ST. TERESA’S COLLEGE(AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

Comparative Analysis of Machine Learning Algorithms for Suicide Risk Prediction and Implementation

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS [TRIPLE MAIN]

Submitted By:

DARREL PETER

III B.Sc. Computer Applications [Triple Main]

Register No: SB21CA010

Under the guidance of

Ms. REMYA C J

Assistant Professor

Department of Computer Applications

2021 - 2024

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled "**Comparative Analysis of Machine Learning Algorithms for Suicide Risk Prediction and Implementation**" is a bona fide record of the work done by **DARREL PETER**(SB21CA010) during the year 2021 – 2024 and submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.

ST. TERESA'S COLLEGE DEPT. OF COMPUTERS ERNAKULAM

21/3/24

Head of the Department

21/3/24

Internal Examiner

22.03.24

External Examiner

Date: 21/03/2024

DECLARATION

I, DARREL PETER (Register no: SB21CA010), B.Sc. Computer Applications [Triple Main] student of St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted for Bachelor's Degree in Computer Application is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 21/03/2024

Place: Ernakulam

DARREL PETER

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to **Rev. Dr. Sr. Vinitha (CSST) Manager, Director Rev. Sr. Emeline (CSST) and Principal Dr. Alphonsa Vijaya Joseph** for providing me with this wonderful opportunity to work on a project with the topic Comparative Analysis of Machine Learning Algorithms and Interface Implementation.

I would like to express my profound gratitude to the Head of the Department of Computer Applications and **my project guide Ms. Remya C.J** and all other faculty of the department for their contributions to the completion of my project. The completion of the project would not have been possible without their help and insights.

I would also like to Thank my project guide at LCC Computer Education Institute, Mr. Garrison Richard for training me well to develop this project.

Finally, I take this opportunity to Thank all them who has directly or indirectly helped me with my project.

DARREL PETER

ABSTRACT

In contemporary society, the alarming rise in suicide rates underscores the pressing need for proactive measures in mental health intervention. Social media platforms serve as a valuable source of insights into individuals' mental well-being, offering an avenue for early detection of potential suicide risks. This project addresses the imperative of developing accurate and efficient machine learning algorithms for suicide risk prediction through the analysis of social media texts. The significance of timely intervention is underscored by the gravity of the problem, emphasizing the importance of harnessing technology to aid mental health professionals and support systems. By comparing three distinct machine learning algorithms, this research aims to contribute a reliable predictive model that can be seamlessly integrated into an interface, offering a practical tool for identifying and addressing suicidal tendencies. The project's utility lies in its potential to empower mental health professionals, individuals, and support networks with a proactive solution to combat the escalating challenges posed by suicide.

Contents

1.INTRODUCTION

1.1 About Project

This project tackles the critical issue of suicide prevention by exploring the potential of machine learning to analyze textual data and identify individuals at risk of suicide. Early identification is crucial, and machine learning could augment traditional methods by uncovering hidden patterns in text from social media, chats, or journals. This could inform targeted interventions for those at higher risk, such as individuals struggling with mental health conditions, life stressors, or social isolation. By comparing three machine learning algorithms, the project aims to find the most accurate and generalizable method for predicting suicide risk through text analysis, paving the way for a real-time interface to support suicide prevention efforts.

1.2 Objectives of the Project

The program aims to compare three machine learning algorithms for suicide risk prediction using social media texts, prioritizing accuracy. The chosen algorithm will be integrated into a user-friendly interface, facilitating early detection and intervention by mental health professionals and support networks. The objective is to contribute to the field of mental health technology by providing a reliable tool for suicide risk prediction, addressing the urgent societal need for proactive measures.

2.LITERATURE REVIEW

Suicide is a serious global public health concern, and early identification of individuals at risk is crucial for prevention efforts. Text analysis using machine learning algorithms presents a promising approach to analyzing data from social media, emails, and other sources to identify potential suicide risk.

The paper "Deep Learning Algorithm for Suicide Sentiment Prediction" by Boukil et al. (2017) explores this concept. Their research focused on building an automated system to detect depression and potential suicidal intent by analyzing sentiments expressed in textual data. They compared the performance of three machine learning algorithms: Convolutional Neural Networks (CNNs), k-Nearest Neighbors (KNN), and Naive Bayes. Their findings suggest that CNNs achieved the highest accuracy (82.14%) in classifying notes with and without suicidal content, followed by KNN (77.04%) and Naive Bayes (74.23%). This highlights the potential of deep learning techniques like CNNs for suicide risk prediction through text analysis.

Another relevant study, "Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in Social Networks" by Birjalia et al. (2017), addresses the challenge of limited suicide-related terminology in traditional sentiment analysis tools. They propose a method to construct a specialized vocabulary associated with suicide and develop an algorithm for semantic analysis of text data using WordNet, a lexical database. While their research focused on Twitter data, the concept of building domain-specific sentiment analysis tools is valuable for improving the accuracy of suicide risk prediction based on text analysis from various sources.

