A. Kumar and S. C. Mukhopadhyay, "Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review," in IEEE Sensors Journal, vol. 19, no. 20, pp. 9036-9046, 15 Oct.15, 2019, doi: 10.1109/JSEN.2019.2922409.