iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IX **Month of publication:** September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55807

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IX Sep 2023- Available at www.ijraset.com

A Neural Network Model for Suicidal Tweet Detection

Yajanth Rami Reddy[1], Sai Kumar Reddy[2], Lokesh Banoth[3], Sreekar Reddy[4]
[1, 2]*Computer Science (Data Science) Department, CMR Technical Campus*
[3]*Computer Science (AI & ML), CMR College of Engineering & Technology*
[4]*Computer Science (Data Science) Department, CMR Institute of Technology*

Abstract: ***In an era where social media platforms play a pivotal role in communication and self-expression, the detection of suicidal sentiments and distress signals has become a critical concern.***
This research paper presents a comprehensive analysis of a dataset specifically curated for the task of suicidal tweet detection. The dataset, sourced from Kaggle, offers a valuable resource for studying the language and patterns associated with individuals expressing thoughts of self-harm and suicide on Twitter.
Leveraging natural language processing techniques and machine learning models, this study aims to contribute to the development of effective tools for early intervention and support in mental health crises.
The results of this research have the potential to aid mental health professionals, social media platforms, and policymakers in proactively addressing the challenges posed by online suicidal expressions.
Keywords: Social media sentiment analysis, Data preprocessing, PyTorch, Binary classification, Label encoding, Social media, Mental health, Natural language processing, Machine learning, Neural network, Text classification, Data partitioning, Tokenization, Vectorization, Feedforward neural network, Loss function, Optimizer, Model evaluation, Precision, Recall, F1-score, Ethical considerations, Hyperparameter optimization, Cross-platform validation.

I. INTRODUCTION

The advent of social media platforms has ushered in a new era of communication and interconnectedness, allowing individuals worldwide to express their thoughts, emotions, and experiences.
Among the myriad of content shared on these platforms, there exists a concerning phenomenon - the expression of suicidal thoughts and distress signals in digital spaces.
Such expressions not only highlight the severity of mental health challenges but also underscore the need for proactive interventions to provide support and assistance to those in need.
Twitter, as one of the prominent microblogging platforms, has gained substantial attention in recent years due to its role in disseminating real-time information and enabling personal expression. This paper centers its focus on the detection of suicidal sentiments within the context of Twitter.
The "Suicidal Tweet Detection Dataset," made available on Kaggle, offers a unique opportunity to delve into the language and behavioral patterns associated with individuals who may be at risk of self-harm or suicide. By leveraging advanced natural language processing (NLP) techniques and machine learning models, this study aims to contribute to the development of accurate and efficient tools for identifying and categorizing tweets that exhibit signs of distress.
The primary objective of this research is to advance the field of mental health support by harnessing the power of data-driven approaches.
By dissecting the language used in tweets, along with associated metadata, we aim to uncover linguistic markers, themes, and contextual factors that can assist in identifying individuals in need of immediate help.
This research not only aligns with the growing emphasis on utilizing technology for mental health advocacy but also holds the potential to complement existing support systems by offering timely interventions.