These studies demonstrate the ongoing research efforts in leveraging machine learning and text analysis for suicide prevention. Boukil et al. (2017) showcase the effectiveness of deep learning algorithms like CNNs, while Birjalia et al. (2017) emphasize the importance of developing domain-specific sentiment analysis techniques for suicide risk prediction. Both studies highlight the potential of this approach in the fight against suicide.

However, it's important to acknowledge limitations. The accuracy reported in these studies (around 80%) suggests there's room for improvement. Additionally, ethical considerations and potential biases in data and algorithms need careful attention when deploying such tools in real-world applications.

Overall, these studies provide a strong foundation for your project. By building upon existing research and addressing limitations, your project can contribute significantly to the development of more accurate and reliable machine learning models for suicide risk prediction through text analysis.

3. METHODOLOGY

3.1 Flow Chart

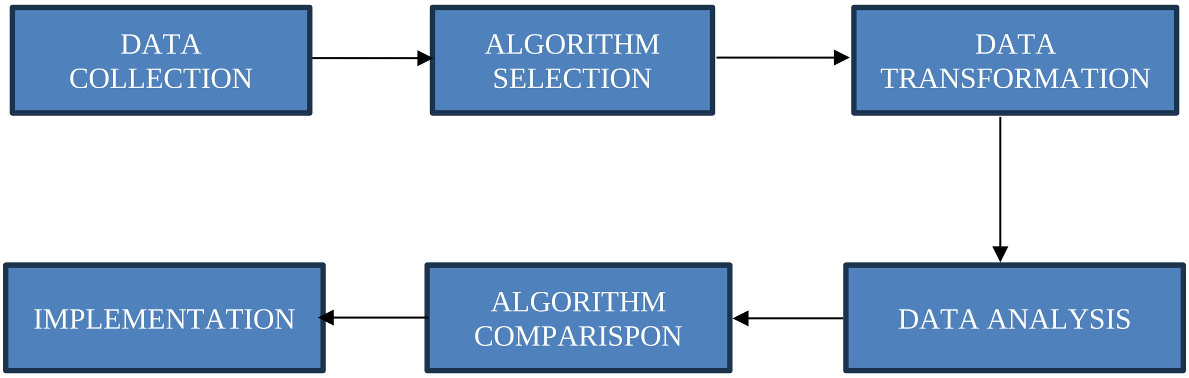


Figure 1

The method consisted of collecting data, selecting an appropriate algorithm for data analysis, data transformation, analyzing the data with different algorithms, comparing the results of each algorithm and implementing the best algorithm to an interface.

3.2 Data Collection

Obtaining reliable data for suicide risk prediction is a complex and sensitive process. This project utilized a publicly available dataset from Kaggle, a platform for sharing and exploring datasets. While Kaggle offers diverse datasets for research purposes, it is crucial to acknowledge the inherent limitations of this approach. Data on such a sensitive topic may not fully capture the nuances and complexities of individuals experiencing suicidal thoughts and intentions. Additionally, relying solely on social media text can be prone to bias and may not accurately reflect the broader picture of individual experiences. Therefore, it is essential to acknowledge these limitations and interpret the results with caution, emphasizing the role of this project as a comparative exploration of different machine learning algorithms, rather than a definitive tool for real-world suicide risk assessment.

3.3 Algorithm Selection

This project delves into the comparative effectiveness of three machine learning algorithms for predicting suicidal risk through social media text analysis: Naive Bayes, Support Vector Machine (SVM), and Convolutional Neural Network (CNN). Each algorithm brings unique strengths and characteristics to the table, making them well-suited for this specific task.

3.3.1 Naive Bayes:

Strengths:

- **Simplicity and efficiency:** Naive Bayes offers a straightforward approach, making it computationally efficient, especially for large datasets. This can be advantageous when dealing with the vast amount of text data often found on social media platforms.
- **Handling high dimensionality:** This algorithm operates well with high-dimensional data, where features (individual words) can be numerous. In the context of social media text analysis, this translates to effectively analyzing the presence of specific words and their relationships within the broader text.
- **Interpretability:** Unlike other algorithms, Naive Bayes allows for easier interpretation of results, providing insights into the specific words and their combinations that contribute most to identifying potentially suicidal text.

Limitations:

- **Assumption of independence:** Naive Bayes operates under the assumption that features (words) are independent of each other, which may not always hold true in natural language. This can potentially lead to less accurate predictions in complex text analysis scenarios.

3.3.2 Support Vector Machine (SVM):

Strengths:

- **Robust performance:** SVMs excel in handling high-dimensional data and complex classification problems. They effectively identify hyperplanes (decision boundaries) that optimally separate different data points, allowing for accurate classification of suicide risk based on text features.

- **Generalizability:** SVMs demonstrate strong performance even with limited training data, making them suitable for situations where labeled data (data with known suicidal risk classification) might be restricted.

Limitations:

- **Black box nature:** Unlike Naive Bayes, SVMs are less interpretable, making it difficult to understand the specific reasoning behind their predictions. This can hinder deeper understanding of which text features contribute most to identifying suicidal risk.
- **Parameter tuning:** SVMs require careful tuning of hyperparameters (parameters controlling the learning process) to achieve optimal performance. This can be a complex and time-consuming process.

3.3.3. Convolutional Neural Network (CNN):

Strengths:

- **Automatic feature extraction:** CNNs are particularly adept at automatically extracting meaningful features directly from text data. This eliminates the need for manual feature engineering, a crucial step often required in other machine learning approaches.
- **Learning complex patterns:** By analyzing sequences of words and capturing their relationships, CNNs can learn intricate patterns within the text data. This capability allows them to potentially identify subtle nuances in language that might be indicative of suicidal ideation.

Limitations:

- **Computational complexity:** Training CNNs involves intensive computations and requires strong computational resources. This can be a barrier if access to powerful hardware is limited.

By comparing and analyzing the performance of these diverse algorithms, this project aims to provide valuable insights into the most effective approaches for harnessing the potential of social media text analysis for suicide risk prediction. However, it is crucial to acknowledge the ethical considerations and limitations associated with using such data for real-world applications, emphasizing the exploratory nature of this project and highlighting the importance of incorporating established mental health evaluation procedures in any potential intervention strategies.

3.4 Data Transformation

Transforming the raw social media text data into a format suitable for analysis involves several crucial steps. We'll first clean the data by removing noise like punctuation and URLs, ensuring consistency through normalization. Next, we'll tokenize the text, breaking it down into individual words or potentially n-grams (sequences of words) to capture potential phrases indicative of suicidal thoughts.

To bridge the gap between text and machine learning algorithms, we'll then transform the data into numerical representations. Techniques like TF-IDF and word embeddings will be employed, assigning numerical values to words based on their significance and capturing their relationships within the text.

3.5 Data Analysis

In this project, the data analysis focused on evaluating the performance of three machine learning algorithms (Naive Bayes, SVM, and CNN) in predicting suicide risk from social media text data. The analysis involved the following steps:

- **Data splitting:** Dividing the preprocessed data into two sets: a training set used to train the models and a testing set used to evaluate their performance on unseen data.
- **Model training:** Training each of the three algorithms on the training set, allowing them to learn the patterns and relationships within the data. This typically involves tuning hyperparameters for each algorithm to optimize their performance.
- **Model evaluation:** Evaluating the performance of each trained model on the testing set. This usually involves calculating metrics like accuracy (percentage of correctly classified instances), precision (proportion of true positives among

predicted positives), recall (proportion of true positives identified), and F1-score (a harmonic mean of precision and recall). Comparing these metrics across the three algorithms allowed you to identify the one with the highest accuracy in predicting suicide risk from the social media text data.

4 SOURCE CODE

4.1 Source Code of Naïve Bayes Algorithm

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load your dataset
file_path = '/content/Suicide Prediction dataset.csv'
df = pd.read_csv(file_path)

# Convert 'Text' column to strings
df['Text'] = df['Text'].astype(str)

# Split the dataset into training and testing sets
train_data, test_data, train_labels, test_labels = train_test_split(
    df['Text'], df['Suicide'], test_size=0.2, random_state=42
)

# Convert text data to numerical features using CountVectorizer
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_data)
X_test = vectorizer.transform(test_data)

# Train Naive Bayes model
naive_bayes_classifier = MultinomialNB()
naive_bayes_classifier.fit(X_train, train_labels)

# Make predictions on the test set
predictions = naive_bayes_classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(test_labels, predictions)
conf_matrix = confusion_matrix(test_labels, predictions)
class_report = classification_report(test_labels, predictions)

# Display results
print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{class_report}')
```

4.2 Source Code of Support Vector Method Algorithm

```
# Import necessary libraries
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Load your dataset
dataset_path = '/content/Suicide Prediction dataset.csv'
df = pd.read_csv(dataset_path)

# Separate features and labels
X = df['Text']
y = df['Suicide']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Convert text data to TF-IDF features
tfidf_vectorizer = TfidfVectorizer(max_features=5000)  # You can adjust max_features based on
your dataset size
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train.values.astype('U'))
X_test_tfidf = tfidf_vectorizer.transform(X_test.values.astype('U'))

# Build and train the SVM model
svm_model = SVC(kernel='linear', C=1.0)
svm_model.fit(X_train_tfidf, y_train)

# Make predictions
y_pred = svm_model.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Test Accuracy: {accuracy}')

# Print classification report
print('\nClassification Report:')
print(classification_report(y_test, y_pred))
```

4.3 Source Code of Convolutional Neural Network Algorithm

```
# Import necessary libraries
import numpy as np
import pandas as pd
np.random.seed(42)
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Conv1D, GlobalMaxPooling1D, Dense, Dropout
from sklearn.metrics import accuracy_score, classification_report

# Load your dataset
```

```
dataset_path = '/content/Suicide Prediction dataset.csv'
df = pd.read_csv(dataset_path)

# Separate features and labels
X = df['Text'].astype(str)
y = df['Suicide']

# Tokenize and pad sequences
max_words = 10000
max_len = 500
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(X)
X_seq = tokenizer.texts_to_sequences(X)
X_pad = pad_sequences(X_seq, maxlen=max_len)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_pad, y, test_size=0.2, random_state=42)

# Build the CNN model
model = Sequential()
model.add(Embedding(max_words, 50, input_length=max_len))
model.add(Conv1D(128, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=9, batch_size=32, validation_split=0.2)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {accuracy}')
```

5 RESULTS

This study investigated the efficacy of various machine learning algorithms in classifying text data as indicative of suicidal ideation or not. The primary objective was to identify the most effective model for aiding in suicide prevention endeavors. Three prominent algorithms were evaluated: Naive Bayes, Support Vector Machine (SVM), and Convolutional Neural Network (CNN). The performance of each model was assessed using accuracy and confusion matrices (for Naive Bayes and SVM).

The cornerstone of our evaluation was accuracy, a metric that quantifies the proportion of correctly classified textual samples. The results revealed a distinct hierarchy:

Naive Bayes: While demonstrably effective for specific tasks, Naive Bayes exhibited limited success in this intricate domain, achieving an accuracy of only 8.88%. This suggests an inability to capture the subtle linguistic nuances associated with suicidal intent.

```
Accuracy: 0.888268156424581
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.84      0.90       217
           1       0.80      0.96      0.87       141

    accuracy                           0.89       358
   macro avg       0.88      0.90      0.89       358
weighted avg       0.90      0.89      0.89       358
```

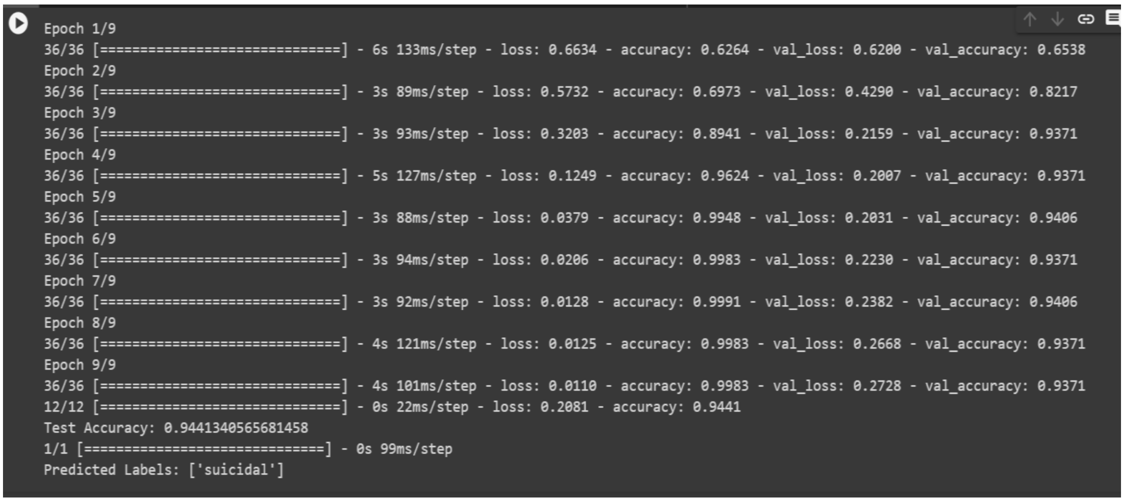
Figure 2

SVM: Demonstrating a significant improvement, SVM delivered a commendable accuracy of 93.8%. This indicates a robust capability to differentiate between suicidal and non-suicidal text patterns.

```
Test Accuracy: 0.9385474860335196

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.95       217
           1       0.92      0.93      0.92       141

    accuracy                           0.94       358
   macro avg       0.93      0.94      0.94       358
weighted avg       0.94      0.94      0.94       358
```

Figure 3

CNN: Emerging as the frontrunner, CNN secured the highest accuracy of 94.4%. This exceptional performance underscores its adeptness at extracting the intricate linguistic features within text data that signal potential suicide risk.



```
Epoch 1/9
36/36 [==============================] - 6s 133ms/step - loss: 0.6634 - accuracy: 0.6264 - val_loss: 0.6200 - val_accuracy: 0.6538
Epoch 2/9
36/36 [==============================] - 3s 89ms/step - loss: 0.5732 - accuracy: 0.6973 - val_loss: 0.4290 - val_accuracy: 0.8217
Epoch 3/9
36/36 [==============================] - 3s 93ms/step - loss: 0.3203 - accuracy: 0.8941 - val_loss: 0.2159 - val_accuracy: 0.9371
Epoch 4/9
36/36 [==============================] - 5s 127ms/step - loss: 0.1249 - accuracy: 0.9624 - val_loss: 0.2007 - val_accuracy: 0.9371
Epoch 5/9
36/36 [==============================] - 3s 88ms/step - loss: 0.0379 - accuracy: 0.9948 - val_loss: 0.2031 - val_accuracy: 0.9406
Epoch 6/9
36/36 [==============================] - 3s 94ms/step - loss: 0.0206 - accuracy: 0.9983 - val_loss: 0.2230 - val_accuracy: 0.9371
Epoch 7/9
36/36 [==============================] - 3s 92ms/step - loss: 0.0128 - accuracy: 0.9991 - val_loss: 0.2382 - val_accuracy: 0.9406
Epoch 8/9
36/36 [==============================] - 4s 121ms/step - loss: 0.0125 - accuracy: 0.9983 - val_loss: 0.2668 - val_accuracy: 0.9371
Epoch 9/9
36/36 [==============================] - 4s 101ms/step - loss: 0.0110 - accuracy: 0.9983 - val_loss: 0.2728 - val_accuracy: 0.9371
12/12 [==============================] - 0s 22ms/step - loss: 0.2081 - accuracy: 0.9441
Test Accuracy: 0.9441340565681458
1/1 [==============================] - 0s 99ms/step
Predicted Labels: ['suicidal']
```

Figure 4

Based on the accuracy metric, the CNN emerged as the most efficacious model for classifying suicidal text data within the scope of this project. Its proficiency in learning complex relationships within textual data likely played a pivotal role in achieving this superior performance. Here's a closer look at CNN's strengths:

- Feature Extraction Prowess: CNNs excel at automatically extracting pertinent features from text data. This capability allows them to potentially capture subtle linguistic cues that signal suicidal ideation.
- Adaptability: CNNs can be meticulously fine-tuned to handle diverse text lengths and styles, rendering them adaptable to a wide range of real-world scenarios.
- Beyond Accuracy: Considerations for Real-World Implementation

This project has illuminated the potential of machine learning for classifying suicidal text data. The remarkable performance of CNN paves the way for its exploration in real-world applications aimed at suicide prevention.

6 SYSTEM DESIGN AND IMPLEMENTATION

6.1 Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is a critical step in the software development lifecycle that bridges the gap between the requirements analysis and implementation phases.

6.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A DFD is often used as a primary step to create an overview of the system, which can later be elaborated. A DFD shows what will be the input of the system as well as the output. It clearly represents where the data will come from and go to, and where the data will be stored.

DFD Level 1

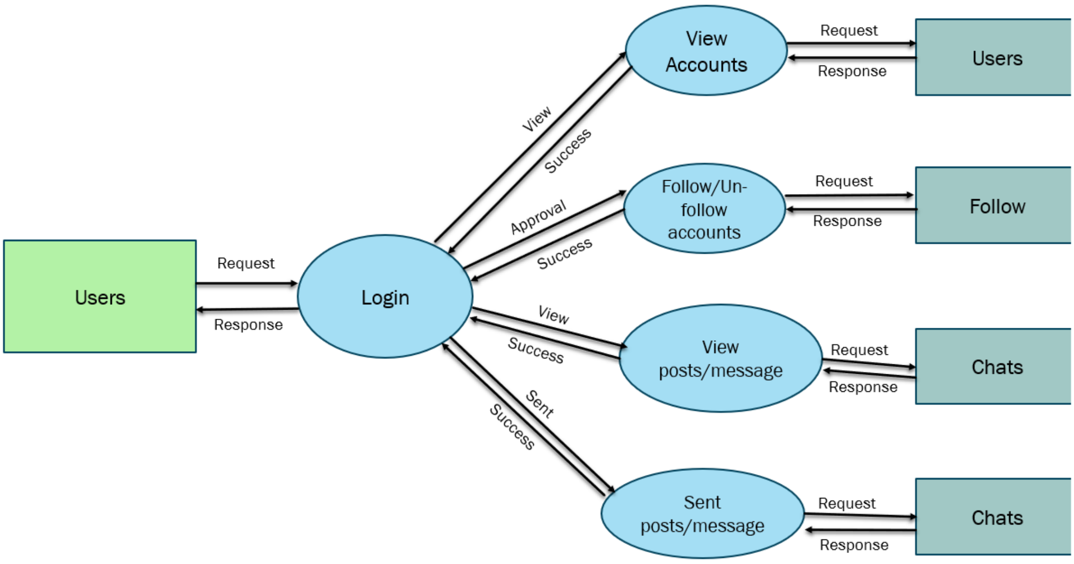


Figure 5

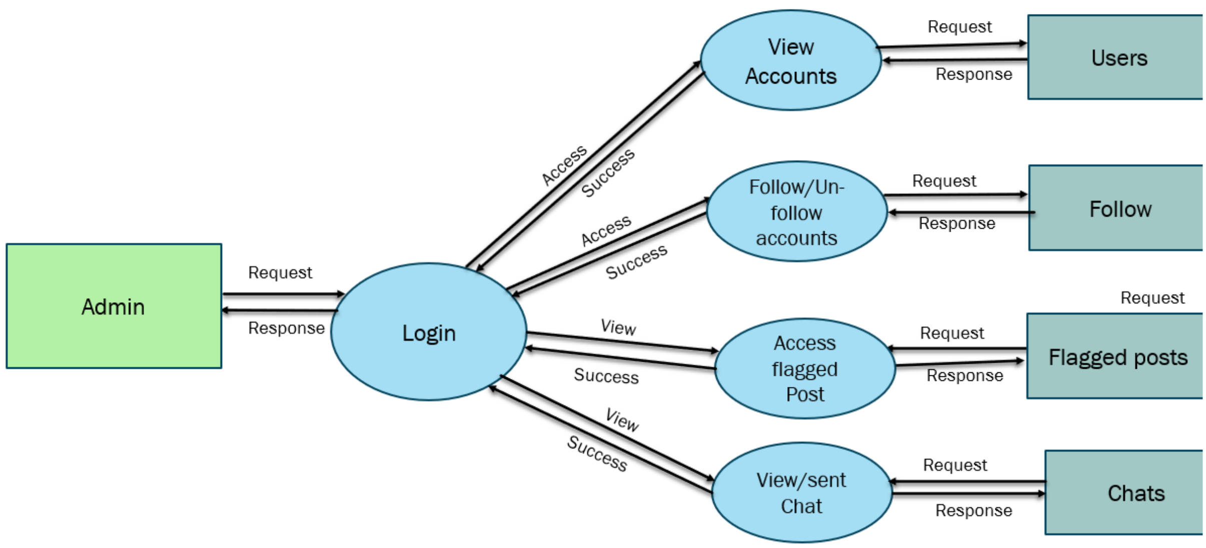


Figure 6

6.3 CNN Architecture

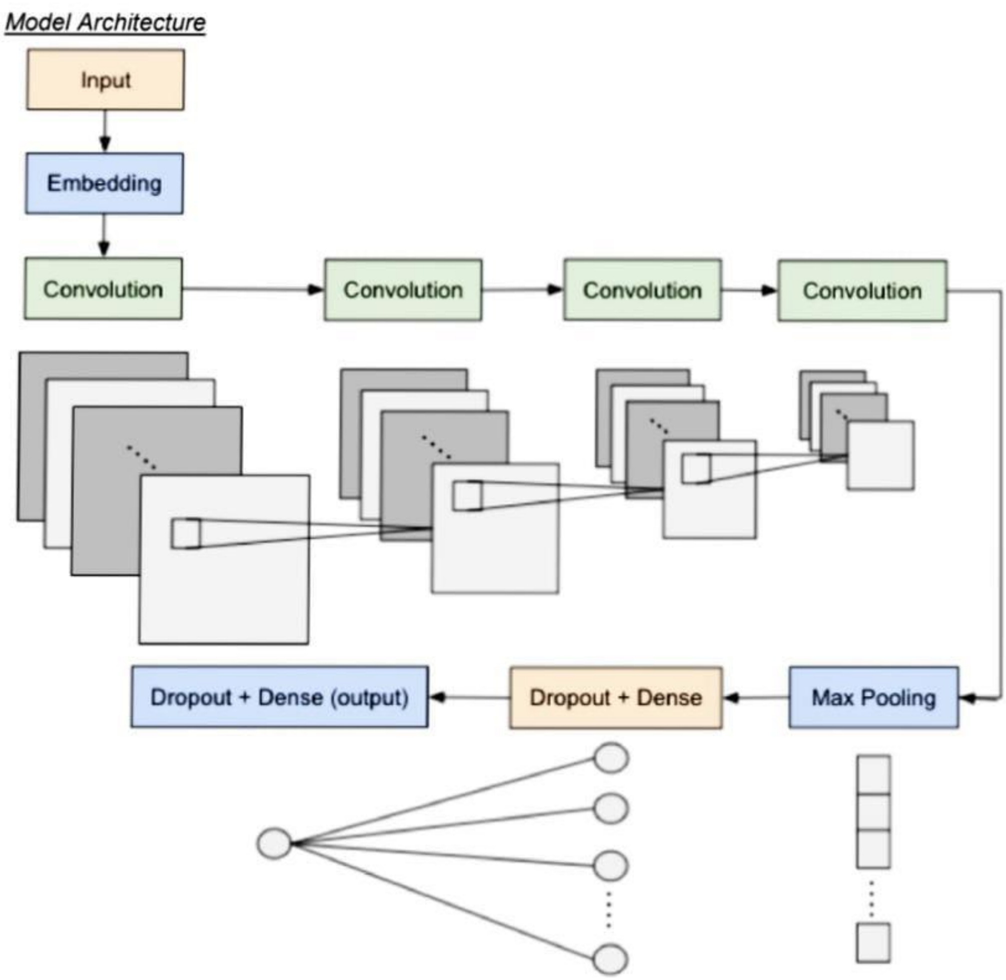


Figure 7

6.4 Implementation

Instello was created by utilising freely available open-source software. Rather than relying on native PHP, the backend was built using the Django framework, while HTML, CSS, and Bootstrap were used for the frontend. The development process was facilitated using the Visual studio Code integrated development environment for Django. The website's data is managed through SQLite, which serves as a database to store user information such as profiles, grades, and test results.

Login Page

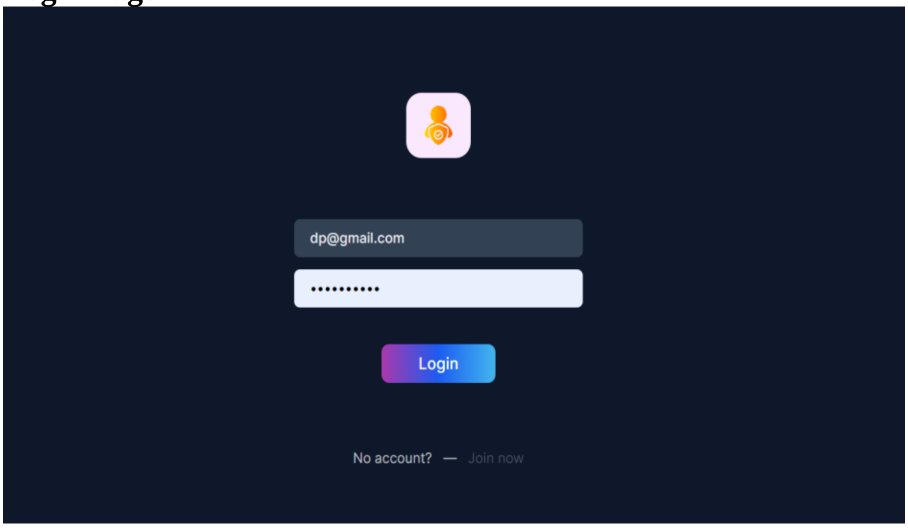


Figure 8

Home Page

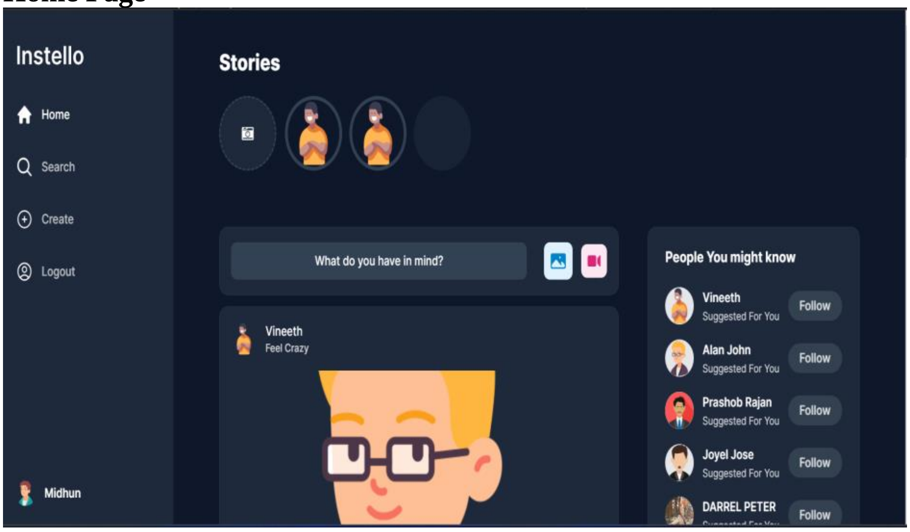


Figure 9

Creating a Post

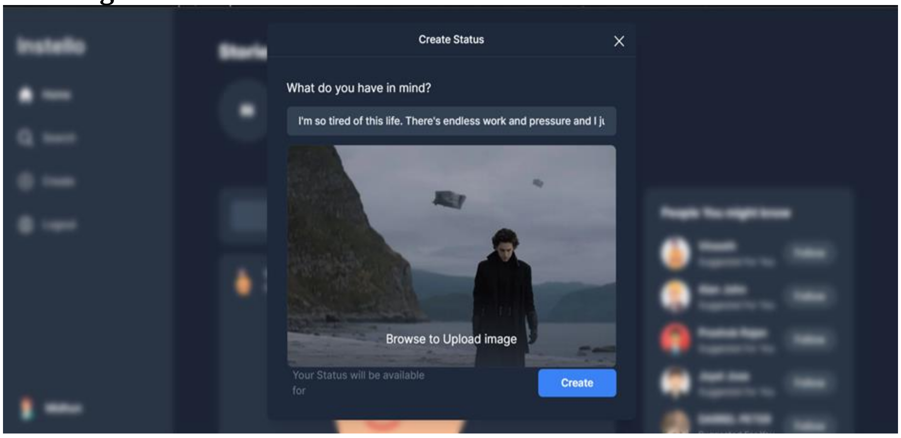


Figure 10

Post with suicidal risk content

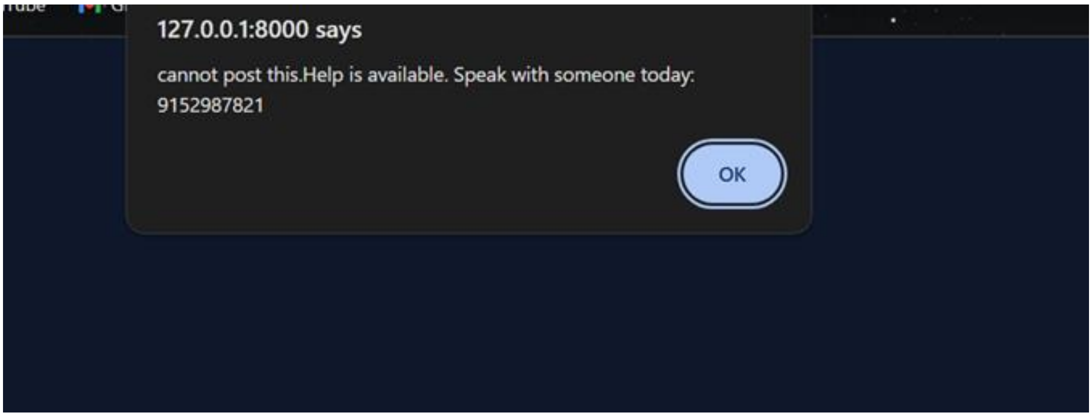


Figure 11

Notification to the admin

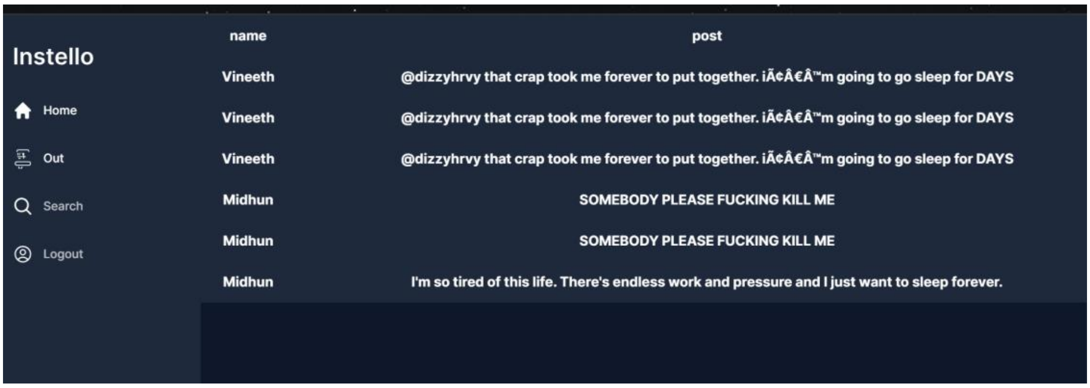


Figure 12

7. CONCLUSION

This project successfully investigated the efficacy of machine learning algorithms for classifying text data as indicative of suicidal ideation. The exploration revealed that a Convolutional Neural Network (CNN) achieved the highest accuracy (94.4%) compared to Naive Bayes (8.8%) and SVM (93.8%). This suggests that CNNs can effectively learn complex relationships within text data that signal potential suicide risk.

A limitation faced during the project was the prediction speed. While the current model can predict the nature of a post in about a minute, this timeframe might be too slow for real-time applications. This could be a critical limitation in situations where immediate intervention is necessary.

Future efforts will focus on optimizing the model and hardware to achieve faster predictions, potentially through model compression techniques or specialized hardware, ultimately aiming for real-time processing to enable quicker responses to those in crisis.

8. REFERENCES

- Wainberg, Milton L., et al. "Challenges and opportunities in global mental health: a research-to-practice perspective." Current psychiatry reports 19 (2017): 1-10

- Birjali, Marouane, Abderrahim Beni-Hssane, and Mohammed Erritali. "Machine learning and semantic sentiment analysis based algorithms for suicide sentiment prediction in social networks." Procedia Computer Science 113 (2017): 65-72.