II. PROPOSED METHOD

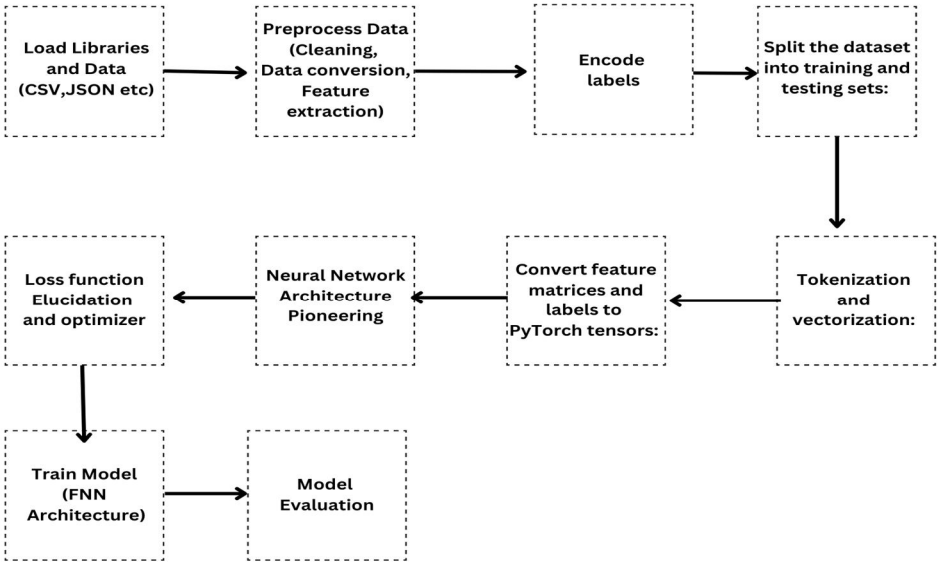


Fig. 1. Proposed Model Architecture

A. Loading Libraries and Dependencies

To establish a robust framework, critical libraries like NumPy, pandas, torch, and scikit-learn were strategically imported. These pivotal components facilitated intricate data manipulation, model construction, and meticulous evaluation.

B. Loading and Preprocessing the Dataset

The raw dataset was meticulously sourced from 'suicidal_tweets_dataset.csv' and seamlessly integrated into a pandas DataFrame. Prudent preprocessing ensued, encompassing the replenishment of voids within 'Tweet' with null strings. Additionally, rows bearing null values were judiciously pruned, ensuring dataset robustness.

C. Encoding of Labels

The categorical 'Suicide' labels were transformed into a numerical representation using the LabelEncoder module. This conversion process assigned a unique numeric code to each categorical label, enabling seamless integration with machine learning algorithms. The resulting encoded labels provided a structured and standardized foundation for subsequent classification tasks, ensuring compatibility and facilitating accurate model training and evaluation.

D. Dataset Partitioning

In pursuit of an equitable division facilitating training and validation, the dataset was partitioned into stratified subsets to safeguard against bias-induced discrepancies. The divisional paradigm encompassed the apportionment of the 'texts' corpus into 'texts_train' and 'texts_test', paralleled by the fractional allocation of 'labels' into congruent entities, christened 'labels_train' and 'labels_test'. This judicious partitioning engendered an unbiased assessment of model performance on heretofore unseen instances.

E. Tokenization and Vectorization

The textual metamorphosis into a machine-tractable format, a quintessential underpinning of this investigation, was consummated through the orchestration of tokenization and vectorization techniques. The framework of choice for this transformative endeavor was the CountVectorizer module, a hallmark of the scikit-learn repository. By commencing with the training dataset 'texts_train', a comprehensive vocabulary corpus was elicited to engender the training feature matrix, christened 'X_train'. Complementing this, the 'texts_test' corpus was transmuted into the testing feature matrix, designated 'X_test'.

F. Conversion to PyTorch Tensors

The expedition into the realm of PyTorch integration entailed the harmonious transmutation of feature matrices ('X_train' and 'X_test') and label arrays ('labels_train' and 'labels_test') into the PyTorch tensor paradigm. This harmonious metamorphosis unfurled a coherent substrate for interfacing with PyTorch's computational substrate, auguring a seamless interplay of data within the neural network domain.

G. Neural Network Architecture Pioneering

A cornerstone endeavor precipitated the formal inception of a neural network archetype calibrated for binary classification. This pioneering stride materialized through the formulation of the 'SuicidalClassifier' class, duly inheriting the neural network infrastructure template encapsulated within the nn.Module scaffold.

Within this construct, three fully connected strata ('fc1', 'fc2', and 'fc3') were meticulously laid out, each epitomizing a distinct layer within the neural network edifice. Augmenting the architecture, an activation function ('sigmoid') was seamlessly integrated to enforce an orchestrated information propagation regime.