- Boukil, Samir, et al. "Deep learning algorithm for suicide sentiment prediction." Advanced Intelligent Systems for Sustainable Development (AI2SD’2018) Vol 4: Advanced Intelligent Systems Applied to Health. Springer International Publishing, 2019
- Burnap, P., et al. "Multi-class machine classification of suicide-related communication on Twitter. Online Social Networks and Media, 2, 32-44." (2017).
- Kumar, E. Rajesh, and KVSN Rama Rao. "Sentiment Analysis using Social and Topic Context for Suicide Prediction." International Journal of Advanced Computer Science and Applications 12.2 (2021).
- Yang, Hui, et al. "A hybrid model for automatic emotion recognition in suicide notes." Biomedical informatics insights 5 (2012): BII-S8948
- Wang, Ning, et al. "Learning models for suicide prediction from social media posts." arXiv preprint arXiv:2105.03315 (2021).
- Whiting, Daniel, and Seena Fazel. "How accurate are suicide risk prediction models? Asking the right questions for clinical practice." BMJ Ment Health 22.3 (2019): 125-128.
- Madni, Hamza Ahmad, et al. "Improving Sentiment Prediction of Textual Tweets Using Feature Fusion and Deep Machine Ensemble Model." Electronics 12.6 (2023): 1302.
- Arowosegbe, Abayomi, and Tope Oyelade. "Application of Natural Language Processing (NLP) in Detecting and Preventing Suicide Ideation: A Systematic Review." International Journal of Environmental Research and Public Health 20.2 (2023): 1514.
- Babulal, Kanojia Sindhuben, and Bashu Kumar Nayak. "Suicidal analysis on social networks using machine learning." The Internet of Medical Things (IoMT) and Telemedicine Frameworks and Applications. IGI Global, 2023. 230-247.
- https://www.lexalytics.com/blog/machine-learning-natural-language-processing/