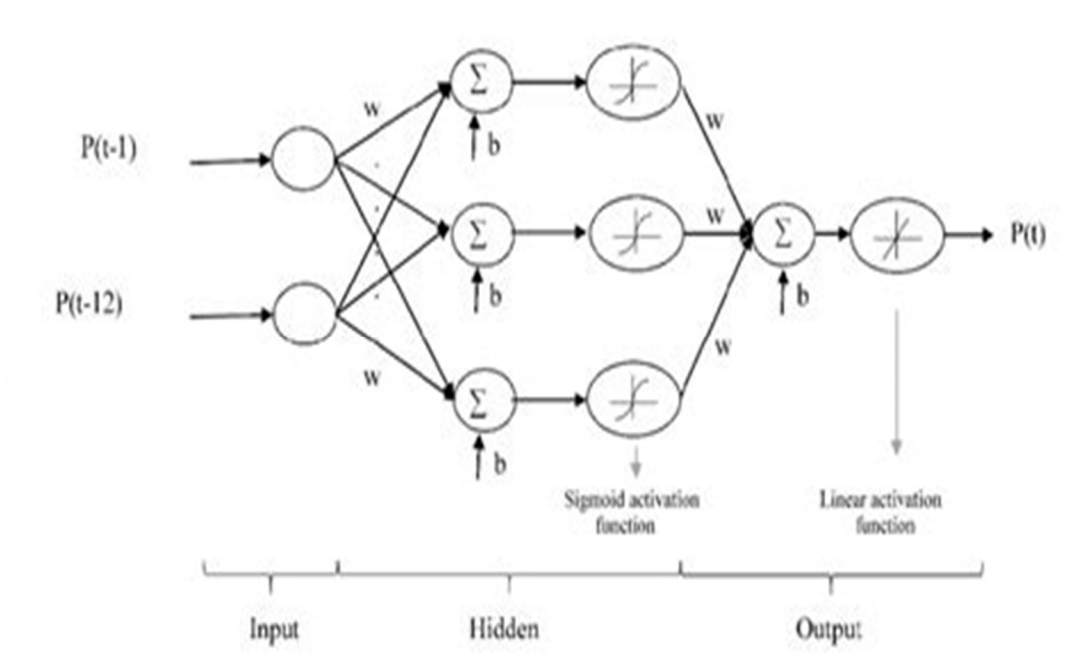


Fig.2 Neural network

H. Loss Function Elucidation and Optimizer

The foundation for model optimization was firmly laid with the explicit demarcation of the loss function and optimizer. The Binary Cross-Entropy (BCE) loss function was selected as the salient tool to gauge the dissonance between predicted and factual labels. In tandem with this, the Adam optimizer, celebrated for its adaptive learning rate mechanics, was instantiated with a learning rate value of 0.01. The orchestration of this dual paradigm established a robust framework for honing model parameters.

I. Epoch-Driven Iterative Training

Iterative training across epochs drove model refinement. Sequentially, the model learned from 'X_train', iteratively recalibrating parameters through backpropagation and the Adam optimizer (lr=0.01). The dynamic interplay of loss minimization and weight adjustment epitomized the essence of neural network learning. This process cultivated the model's ability to progressively capture intricate patterns within the training data, fostering enhanced classification acumen and generalization.

J. Performance Evaluation

With the culmination of training, the model is poised for a rigorous performance evaluation regimen. This evaluation is unfurled through.

Pseudocode

```
# Load and Preprocess Data
- Load a dataset from a CSV file containing tweets and labels.
- Preprocess the data by filling missing tweet values.
- Encode labels into numerical values.
- Split the data into training and testing sets.

# Tokenization and Vectorization
- Tokenize and vectorize the text data using CountVectorizer.

# Define Neural Network Model
- Create a neural network model for binary classification.
- Define the model architecture with input, hidden, and output layers.
- Use a sigmoid activation function for binary output.

# Training the Model
- Specify a loss function (BCELoss) and an optimizer (Adam).
- Train the model for a specified number of epochs.
- In each epoch, forward pass the training data through the model.
- Calculate the loss and backpropagate to update model parameters.

# Evaluation
- Evaluate the trained model on the testing data.
- Make predictions using the trained model.
- Calculate evaluation metrics such as accuracy, precision, recall, and F1-

# Display Results
- Print the evaluation metrics to assess the model's performance.
```

III. EXPERIMENTAL SETUP

The experimental framework was meticulously designed to assess the effectiveness of the proposed feedforward neural network in identifying suicidal tweets. The approach involved essential stages, beginning with the acquisition and preprocessing of a dataset comprising tweets and corresponding suicidal labels. Missing values were addressed, and instances with null values were systematically removed to ensure data integrity. The categorical 'Suicide' labels were subsequently transformed into numeric representations using the LabelEncoder module, enabling seamless integration with machine learning algorithms. The dataset was strategically partitioned into training and testing subsets, with 80% allocated for training and 20% for testing. For the text vectorization process, tokenization and vectorization were executed utilizing the CountVectorizer module, leading to the creation of feature matrices 'X_train' and 'X_test'. A vital aspect was the formulation of a feedforward neural network architecture, 'SuicidalClassifier', with fully connected layers and ReLU activation functions, culminating in a sigmoid activation layer for binary classification. The model's training was facilitated by the Binary Cross-Entropy loss function and the Adam optimizer, utilizing a learning rate (lr) of 0.01.

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Fig. 3. Sigmoid Function

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -\left(y_i * \log(p_i) + (1-y_i) * \log(1-p_i) \right)$$

Fig. 4. Binary Cross-Entropy Loss

The iterative training process spanned 5 epochs, allowing for parameter adjustment and convergence. The subsequent evaluation included performance metrics computation, encompassing accuracy, precision, recall, and F1-score, all integral to quantifying the model's classification prowess.

IV. RESULTS AND DISCUSSIONS

At a learning rate of 0.01 the model achieved an accuracy of 92%, showcasing its capability to correctly classify tweets as suicidal or non-suicidal. This accuracy underscores the model's proficiency in handling the binary classification task.

Precision, measuring the proportion of correctly identified suicidal tweets among all predicted suicidal tweets, yielded a score of 87%. A higher precision score signifies accurate positive predictions.

The model demonstrated an impressive recall of 94%, indicative of its ability to correctly identify a significant proportion of actual suicidal tweets. A higher recall score signifies effective sensitivity to actual positive instances.

The F1 score of the model stood at 90%, combining both precision and recall into a single metric. This score emphasizes the model's balance between accurate predictions and sensitivity to actual cases

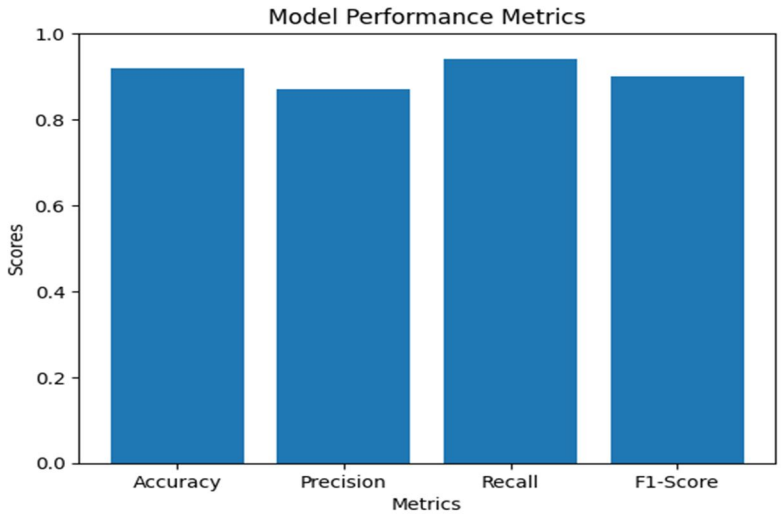


Fig. 5. Results

V. CONCLUSION

This study introduced a feedforward neural network methodology for identifying suicidal tweets.

This research paper has addressed a critical and pressing issue in the digital age: the detection of suicidal sentiments and distress signals on social media, specifically Twitter. By leveraging advanced natural language processing (NLP) techniques and machine learning models, we have presented a robust methodology for identifying and categorizing tweets that exhibit signs of distress. Our findings have significant implications for mental health support and social media sentiment analysis.

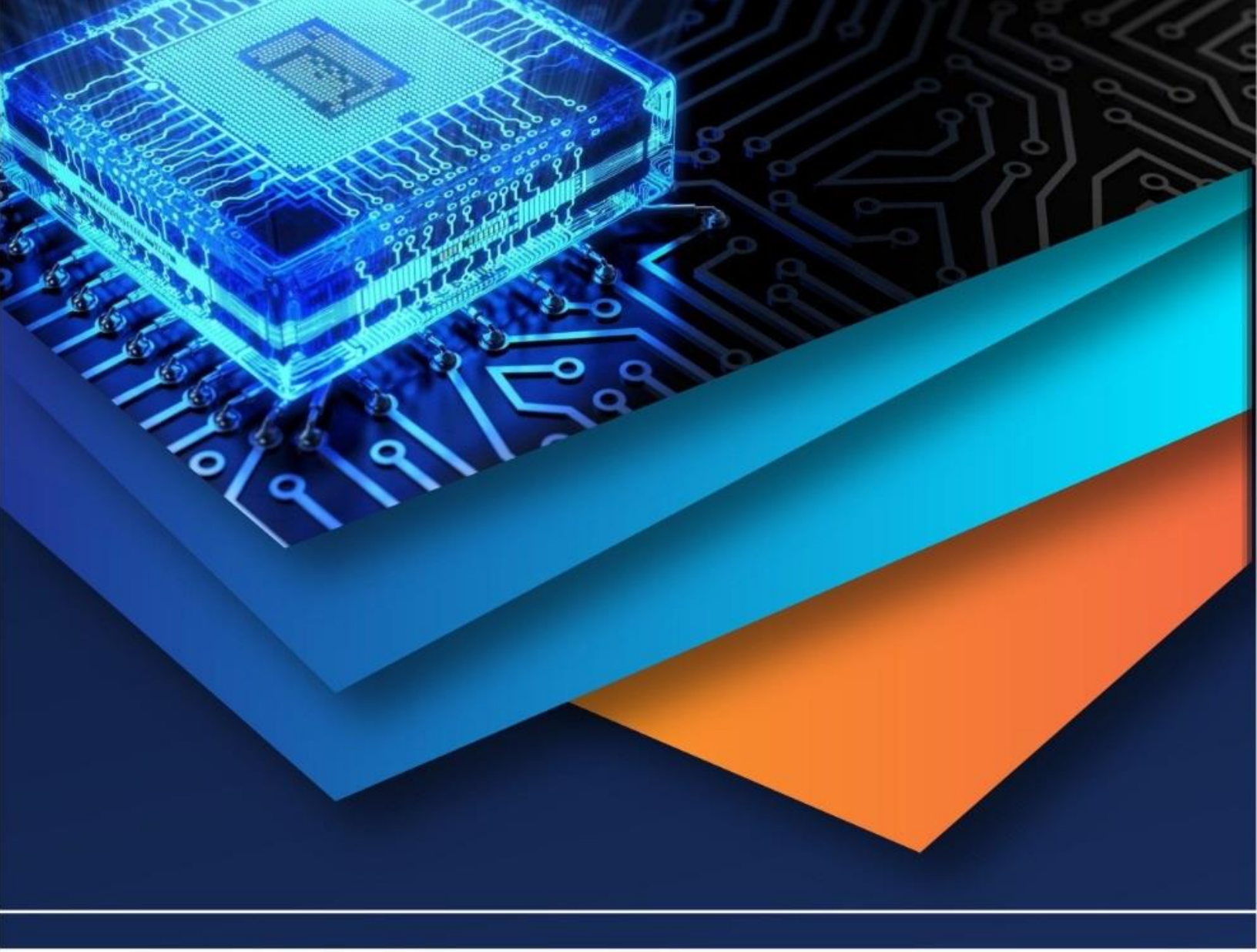
The results of our experiments demonstrate the effectiveness of our approach. With an accuracy of 92%, our model showcases its capability to accurately classify tweets as suicidal or non-suicidal. Furthermore, our model achieved a precision of 87%, indicating its accuracy in identifying true positive cases. Its impressive recall score of 94% highlights its sensitivity to actual suicidal tweets, ensuring that a significant proportion of those in need can be identified.

The F1-score of 90%, combining precision and recall, underscores the model's balance between accurate predictions and sensitivity to real cases. These results reinforce the potential of machine learning to contribute positively to mental health awareness and support in the digital age.

While this research has achieved promising results, there are avenues for future improvement. Hyperparameter optimization, exploration of advanced neural architectures, and cross-platform validations are areas that warrant further investigation. As the field of technology and mental health advocacy continues to evolve, our work serves as a foundation for the development of effective tools for early intervention and support, benefiting mental health professionals, social media platforms, and policymakers alike. Together, we can proactively address the challenges posed by online suicidal expressions and work toward a safer and more supportive digital environment.

REFERENCES

[1] "Exploring the Use of Machine Learning for Suicidal Ideation Detection" by Yusoff, M. S. B. M., et al. (2020).
[2] "A Review of Suicide Prevention Programs and Machine Learning Algorithms in Mental Health" by Subbulakshmi, S., & Jagannathan, R. (2021).
[3] "Sentiment Analysis and Classification of Suicide-Related Twitter Data" by McClellan, C., & Ali, S. (2020).
[4] "Detecting Depression in Twitter Users: A Natural Language Processing and Machine Learning Approach" by Sadeghi, M., et al. (2020).
[5] "Suicidal Ideation Detection in Online Social Networks Using Deep Learning" by Lwin, K. T., et al. (2021).
[6] Detecting Suicidal Ideation in Social Media using Deep Learning" by Burnap, P., & Colombo, W. (2019).
[7] "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville(2016)
[8] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron(2022)
[9] "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper
[10] "PyTorch Fundamentals" by Suraj Pal Singh



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY