Ecole Supérieure Privée d'Ingénierie et de Technologies Monastir ,TUNIS

PFA

Mental health recognition from tweets

Élèves :
Prénom Nom
Prénom Nom
Prénom Nom

Encadrant :
Prénom Nom

Jury :
Prénom Nom
Prénom Nom

Année Universitaire 2024-2025

Abstract

Emotion recognition, a vital aspect of human communication, plays a pivotal role in shaping the future of human-machine interaction and artificial intelligence as it have a profound significance in our lives, elucidating its intricate relationship with artificial intelligence and its transformative impact on human-machine interaction. Emotions are fundamental to human existence, shaping our decisions, relationships, and overall well-being. Recognizing emotions accurately in ourselves and others is essential for effective communication, empathy, and social cohesion. In an increasingly digital world, integrating emotion recognition into technology becomes paramount. Artificial intelligence has witnessed remarkable advancements, particularly in the field of emotion recognition. AI-driven algorithms are now capable of discerning subtle emotional cues from facial expressions, vocal tone, and text, enabling machines to comprehend human emotions with remarkable accuracy. This development has far-reaching implications for various industries, including healthcare, customer service, and entertainment especially in the field of E-learning that the event such as covid19 force people to use distant study which posed a challenge to interact with each other and lack of supervising which push us to try and solve the problem using emotion recognition system. In the context of humanmachine interaction, emotion recognition bridges the gap between technology and humanity. Emotionally aware AI systems can adapt their responses and behaviors to align with the emotional states of users, enhancing the user experience and facilitating more meaningful interactions,in our work we focus in creating a web platform developed to allow doctors and administrators to log in and monitor their patients by reviewing and analyzing their tweets. This platform acts as a smart tool for emotional surveillance and mental health support.

This project combines advanced technology with social responsibility, contributing to the early detection and prevention of emotional distress.

KEY WORDS : Mentalhealth ,machine learning,deep learning,BERT,LSTM,BILSTM ,XLNET,tweets,api,NLP.

Liste des Abréviations

CNN : Convolutional Neural Network
LSTM : Long Short-Term Memory
BERT : Bidirectional Encoder Representations from Transformers
RNN : Recurrent Neural Network
NLP : Natural Language Processing
TER : Translation Edit Rate
HCI : Human-Computer Interaction
SER : Speech Emotion Recognition
BOW : bag of words
CLS : "[CLS] token." It is a special token used to represent the classification or aggregation of information in BERT models
SEP : [SEP] token. It is another special token used to separate two sentences in the input text.
NSP : Next Sentence Prediction
QA : Question Answering
NLI : Natural Language Inference
Svc support vector classifier

Introduction générale

With the exponential growth of social media, platforms like Twitter (now X) have evolved into vast repositories of human expression. Beyond sharing opinions and daily updates, users frequently articulate their emotions, frustrations, and moments of psychological distress, often in raw and unfiltered ways. This constant stream of public sentiment presents a unique and largely untapped opportunity to analyze emotional states in near real-time. Such insights hold immense potential value within the mental health domain, offering possibilities for earlier identification of concerning trends and augmenting traditional methods of care. The inherent challenge lies in interpreting the nuances of language used in these environments – often short, informal, context-dependent, and laden with slang or abbreviations. This project directly addresses this challenge by developing MindInsight, an intelligent system designed to analyze Twitter posts and classify the underlying emotional state. Leveraging advanced techniques in Natural Language Processing (NLP) and Artificial Intelligence (AI), specifically employing a fine-tuned bert-base-uncased model, the system aims to categorize tweets into five distinct states relevant to mental well-being : Normal, Stressed, Anxiety, Depression, and Potential Suicide Post. The choice of BERT provides a powerful foundation for understanding complex linguistic patterns even within brief social media texts. Recognizing the need to bridge sophisticated AI capabilities with practical clinical application, a core component of this project involved the development of a secure web platform. Built with a modern frontend interface communicating with a robust FastAPI backend and MongoDB database, this platform serves as a dashboard for authorized mental health professionals. It allows clinicians to securely manage patient profiles (linked to Twitter accounts with consent), initiate analyses, and monitor the classified emotional states derived from their patients' recent activity. The platform emphasizes clear data visualization, presenting overall emotional distributions (e.g., via pie charts), sentiment evolution over time, and lists of individual analyzed tweets with their predicted classifications, thereby facilitating easier interpretation of complex data. The system's analytical engine underwent rigorous evaluation, achieving a promising overall accuracy of 0.91 on a test dataset, with strong performance metrics (F1-scores) particularly for detecting 'Stressed' and 'Potential Suicide Post' categories. While acknowledging the prototype nature of the current system and the necessity for further refinement and crucial clinical validation, MindInsight demonstrates significant potential. By integrating cutting-edge AI for sentiment analysis with a user-centric platform for healthcare professionals, this work aspires to contribute meaningfully to increased mental health awareness, facilitate earlier detection of potential distress, and explore the possibilities of digitally-supported interventions, always prioritizing ethical considerations and data privacy.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Basic Concepts

1.1 Introduction

Emotion recognition, a field at the nexus of psychology, artificial intelligence, and human computer interaction, delves into the intricate world of human emotions. It represents a pivotal discipline with far-reaching applications, from improving mental health care to enhancing user experiences in the realm of technology. This introduction lays the foundation for exploring the fundamental concepts that underpin emotion recognition, shedding light on its core principles and significance. Emotions, the intricate tapestry of human feelings, are central to our daily lives, influencing our decisions, interactions, and overall well-being. They manifest through a range of expressions, including facial cues, vocal nuances, physiological responses, and textual cues, making them both complex and multidimensional. Emotion recognition, in its essence, involves the process of identifying and understanding these emotional cues in individuals. It requires deciphering subtle variations in facial expressions, recognizing changes in voice tone and pitch, and analyzing textual content for emotional content. The goal is to extract meaningful insights about a person's emotional state, enabling more empathetic and effective communication. At its core, emotion recognition draws inspiration from the rich body of research in psychology and neuroscience, which has sought to unravel the intricacies of human emotions for decades. By leveraging this knowledge, coupled with advancements in technology and artificial intelligence, researchers and practitioners in the field aim to develop algorithms and models capable of recognizing and responding to human emotions with increasing accuracy.

1.2 Mental health Recognition

Mental health refers to our emotional, psychological, and social well-being. It influences how we think, feel, and behave, and it plays a crucial role in how we handle stress, relate to others, and make choices. Mental health challenges can take many forms, including anxiety, depression, stress, and more severe conditions like bipolar disorder or schizophrenia. These challenges often involve overwhelming emotions, disrupted thinking patterns, and physiological responses such as fatigue, restlessness, or changes in appetite and sleep.[9]

Anxiety and stress are common aspects of mental health issues, both involving heightened alertness and physical symptoms like a racing heart or shortness of breath. However, while stress is usually a response to external pressures and builds gradually, anxiety can persist even in the absence of an obvious threat. Depression, another major mental health concern, is characterized by persistent sadness, hopelessness, and a loss of interest in activities. It often overlaps with anxiety and stress, and in more severe cases, can lead to suicidal thoughts—especially when individuals feel isolated or emotionally overwhelmed.

Mental health conditions rarely exist in isolation ; they are often interrelated and can

influence one another in complex ways. This interconnectedness makes accurate diagnosis and comprehensive treatment critical for recovery and long-term well-being.

1.3 Mental health Detection

Detection of mental health involves observing a combination of psychological, physiological, behavioral, and technological indicators. Clinically, it can be identified through structured interviews and standardized questionnaires like the Panic Disorder Severity Scale (PDSS) or the Beck Anxiety Inventory (BAI), which help assess the intensity and frequency of symptoms. Physiologically, signs such as increased heart rate, rapid breathing, sweating, trembling, and changes in skin conductance can indicate panic. Behaviorally, individuals may suddenly withdraw from situations, exhibit escape behaviors, or show signs of distress such as pacing or restlessness. With advancements in technology, panic can also be detected through artificial intelligence tools that analyze text, voice, and facial expressions to recognize emotional cues. For example, natural language processing (NLP) can identify panic-related language in messages, while wearable devices and mobile apps can monitor vital signs and alert users during sudden spikes in stress or fear. Together, these methods offer a comprehensive approach to recognizing and managing panic in both clinical and everyday settings.[10][11]

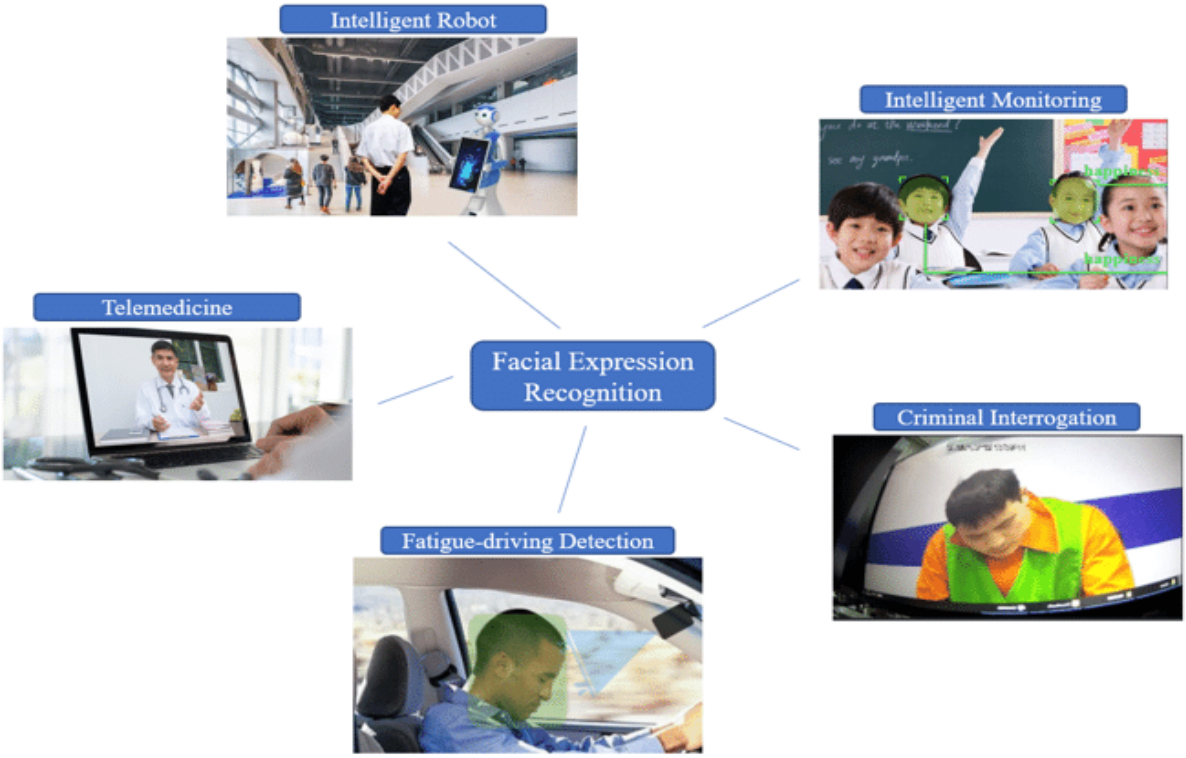

FIGURE 1.1 – Multi way of detection.

1.3.1 Multimodal emotion recognition

Multimodal emotion recognition is a technology or system that uses multiple sources of information, such as facial expressions, voice tone, gestures, and physiological data,

to accurately identify and understand the emotional state or feelings of a person. This approach combines various modalities to provide a more comprehensive and precise assessment of emotions compared to relying on a single source of data. It is often used in fields like human-computer interaction, psychology, and artificial intelligence to enhance the recognition and interpretation of human emotions.

1.3.2 Text emotion recognition

Text emotion recognition refers to the process of using natural language processing and machine learning techniques to analyze and determine the emotional content or sentiment expressed in written text.[12] It involves classifying text documents, such as emails, social media posts, or customer reviews, into different emotional categories such as happy, sad, angry, or neutral, based on the words, phrases, and context used in the text. Text emotion recognition is commonly applied in sentiment analysis, social media monitoring, customer feedback analysis, and other applications where understanding the emotional tone of text data is important for decision-making and insights.[13]

1.3.3 Speech emotion recognition

Speech emotion recognition is a technology or system that focuses on the identification and analysis of emotional states or sentiments expressed through spoken language. It involves using techniques from speech processing, natural language processing, and machine learning to detect and classify emotions conveyed in verbal communication. The goal is to categorize spoken words or phrases into various emotional categories such as happiness, sadness, anger, fear, and more. Speech emotion recognition has applications in fields like humancomputer interaction, customer service, mental health assessment, and voice assistants, where understanding the emotional state of a speaker can enhance the quality of interaction and communication.

1.4 field of application

Panic detection has valuable applications across various fields, especially in mental health, public safety, and technology. In clinical psychology and psychiatry, early identification of panic symptoms helps in diagnosing panic disorder, anxiety disorders, or comorbid conditions like depression and PTSD, allowing for timely intervention. In emergency response and public safety, panic detection systems can monitor crowds (e.g., via CCTV or drones) for signs of distress or mass panic during disasters, helping authorities respond faster. In the tech and wearable industry, smart devices and mobile applications integrate panic detection features—such as monitoring heart rate and voice tone—to alert users or caregivers during attacks. Human-computer interaction (HCI) and affective computing use panic detection to make machines more responsive to emotional states, improving user experience in gaming, virtual reality, or educational tools. Additionally, in social media analysis, panic-related language can be flagged to detect emotional crises or suicidal ideation, contributing to digital mental health surveillance. These applications show how panic detection plays a crucial role in improving well-being, safety, and emotional intelligence in modern systems[20].

1.5 Development Methodology

A development methodology is a structured framework based on principles, practices, and processes designed to organize the planning, management, and execution of a software project. It structures the various stages of the life cycle, from requirements gathering to the delivery of the final product[18].

1.5.1 Choice of Methodology

A selection of project management methodologies is proposed, which can be appropriately implemented to meet the specific requirements and objectives of the project.

1.5.2 Agile Methodology

Agile is an iterative and incremental methodology that prioritizes flexibility, adaptation to change, and the frequent delivery of features, organized into sprints. Each sprint allows for the development and testing of specific features, thus facilitating the rapid integration of technologies and the necessary adjustments to improve the user interface or the AI model.

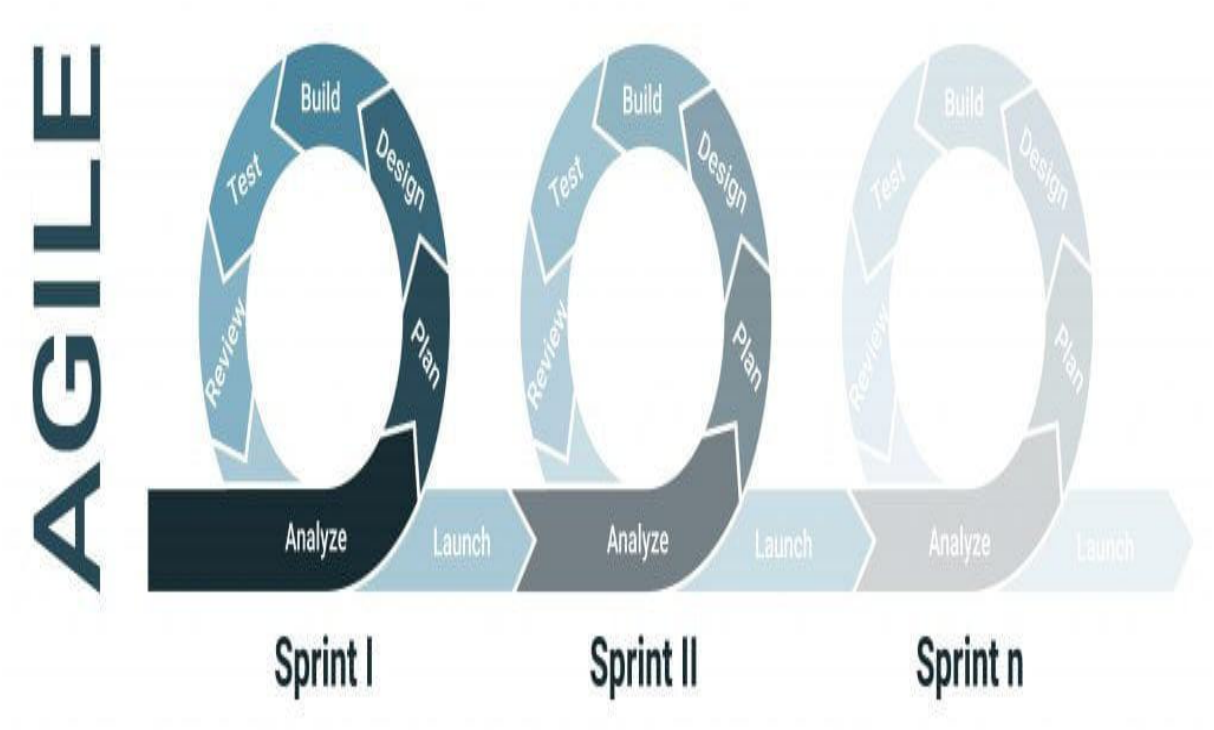

FIGURE 1.2 – Methodology AGILE.

1.5.3 CRISP-DM Methodology

CRISP-DM is a standard methodology for projects related to artificial intelligence and data analysis. It provides a well-defined structure for processing data, training and validating the machine learning model, and then effectively integrating it into the system. This

methodology is perfectly suited for the part of the project related to image interpretation with the model, ensuring a rigorous approach for data preparation, analysis, and result evaluation.

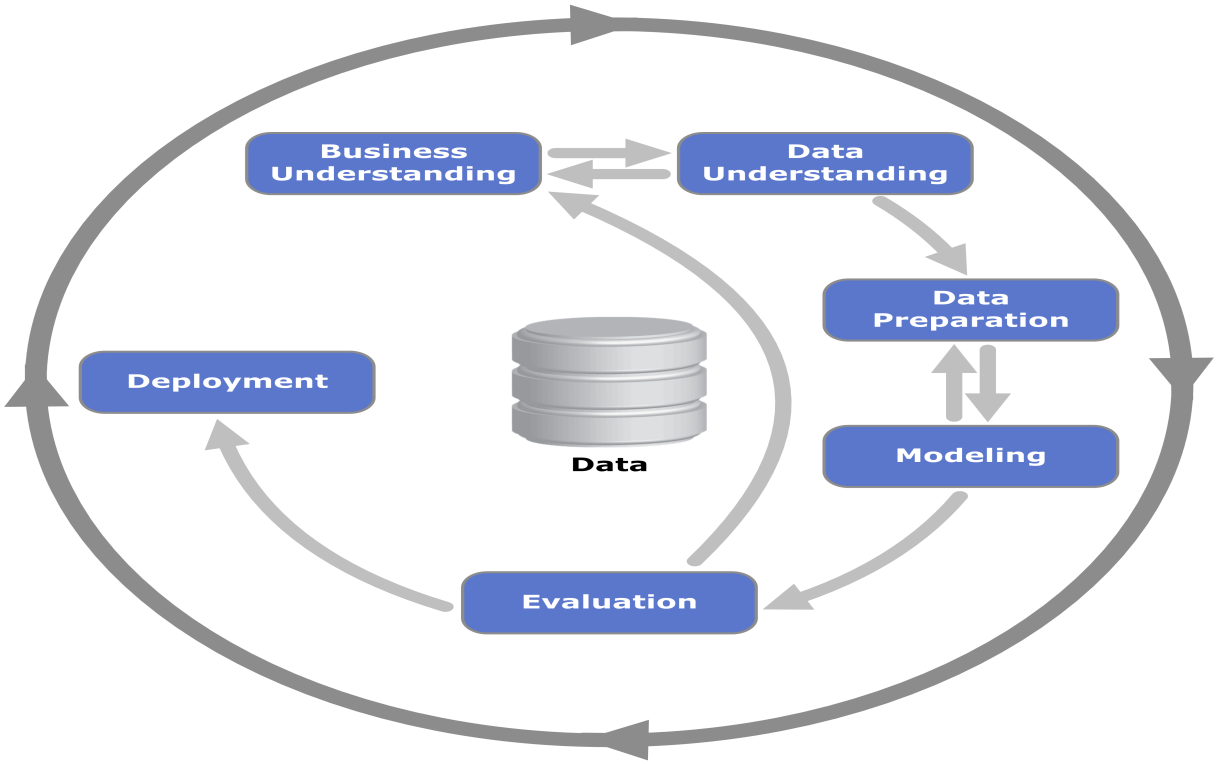

FIGURE 1.3 – Methodology CRISP-DM.

1.5.4 Integrated Agile and CRISP-DM Approach

The integrated Agile and CRISP-DM approach combines the flexibility of Agile with the structured methodology of CRISP-DM to effectively manage the microscope image analysis project. This combination allows for the development of an application that loads, analyzes, and stores images while respecting time, hardware resource constraints, and offline operation. It ensures iterative management and rapid adaptation to challenges, guaranteeing reliable and optimized results. The Gantt chart below illustrates the planning of the different project phases, following an iterative approach based on the Scrum method. Each sprint is aligned with a specific stage of the CRISP-DM model, covering activities from understanding objectives to the final deployment phase.

table below presents the alignment of each Agile sprint with the phases of the CRISP-DM process, as well as descriptions of the tasks and the estimated duration for each sprint. This allows for a clear visualization of how each project step is structured.

Agile Sprint	**CRISP-DM Phase**	**Description**	**Estimated Duration**
Sprint 1	Business Understanding	Project planning	1 week
Sprint 2	Data Understanding	Data collection (source analysis, data extraction)	3 weeks
Sprint 3	Data Preparation	Data preprocessing (cleaning, data augmentation)	8 weeks
Sprint 4	Modeling	Training and evaluation (initial training, performance evaluation)	4 weeks
Sprint 5	Evaluation	Final model selection (model comparison, validation)	1 week
Sprint 6	Deployment	Deployment (interface development, integration, testing, delivery)	7 weeks

1.6 Conclusion

Mental health recognition, whether in clinical settings or through technological tools, plays a crucial role in identifying and addressing mental disorders and emotional distress. It involves a combination of psychological assessments, physiological monitoring, and advanced technologies like natural language processing and AI-based emotion detection systems. By recognizing panic early, it becomes possible to provide timely intervention, whether through therapy, digital tools, or emergency response systems. The integration of methodologies such as Agile and CRISP-DM in the development of panic recognition systems ensures adaptability and efficiency, enhancing the system's ability to respond to the emotional state of individuals. These advancements are vital in improving mental health care, ensuring safety, and fostering well-being through proactive and responsive systems.

Chapitre 2

Related work

2.1 Introduction

Mental health recognition, particularly through social media platforms like Twitter, has become an important research area due to its implications for mental health monitoring, crisis response, and public safety. Researchers have increasingly explored natural language processing (NLP) and machine learning techniques to detect panic and related emotional states such as stress, anxiety, and depression from short, real-time textual data. These studies leverage diverse datasets, ranging from manually labeled tweets to large-scale corpora derived from crisis events or self-reported symptoms. Early works focused on basic classification techniques, including Support Vector Machines (SVM) and logistic regression, to differentiate between normal and panic-indicative tweets. More recent approaches have adopted deep learning architectures such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and transformer-based models like BERT, achieving higher accuracy and contextual understanding. These models often incorporate temporal patterns, semantic analysis, and even multimodal data to enhance detection accuracy. In parallel, domain-specific methodologies such as CRISP-DM and Agile development frameworks have been employed to guide the design and integration of panic recognition systems, particularly when embedded in intelligent applications or mental health platforms. This body of work demonstrates the increasing reliability and applicability of panic recognition systems, which now extend to early intervention, public health monitoring, and even suicide prevention efforts.

2.2 Text recognition related works

Mental health recognition in text is an important natural language processing (NLP) task that has numerous applications in different fields, including data mining, e-learning, information filtering systems, human-computer interaction, and psychology. Emotion detection, also known as emotion recognition, refers to the process of identifying a person's various feelings or emotions from text. Researchers have been working hard to automate emotion recognition for the past few years [14]. There are different methods for emotion detection from text, such as keyword spotting, lexical affinity, learning-based, and hybrid methods. The keyword spotting method involves identifying words from the text and matching them with keywords related to different emotions. Lexical affinity method uses affective lexicons to determine the emotion of the text. Learning-based methods employ machine learning algorithms to identify and recognize emotions from text. Hybrid methods combine two or more of the above methods to increase accuracy and efficiency. Emotion detection in text documents is essentially a content-based task. Several pre-processing techniques such as stemming, stop-words removal, and digit removal can be used to improve the accuracy of the emotion recognition system . Sentiment analysis and emotion

analysis are two related tasks. While sentiment analysis aims to detect positive, neutral, or negative feelings from text, emotion analysis aims to recognize types of feelings such as anger, disgust, fear, happiness, sadness, and surprise.in table fellow you can find more detail.[15]

Author(s)	**Year**	**Dataset**	**Model**	**Key Results**
Gruda & Hasan	2019	Tweets annotated for anxiety levels	SVM, Random Forest	Predicted perceived anxiety over time ; identified a negative correlation between anxiety and social engagement. [Link]
Reece et al.	2016	279,951 tweets from 204 users (105 depressed, 99 healthy)	Logistic Regression	Successfully predicted depression and PTSD onset ; outperformed GPs in diagnosing depression. [Link]
Nugroho et al.	2023	Tweets from Indonesian users	BiLSTM	Achieved 94.12% accuracy in detecting depression and anxiety. [Link]
Bendebane et al.	2023	Twitter data labeled for depressive and anxiety disorders	Deep Learning (multi-class)	Developed model for early detection of anxiety/depression. [Link]
Alam et al.	2018	Crisis-related tweets from real-world events	CNN, semi-supervised learning	Improved classification of crisis tweets ; leveraged unlabeled data. [Link]
Qu et al.	2021	Tweets during crisis events	CNN-LSTM, LSTM-CNN	LSTM-CNN outperformed SVM by 8.7% in crisis detection. [Link]
Chancellor et al.	2022	4,195 tweets (1,560 with chronic stress)	BERT, SVM, RF	BERT achieved 83.6% accuracy in detecting self-reported chronic stress. [Link]

2.3 Text limitation

Mental health recognition through text can be a challenging task as it involves identifying emotions expressed through the meanings of words and their relations. There are different levels of text emotion recognition, including document level, paragraph level, sentence level, and word level . As the level increases, the complexity of the problem increases. A survey of textual emotion recognition (TER) indicates that it has become an important topic in natural language processing due to its significant academic and commercial potential Textual emotion recognition can be approached by finding emotions from emo-

tion words, emoji, and extracted semantic or hidden emotion expressions . Researchers have used various techniques to tackle the task of learning to identify emotions from text, including a multi-layered neural network with hidden layers of neurons. Emotion detection in text documents involves content-based classification problems using concepts from the domains of natural language processing and machine learning. Emotion recognition systems assume a small number of distinct and universal emotional categories [14]. To detect emotions in text, natural language processing techniques, machine learning, and computational linguistics are used. Emotion detection in text documents is essentially a content-based classification problem involving concepts from natural language processing as well as machine learning . The limit of emotion recognition using text can be subjective due to the subjectivity of language and phenomena such as sarcasm, irony, and cultural references. Therefore, the accuracy of emotion recognition systems can vary depending on the dataset used and the algorithm training[16].

2.4 Conclusion

Mental health recognition from social media platforms like Twitter represents a promising and evolving area of research at the intersection of artificial intelligence and mental health. The reviewed studies demonstrate that advanced machine learning and deep learning models can effectively detect panic-related emotions such as anxiety, stress, depression, and suicidal ideation. These systems not only offer valuable tools for early diagnosis and intervention but also support large-scale mental health monitoring and crisis management. As methodologies continue to improve in terms of accuracy, context awareness, and real-time processing, panic recognition has the potential to become an integral part of intelligent health systems, offering timely insights and preventive support for vulnerable individuals and communities.

Chapitre 3

Proposed Approaches

3.1 Introduction

Emotion recognition, a pivotal field in artificial intelligence and human-computer interaction, seeks to identify and comprehend human emotions through various sensory channels, including speech, text, and facial expressions. The accuracy of emotion recognition has far-reaching implications across diverse domains, spanning healthcare, customer service, education, and entertainment. In recent years, the adoption of multimodal approaches, which amalgamate information from multiple sources, has emerged as a promising strategy to enhance both accuracy and precision in these systems. Multimodal emotion recognition, a cornerstone of this evolution, entails the fusion of data from different sensory modalities such as audio, visual, and textual information. By synergizing multiple sources of data, these approaches aim to capture a more comprehensive and nuanced understanding of human emotions. Leveraging information from diverse modalities, such as spoken words, facial expressions, and textual content, equips these systems to provide more robust and dependable emotion recognition. Speech-based emotion recognition, another significant facet, involves the analysis of acoustic features, including pitch, intensity, and speech patterns, to identify emotional cues in spoken language. Modern machine learning, especially deep learning, has drastically improved the accuracy of speech-based emotion recognition. This has had a profound impact, allowing real-time emotional analysis for applications such as call center quality monitoring and mental health support. Text-based emotion recognition, on the other hand, focuses on extracting emotional content from written or typed text, primarily through natural language processing (NLP) techniques [56]. Sentiment analysis, emotion classification, and context extraction are common methods used in this domain. Text-based emotion recognition finds applications in diverse areas such as social media sentiment analysis, chatbots, and personalized content recommendations. Its impact lies in its ability to understand and respond to user emotions in text-based interactions.

3.1.1 Mental health recognition system

In this project, the primary objective is to build a smart web application capable of detecting panic levels in tweets and classifying them into one of five categories : Normal (no panic), Stress, Depression, Suicidal, and Anxiety[7]. To achieve this, a multi-step pipeline was designed, integrating data collection, preprocessing, feature extraction, model training, and deployment into a user-friendly web application.Text emotion recognition leverages various NLP methods, including machine learning algorithms and linguistic analysis, to analyze language patterns, context, and the choice of words to determine the underlying sentiment. This technology finds applications in customer feedback analysis, social media monitoring, market research, and personalized content recommendation systems, among others, where understanding the emotional content of text is essential for

making informed decisions and providing more tailored user experiences[8][6]. Below, we detail the proposed approaches and methodologies used throughout the project.

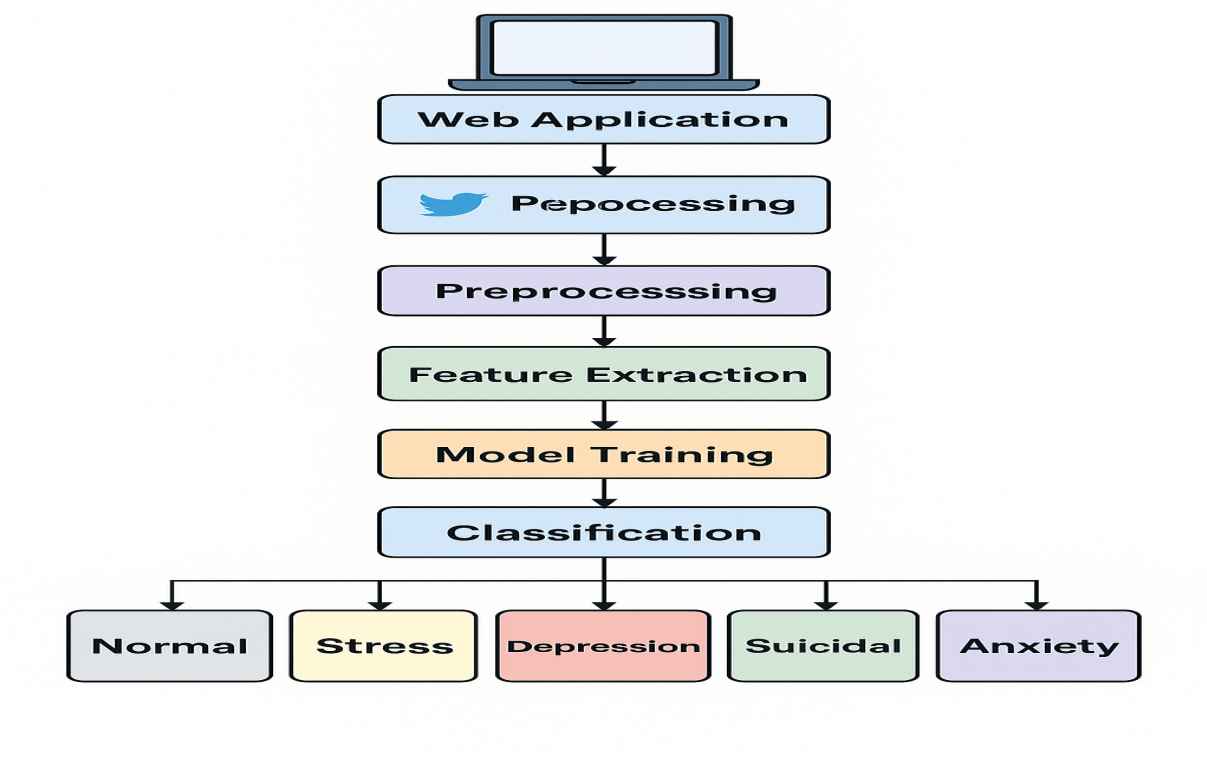

FIGURE 3.1 – Panic recognition system.

3.2 Data Collection

The foundation of the system lies in the quality and relevance of the data. For this project, data was sourced from Twitter using a tweet scraping mechanism. The objective was to gather real-world, user-generated content that reflects the emotional and psychological states of users. The scraping system allows the user to input keywords or hashtags related to mental health and retrieves a set of tweets associated with those terms.[17]

To ensure diversity and relevance, tweets were collected using keywords such as : depression, anxiety, mentalhealth, stress, suicidal, etc. Each tweet is associated with metadata such as timestamp, user ID, and location (if available), but only the text content was used for classification.

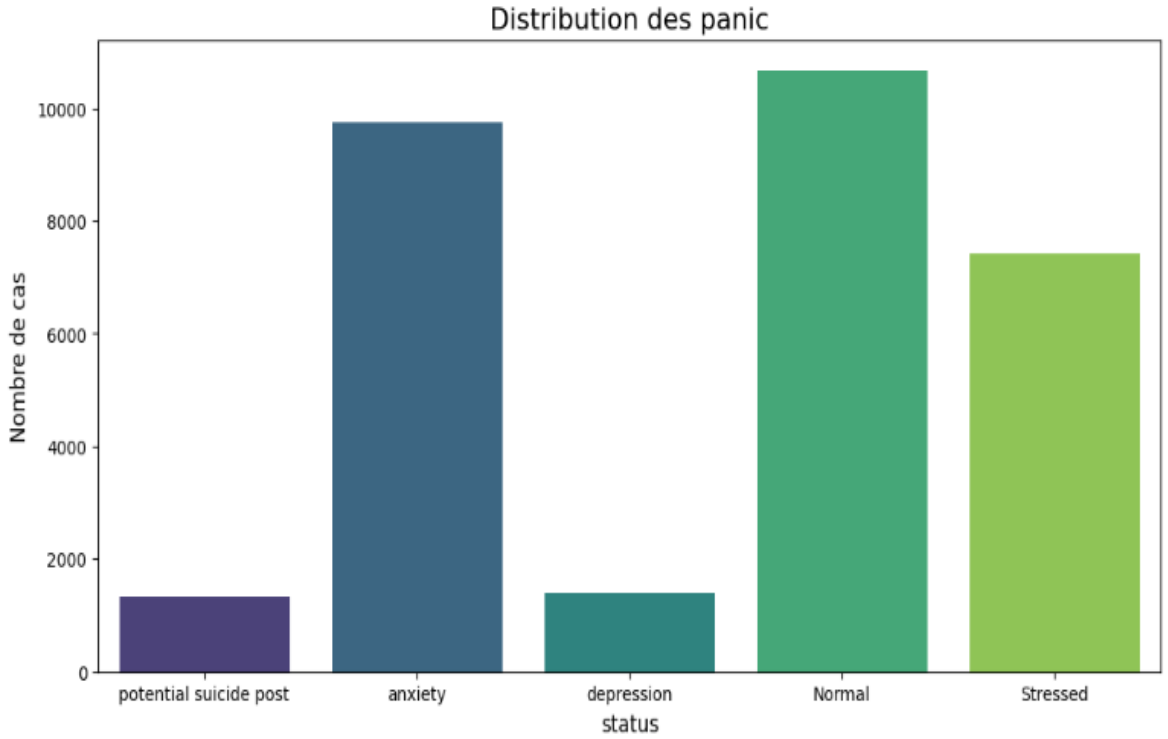

FIGURE 3.2 – Panic database.

3.3 Dataset Pre-Processing

Pre-processing plays a crucial role in machine learning systems as it involves cleaning and refining the dataset to prepare it for subsequent classifier training. In our dataset, YouTube comments are written in different Arabic dialects rather than standard Arabic. To address the challenges posed by Arabic language comments, we utilized NLP techniques that encompass normalization, stemming, and stop-word removal. These techniques were employed to ensure the effective processing of the data and improve the quality of the dataset for further analysis [33].

3.3.1 Noise Removal

Noise removal is a valuable technique that not only speeds up analysis but also enhances accuracy. Noise refers to elements such as hashtags, URLs, and repeated letters. The first step in noise removal is to eliminate URLs and HTML tags. This is achieved by searching for the "https ://" followed by any string of characters and the tags, using a specific regular expression. The "sub" method from the "re" package is then utilized to perform the removal. The next step involves getting rid of symbols, noise, and frequently used symbols like "@" and ".". To accomplish this, a dedicated regular expression is constructed and applied using the "sub" method from the "re" package. Lastly, to eliminate repeated letters, another specific regular expression is created and applied using the "sub" method from the "re" package. This ensures the removal of any instances of repeated letters. By implementing these steps, the noise is effectively removed from the data, allowing for faster analysis and improving the overall accuracy of the process.

3.3.2 Normalization

The normalization process serves two purposes : removing noise from the data and correcting spelling errors in Arabic. This step has proven to significantly enhance model accuracy in various studies. For instance, Huq et al. [8] achieved improved scores by implementing normalization and other preprocessing techniques. In their work, the accuracy score for the SVM classifier model increased from 0.61 to 0.8. To achieve normalization, we employed a script that specifically addresses certain English letter variations. The script initially eliminates leading or trailing whitespace in the text. It then replaces Arabic character variants with their standardized versions using regular expressions. By applying these normalization methods, we ensure consistent representation of Arabic characters, which proves beneficial for text processing and analysis tasks.as it is shown in figure

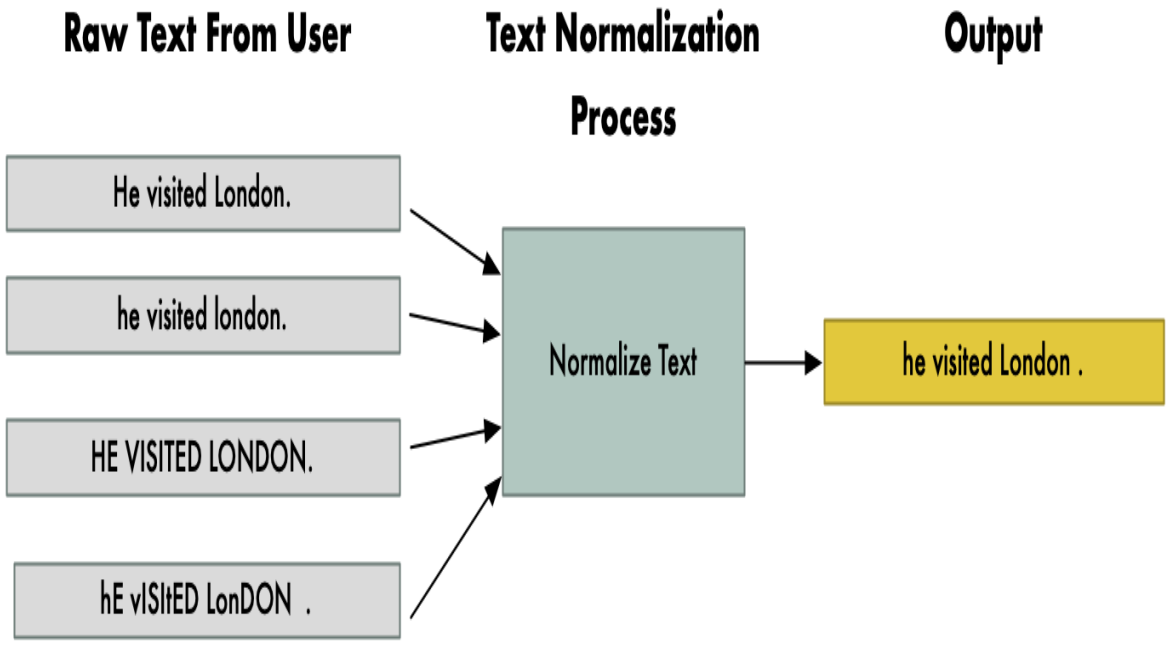

FIGURE 3.3 – Text Normalization.

3.3.3 Tokenization

Tokenization is a crucial initial step in text analysis as it helps reduce variations in word typos [9]. Furthermore, performing tokenization aids in improving performance when utilizing feature extraction techniques like bag of words (BOW) [10]. Various factors can influence tokenization, including N-gram, co-occurrence, and stemming. In tokenization, spaces, tabs, and newlines are typically treated as delimiters. Other characters such as parentheses, question marks, and exclamation marks can also serve as delimiters. It is worth noting that Arabic users often express their opinions in informal Arabic on social media platforms. Consequently, the use of commas can be inconsistent, and it is advisable to consider them as delimiters if they appear between words. One of the primary applications of tokenization is to calculate the frequency of word occurrences. Tokenization is an

integral step that facilitates subsequent text analysis by reducing typographical variations [9]. It also aids in performance improvement when utilizing feature extraction techniques like bag of words (BOW) [10]. Factors such as N-gram, cooccurrence, and stemming can affect tokenization. During this process, spaces, tabs, and newlines are commonly used as delimiters. Other characters like parentheses, question marks, and exclamation marks can also serve as delimiters. In the context of Arabic text, informal language usage on social media platforms may result in inconsistent comma usage. Considering commas as delimiters when they appear between words can help overcome this issue. An important application of tokenization is the calculation of word frequencies, see figure

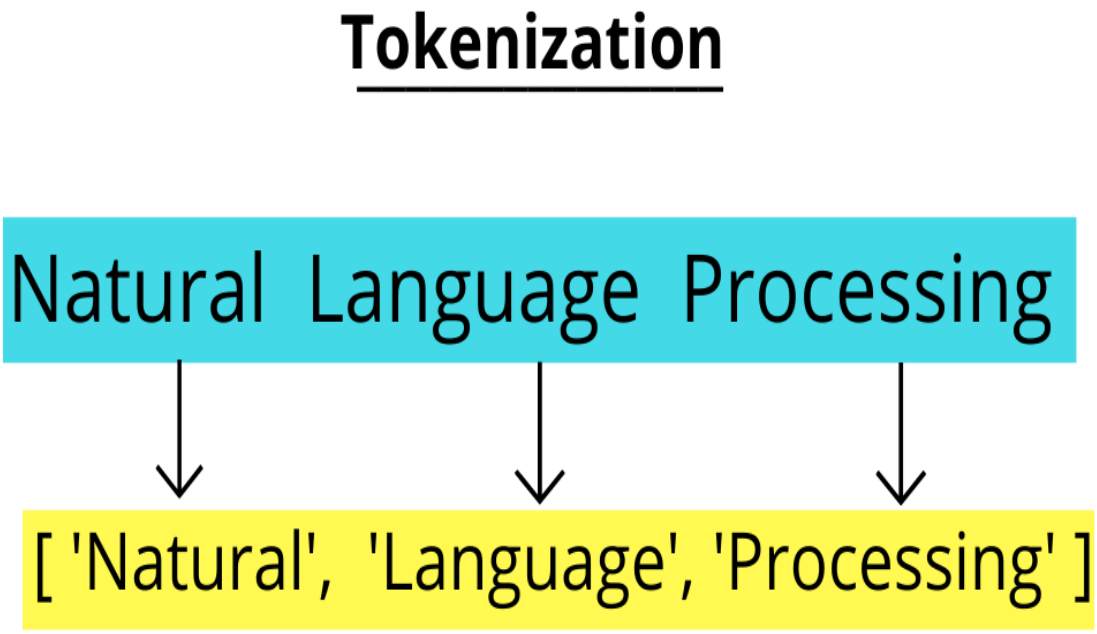

FIGURE 3.4 – Text Tokenization .

3.3.4 Stop-word

Stop-word removal involves two sources of stop words. The first source is obtained from the "NLTK" library using the "stopwords" class and selecting the Arabic language. The second source involves manually writing stop words directly in the code. Both sources are prepared to be used for stop-word removal. Subsequently, the comments are tokenized, separating the words within them. Each word is then compared with the stop words from the two sources mentioned earlier. If a match is found, the word is considered a stop word and therefore removed from further analysis. The process continues with the loops iterating through each word of the comments, performing the comparison until the last word is processed, as it is shown in figure

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

FIGURE 3.5 – Text Stop-words.

3.3.5 Stemming

Stemming is a process that involves removing prefixes and suffixes from words to restore them to their original root form. This enables the word to retain its essential meaning. To perform stemming, we utilized the ISRIStemmer provided by the NLTK library, as it is shown in figure

	raw_word	cleaned_word	stemmed_word
0	..trouble..	trouble	troubl
1	trouble<	trouble	troubl
2	trouble!	trouble	troubl
3	<a>trouble	trouble	troubl
4	1.trouble	trouble	troubl

FIGURE 3.6 – Text Stemming.

3.3.6 Emoji

Emoji often provide additional contextual information that enhances the understanding of a text-based message. Text-based communication lacks nonverbal cues present in face-to-face interactions, such as facial expressions and body language. Emoji act as compensatory signals, helping to bridge this gap by conveying emotions and attitudes. Recognizing and interpreting emoji alongside textual content can lead to more accurate emotion recognition and sentiment analysis.in this work we replace emoji with text, see figure 10 for more detail.

```
emoji_dict = {
    # Original entries
    "😀": "HAPPY", "😊": "HAPPY", "😎": "COOL", "🤗": "HUG", "😌": "RELAXED",
    "🙃": "AMUSED", "💪": "STRENGTH", "😟": "WORRIED", "😱": "PANIC",
    "😠": "SAD", "😰": "ANXIOUS", "😫": "FRUSTRATED", "😞": "DISAPPOINTED",
    "😕": "UNEASY", "😒": "DISPLEASED", "😤": "FRUSTRATED", "  ": "EMBARRASSED",
    "🤯": "EXTREME STRESS", "😕": "CONFUSED", "😢": "SAD", "💔": "HEARTBROKEN",
    "😕": "CONFUSED", "😳": "SHOCKED", "😮": "SURPRISED", "😡": "ANGRY",
    "😨": "SCARED", "💃": "DANCING",

    # New additions
    "😍": "LOVE", "🤩": "EXCITED", "😨": "FEAR", "😬": "PAIN", "😪": "TIRED",
    "😷": "SICK", "🥵": "HOT", "🥶": "COLD", "😵": "DRUNK", "😈": "MISCHIEVOUS",
    "👹": "MONSTER", "👺": "EVIL", "🤠": "ADVENTUROUS", "🥳": "CELEBRATING",
    "🤑": "GREEDY", "😑": "UNIMPRESSED", "😶": "SPEECHLESS", "😐": "NEUTRAL",
    "🙄": "SARCASM", "😏": "SMUG", "😋": "HUNGRY", "😋": "TASTY", "🥱": "BORED",
    "🤫": "SECRETIVE", "😲": "SURPRISED", "🧐": "CURIOUS", "😮‍💨": "RELIEF",
    "😵‍💫": "DIZZY", "🥺": "PLEADING", "🙏": "PRAYING", "✨": "MAGIC",
    "❤️": "LOVE", "💥": "EXPLOSION", "💫": "DIZZY", "🎉": "CELEBRATE",
    "🍕": "HUNGRY", "☕": "CAFFEINE", "🐶": "PET", "🐱": "CAT", "🌞": "SUNNY",
    "🌧️": "RAINY", "🌈": "HOPE", "🎂": "BIRTHDAY", "🎓": "GRADUATE",
    "🏆": "WIN", "⚽": "SPORTS", "🏋️": "WORKOUT", "🧘": "ZEN", "🚀": "LAUNCH",
    "💡": "IDEA", "🔥": "FIRE", "🕶️": "COOL", "🎧": "MUSIC", "🎮": "GAMING",
    "📚": "STUDYING", "🛌": "REST"
}
```

FIGURE 3.7 – Emoji documentation.

3.3.7 abbreviation resolution

Abbreviation expansion is an essential step in the text preprocessing phase, especially when dealing with informal and user-generated content such as tweets. Social media users often rely on abbreviations, acronyms, and slang to express themselves concisely. These shortened forms can hinder the performance of natural language processing models if not properly expanded to their full meanings. For example, terms like "OMG" (Oh my God), "IDK" (I don't know), or "SMH" (Shaking my head) carry significant emotional or contextual weight that may be lost if left unresolved. In this project, a dedicated abbreviation expansion step was included to normalize such terms and ensure that the model can accurately interpret the underlying emotional tone. This process improves the quality of the input data and enhances the model's ability to detect and classify panic-related content effectively.

```
{'LOL': 'Laugh Out Loud',
 'BRB': 'Be Right Back',
 'IDK': "I Don't Know",
 'IMO': 'In My Opinion',
 'BTW': 'By The Way',
 'TTYL': 'Talk To You Later',
 'OMG': 'Oh My God',
 'FYI': 'For Your Information',
 'ASAP': 'As Soon As Possible',
 'GTG': 'Got To Go',
 'TTYT': 'Talk To You Tomorrow',
 'TMI': 'Too Much Information',
 'IMHO': 'In My Humble Opinion',
 'ICYMI': 'In Case You Missed It',
 'AFAIK': 'As Far As I Know',
 'FAQ': 'Frequently Asked Questions',
 'TGIF': "Thank God It's Friday",
 'FYA': 'For Your Action'}
```

FIGURE 3.8 – Abbreviation resolution.

3.4 Prediction Models

Several machine learning and deep learning models were explored to classify tweets into the five categories. These included :Classical Machine Learning Models,Deep Learning Models and Transformer-based Models

3.4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers)[1] is a pre-trained model that learns to understand the contextual meaning of words and sentences by training on a large corpus of text. It uses a transformer architecture, which is a type of neural network that excels at capturing long-range dependencies in sequential data, such as language . Unlike previous models that only considered the left or right context of a word during training, BERT leverages both left and right contexts simultaneously. This bidirectional training enables BERT to have a deeper understanding of the context in which words appear, leading to better performance on downstream NLP tasks.

Bert Model Architecture

Training BERT should be in two phases. The first phase is pre-training, where the model understands language and context, and the second phase is fine-tuning, where the model learns the language and how to solve the problems [5]. BERT can solve many issues, such as Neural Machine Translation, Question Answering, Sentiment Analysis, Text summarization, etc. Also, BERT achieved state-of-the-art outcomes in more than

11 NLP tasks, the first phase is pre-training unsupervised datasets simultaneously by two techniques. The first is masking out some percentage of the words in the input and then conditioning each word bidirectionally to predict the masked words with Masked Language Modeling (MLM). In this stage, they used WordPiece embedding, which is sub-word tokenization that enables the model to process the unknown words by decomposing them into known subwords, every embedding token has a particular classification token at the beginning of every sentence [CLS] and uses [SEP] to separate them. Also, to help the model differentiate among the different sentences, they add a learned embedding indicating sentence A or sentence B is added to each token, which is segment embedding. In this process, E has been marked as input embedding at the first hidden vector of [CLS] token .

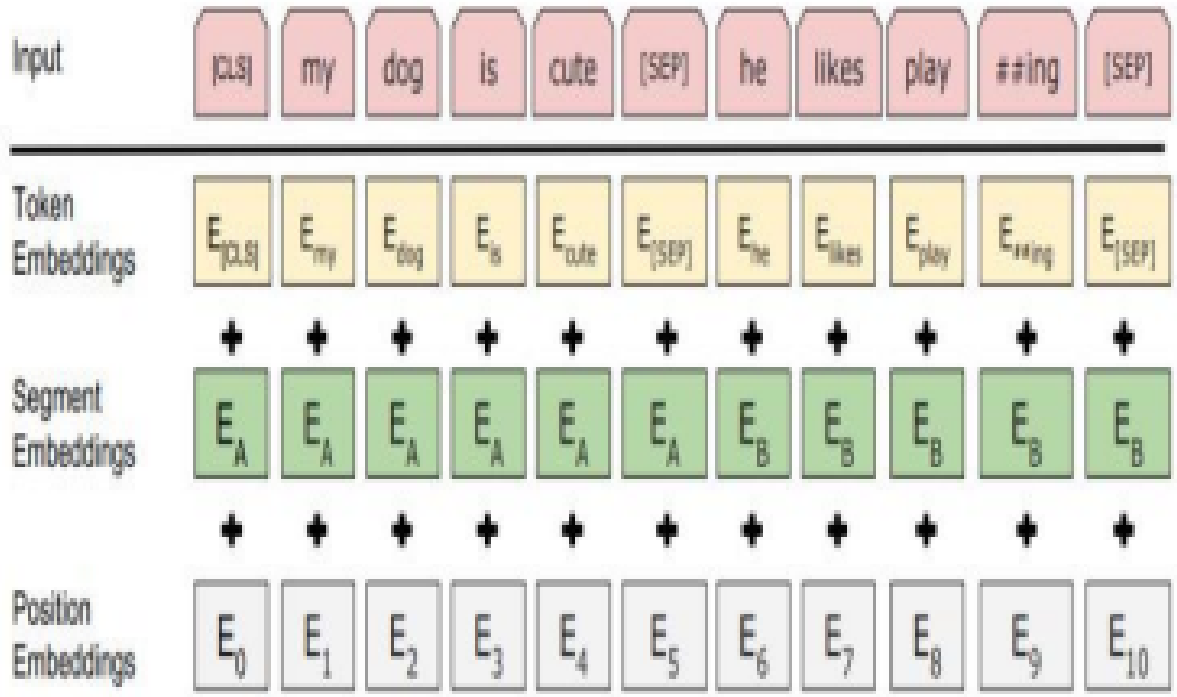

FIGURE 3.9 – Bert architecture.

The second technique is understanding the relationship between sentences, which is essential in this model on Question Answering (QA) and Natural Language Inference (NLI), by applying the Next Sentence Prediction (NSP) classification task to predict whether sentence B immediately follows sentence A. For each pre-training example, the sentences A and B are chosen randomly from the corpus, with 0.5 of the time B being the sentence that follows A (labeled as IsNext) and 0.5 being a random sentence (labeled as NotNext), The capability of the transformer's self-attention mechanism to simulate many downstream tasks, whether they entail single texts or pairs of texts, by simply swapping out the appropriate inputs and outputs makes fine-tuning simple. As encoding a concatenated text pair with self-attention effectively involves bidirectional cross-attention between two sentences, BERT uses this method to combine both stages. Feed the inputs and outputs particular to each task into BERT and fine-tune all the parameters end-to-end [35]. The second phase is fine-tuning the model, where the pre-trained parameters are

used to initialize the BERT model and labeled data from the downstream tasks is used to fine-tune each parameter. Represents the overall pre-training and fine-tuning procedures for BERT. In both pre-training and fine-tuning, the same architectures are used as given in figure .

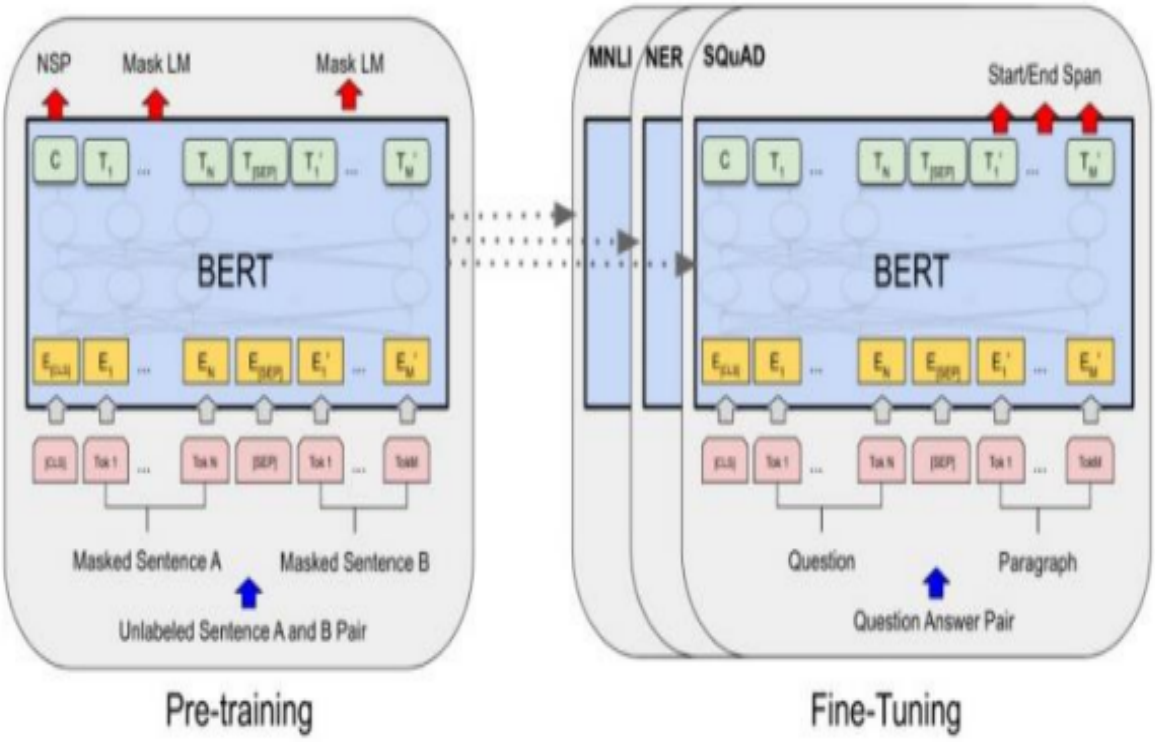

FIGURE 3.10 – Bert fine-tuning.

Bert uncased

BERT-Base-Uncased refers to one of the variants of the BERT (Bidirectional Encoder Representations from Transformers) model introduced by Google researchers. It is trained on a large corpus of text and is widely used in various natural language processing (NLP) tasks [35]. BERT-Base-Uncased has a vocabulary size of 30,522 tokens, including words, subwords, and special tokens. It is trained using a masked language modeling (MLM) objective, where random words are masked in the input text, and the model learns to predict the masked words based on the surrounding context BERT-Base-Uncased is a powerful NLP model that can understand the contextual meaning of words and sentences, making it a valuable tool for various NLP applications and research in the field of emotion recognition.

MentalBERT

MentalBERT is a domain-specific language model based on BERT, pre-trained on mental health-related text data. It was designed to better understand and analyze content related to mental health, which typically includes sensitive language, psychological terms, and expressions laden with emotions.

TwitBERT

TwitBERT is a BERT-based language model pre-trained specifically on Twitter data. Since Twitter language is highly informal, full of abbreviations, hashtags, emojis, and slang, TwitBERT is fine-tuned to handle such characteristics effectively.

3.4.2 LSTM with Word Embedding

Long Short-Term Memory (LSTM)[2] is a type of Recurrent Neural Network (RNN) that is specially designed to capture long-range dependencies and patterns in sequential data, such as natural language. In the context of text classification, LSTM is highly effective at understanding the context and flow of words in a sentence. When combined with Word Embedding—a technique that transforms words into dense vectors that capture semantic relationships—the model can learn meaningful patterns in textual data. This architecture is widely used in NLP tasks such as sentiment analysis, emotion detection, and mental health classification.

Architecture

The typical architecture of an LSTM with Word Embedding model includes the following layers :

1. Input Layer :Receives tokenized and padded sequences of text (tweets in this case).Each tweet is converted into a fixed-length sequence of integers (word indices).
2. Embedding Layer :Maps each word index to a dense vector of fixed size (e.g., 100 or 300 dimensions).This layer can use pre-trained embeddings (e.g., GloVe, Word2Vec) or learn embeddings during training.Captures semantic similarity between words.
3. LSTM Layer :Processes the sequence of word embeddings while maintaining memory of previous words through its internal cell state.Handles variable-length context and preserves sequential information.Capable of learning temporal patterns and emotional flow in text.
4. Dropout Layer (Optional) :Added after the LSTM to prevent overfitting.Randomly deactivates a percentage of neurons during training.
5. Dense Layer(s) :Fully connected layers that interpret the high-level features extracted by the LSTM.Can include activation functions such as ReLU.
6. Output Layer :Uses a Softmax activation function for multi-class classification.Outputs a probability distribution over the five panic categories : Normal, Stress, Depression, Suicidal, and Anxiety.

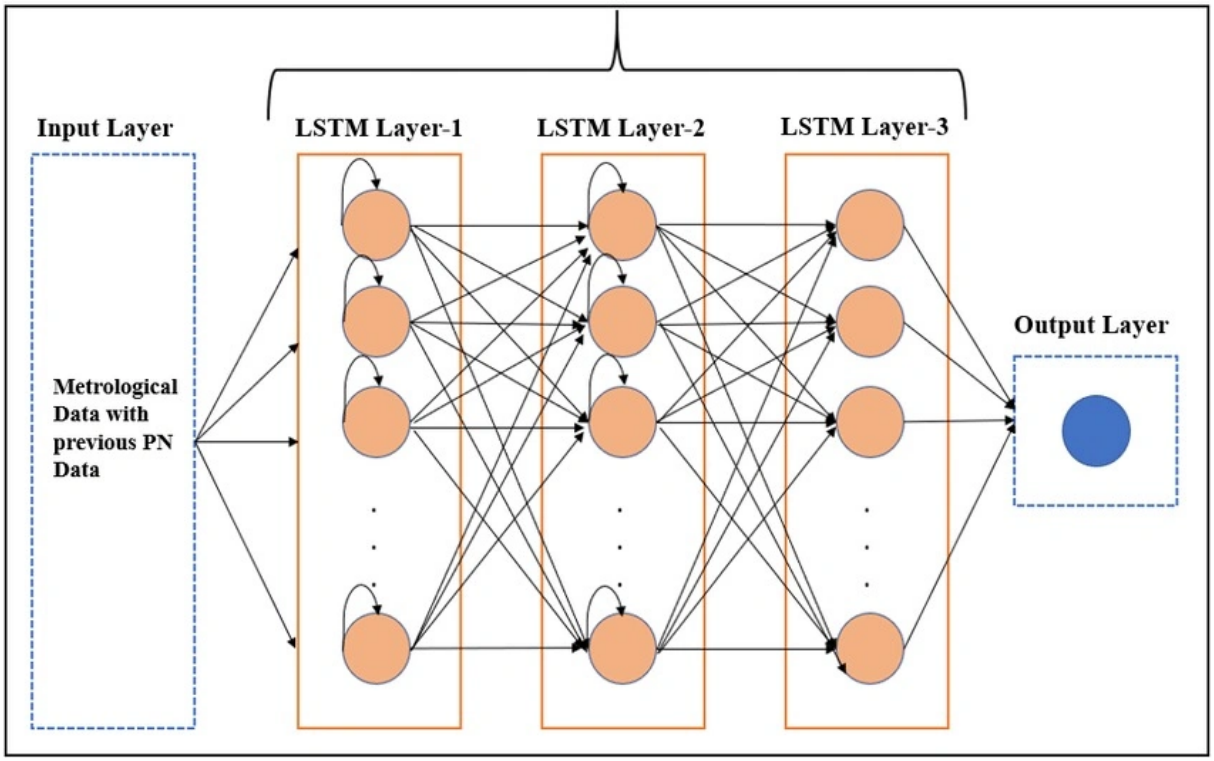

FIGURE 3.11 – LSTM Architecture.

3.4.3 XLNet

XLNet [4] is a transformer-based language model developed by Google and Carnegie Mellon University. It builds upon and improves the popular BERT model by addressing some of its limitations, particularly in how it handles word order and context. Unlike BERT, which uses a masked language modeling objective (randomly masking words during training), XLNet uses a permutation-based language modeling approach that allows it to better capture the bidirectional context of words without masking. This makes XLNet more powerful in understanding complex dependencies and sentence structures.

In the context of mental health detection or panic classification, XLNet can effectively understand the nuanced language used in tweets and learn deeper emotional and psychological patterns in text.

Architecture

The XLNet architecture is based on the Transformer-XL framework, which allows it to model long-range dependencies and sequences more efficiently. Here's a breakdown of its main components :

1. Input Representation : Input tokens are embedded into vectors (just like in BERT). Segment embeddings and positional embeddings are added to preserve word positions.
2. Permutation Language Modeling : Instead of masking random words like BERT, XLNet randomly permutes the order of input tokens and predicts the next token based on previous ones. This allows the model to learn from all possible factorization orders, improving its bidirectional understanding.

3. Transformer-XL Backbone : Incorporates recurrence between segments, allowing the model to remember information across longer text spans. Enables better modeling of long-term dependencies and context.
4. Multi-Head Attention : Like other transformer models, XLNet uses self-attention to capture relationships between all tokens in the input sequence, regardless of distance.
5. Feed-Forward Networks : Applies a fully connected neural network to the output of each attention layer for deep representation learning.
6. Output Layer : For classification tasks, the final hidden state is passed through a dense + softmax layer to produce the predicted class (e.g., Normal, Stress, Depression, Suicidal, Anxiety).

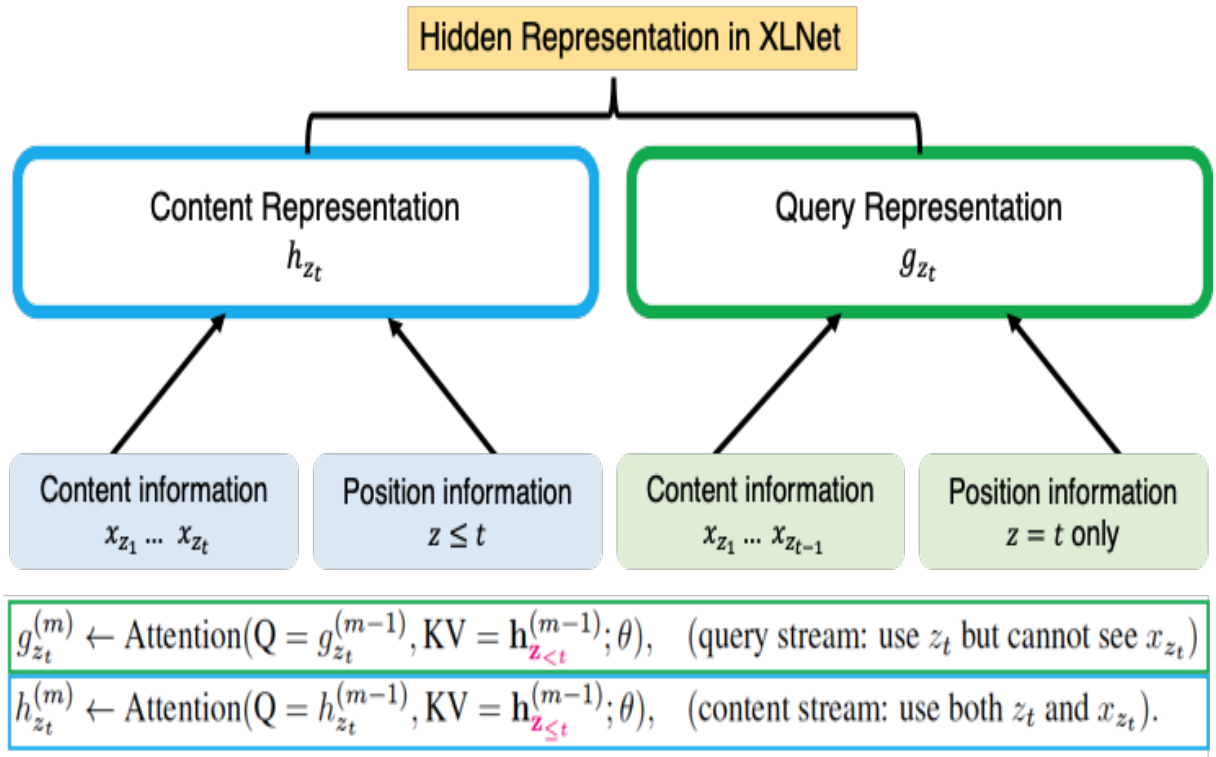

FIGURE 3.12 – XLNET Architecture.

3.4.4 BiLSTM (Bidirectional Long Short-Term Memory)

BiLSTM[3], or Bidirectional Long Short-Term Memory, is an extension of the standard LSTM architecture. While a regular LSTM processes sequences in a forward direction (from past to future), a BiLSTM processes the sequence in both forward and backward directions, effectively capturing context from both the past and the future in a sentence. This bidirectional nature makes it particularly well-suited for tasks like sentiment analysis, emotion detection, and panic classification, where both prior and subsequent words affect the meaning of a word or phrase.

By learning from both directions, BiLSTM improves the model's understanding of the structure and semantics of the input text, which is crucial in detecting subtle emotional cues in tweets.

Architecture

The BiLSTM architecture is similar to LSTM but with two parallel LSTM layers :

1. Input Layer : Tokenized and padded sequences are fed as input. Each sequence represents a tweet or sentence.
2. Embedding Layer : Converts word indices into dense vector representations. Captures semantic relationships between words.
3. Consists of two LSTM layers : Forward LSTM processes the sequence from left to right. Backward LSTM processes it from right to left. Outputs from both directions are concatenated or averaged to form a comprehensive context-aware representation.
4. Dropout Layer (Optional) : Prevents overfitting by randomly deactivating a fraction of neurons during training.
5. Dense Layer(s) : Fully connected layers used to interpret the features from the BiLSTM output.
6. Output Layer : Applies a Softmax function to output the probability of each panic class.

Layer (type)	Output Shape	Param #
embedding_22 (Embedding)	(None, 100, 128)	8,057,856
conv1d_22 (Conv1D)	(None, 98, 64)	24,640
max_pooling1d_22 (MaxPooling1D)	(None, 49, 64)	0
bidirectional_22 (Bidirectional)	(None, 128)	66,048
dropout_22 (Dropout)	(None, 128)	0
dense_44 (Dense)	(None, 32)	4,128
dense_45 (Dense)	(None, 7)	231

FIGURE 3.13 – BILSTM Architecture.

3.5 Web Application Development

To make the panic detection system accessible and user-friendly, a web application was developed as the front-end interface for interacting with the machine learning model. This application is designed to enable real-time analysis of Twitter data and present results in

a meaningful way for both patients and administrators. The development process integrated several modern web technologies and machine learning tools to ensure performance, interactivity, and scalability.

3.5.1 Technologies Used

Backend and Frontend Frameworks :

— React.js : Used for building reusable UI components, handling patient registration/login, and rendering dynamic results from the ML model. React's component-based architecture allowed modular and maintainable development.

— Angular : Utilized in the administrative dashboard to provide robust state management, route protection, and interaction with backend services for viewing patient activity and prediction history.

API Integration :

— Twitter Scraper API : A scraping system was implemented to fetch tweets from user-provided handles or keywords, while respecting Twitter's data access policies. This API acted as a bridge between real-world social media activity and the trained model.

— Trained ML Model Integration : A pre-trained classification model (e.g., LSTM, BiLSTM, or XLNet) was deployed on the server and exposed via REST API endpoints. The model receives cleaned tweets and returns predicted classes (Normal, Stress, Depression, Suicidal, Anxiety).

— User Authentication and Roles :

Patient Account Creation : Patients can create personal accounts and log in to view their analyzed tweets and mental state predictions over time. Each account securely stores personal tweet history and diagnosis results.

Administrator Connection : An admin interface allows authorized medical or supervisory personnel to monitor patient mental health trends, flag high-risk cases (e.g., suicidal), and export reports if needed.

3.5.2 System Workflow

— Patient Account Management :

Patients register and log in securely. Upon login, they can initiate analysis by providing their Twitter handle or search terms.

— Tweet Scraping : The Twitter Scraper API fetches recent tweets for the patient. Tweets are preprocessed (tokenized, cleaned, abbreviation-expanded) before being sent to the model.

— Prediction and Visualization : The cleaned tweets are passed to the backend model. Predictions (e.g., Stress, Normal) are returned and visualized using dynamic charts or graphs (React/Angular). Patients receive real-time feedback and can track their mental state over time.

— Administrator Access :Admins can log in separately to view a dashboard of all connected patients.Admins can access summary statistics, individual history, and receive alerts for critical cases.

3.5.3 Functional and Non-Functional Requirements

Functional Requirements

The functional requirements describe the core functionalities that the web application must perform :

1. Administrator Login Admins must be able to log in to a secure dashboard to monitor patient data and system performance.
2. Twitter Scraping Functionality The system must fetch tweets based on the patient's handle or custom keywords using the Twitter Scraper API.
3. Tweet Preprocessing The system should clean and preprocess the tweets (e.g., removing links, punctuation, expanding abbreviations).
4. Prediction Using ML Model Preprocessed tweets must be sent to the trained ML model to detect the mental state (Normal, Stress, Depression, Suicidal, or Anxiety).
5. Results Visualization The system should display prediction results in an intuitive format (e.g., bar charts, tables) on the user interface.
6. History Tracking Each admin should be able to view their pateint past mental health predictions.
7. Admin Monitoring Tools Admins must be able to access a list of patients, view their history, and receive alerts for high-risk predictions .

Non-Functional Requirements

The non-functional requirements define the quality attributes and constraints of the system :

1. Security :The system must implement authentication and authorization to protect user data.Sensitive data (e.g., patient identity, prediction results) should be encrypted or secured.
2. Scalability :The system should be scalable to handle multiple users and prediction requests simultaneously.
3. Performance :The application must provide quick response times for scraping, processing, and classification tasks.
4. Usability :The user interface must be simple, intuitive, and responsive for both patients and administrators.
5. Reliability :The system should provide consistent and accurate predictions.It must handle failures gracefully (e.g., API downtime or no tweets found).
6. Maintainability :The codebase should follow modular and clean architecture to allow future updates and improvements.
7. Compatibility :The web application should work across major browsers (Chrome, Firefox, Edge) and different screen sizes.

3.6 System Design

This section details the architectural and functional design of the "MindInsight" system for recognizing mental health states from tweets. The design is based on the following UML (Unified Modeling Language) diagrams : the use case diagram, the class diagram, and several sequence diagrams illustrating key interactions.

3.6.1 Use Case Diagram

The use case diagram (see Figure 3.14) illustrates the main functionalities of the system from the perspective of the primary actor, the **Doctor**.

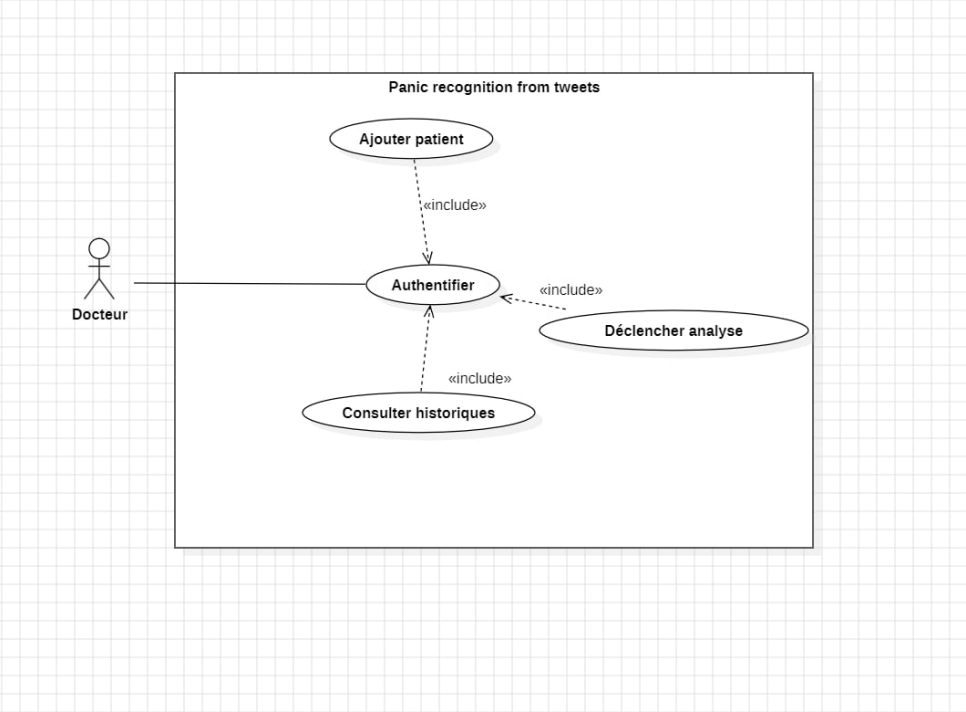

FIGURE 3.14 – Use Case Diagram of the MindInsight system.

- **Actor :**
  - **Doctor :** The main user of the system, authorized to manage patients and their analyses.
- **Main Use Cases :**
  - **Authenticate :** Allows the doctor to securely log into the system. This use case is included (via the `«include»` relationship) by all other use cases, indicating that authentication is required before accessing other functionalities.
  - **Add Patient :** Allows the doctor to register a new patient in the system.
  - **Trigger Analysis :** Allows the doctor to initiate a tweet analysis for a selected patient to determine their mental health state.
  - **Consult Histories :** Allows the doctor to view the history of analyses and detected states for their patients.

This diagram provides an overview of the services offered by the "Panic recognition from tweets" system to a Doctor-type user.

3.6.2 Class Diagram

The class diagram (see Figure 3.15) defines the static structure of the system, presenting the main classes, their attributes, their methods, and the relationships between them.

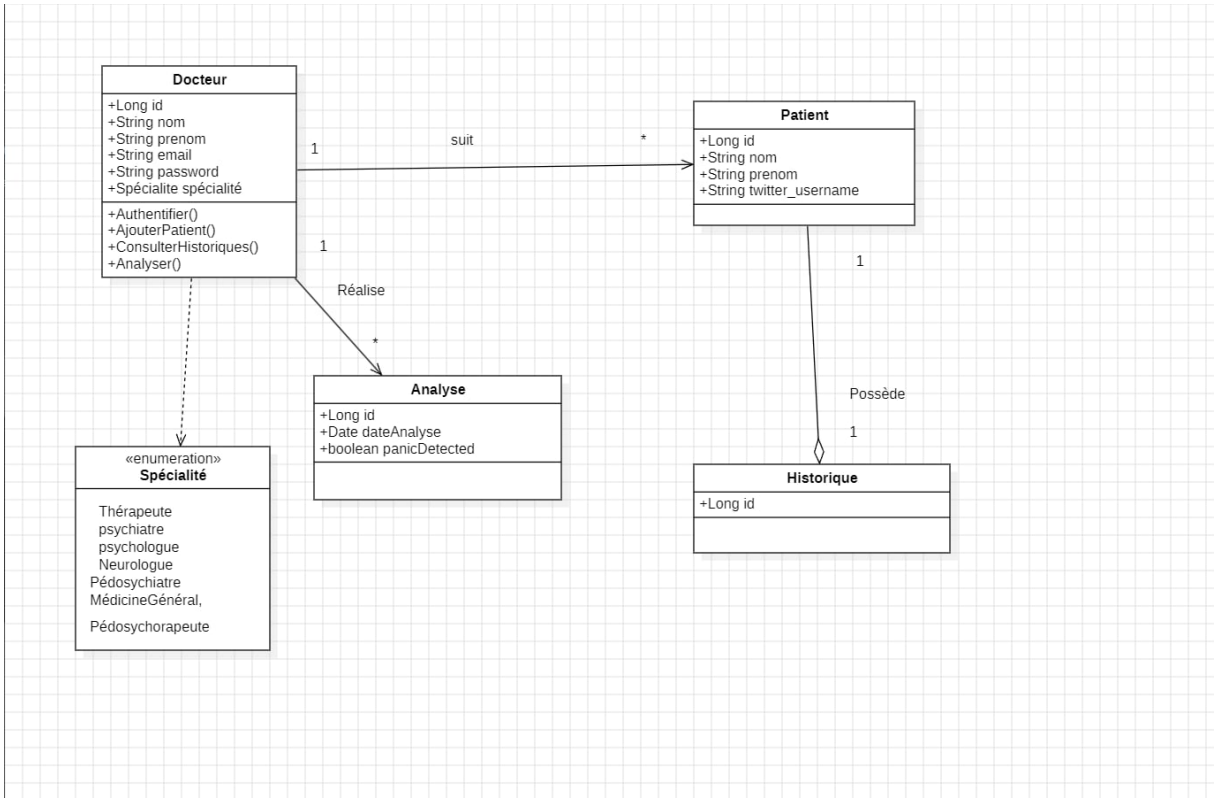

FIGURE 3.15 – Class Diagram of the MindInsight system.

- **Main Classes :**
  - **Doctor :** Represents a healthcare professional.
    - Attributes : `id` (Long), `name` (String), `firstName` (String), `email` (String), `password` (String), `specialty` (Specialty - enumeration).
    - Methods : `Authenticate()`, `AddPatient()`, `ConsultHistories()`, `Analyze()`.
  - **Patient :** Represents a patient monitored by a doctor.
    - Attributes : `id` (Long), `name` (String), `firstName` (String), `twitter_username` (String).
  - **Analysis :** Represents the result of a tweet analysis for a patient.
    - Attributes : `id` (Long), `analysisDate` (Date), `panicDetected` (boolean).
  - **History :** Represents the history of a patient's analyses.
    - Attributes : `id` (Long).
  - **Specialty (enumeration) :** Defines the different possible specialties for a doctor (Therapist, psychiatrist, etc.).
- **Main Relationships :**
  - A `Doctor` `follows` (follows/manages) one or more (`*`) `Patient`(s).
  - A `Doctor` `Performs` (performs) one or more (`*`) `Analysis`.
  - A `Patient` `Has` (has) one (1) `History`. (Note : The relationship between History and Analysis is not explicitly detailed with multiplicity in this diagram, but a

history would logically contain multiple analyses).
— An `Analysis` is performed by a `Doctor`.

This diagram is fundamental for understanding the data structure and the entities manipulated by the system.

3.6.3 Sequence Diagrams

Sequence diagrams describe the dynamic interactions between system objects to achieve specific use cases.

Authentication Sequence (`sd Authenticate`) (See Figure 3.16) This diagram illustrates the doctor's login process :

1. The `Doctor` requests the login form from the `user interface`.
2. The `user interface` displays the form.
3. The `Doctor` enters their credentials (email, password).
4. The `user interface` sends a login request (`requestLogin`) to the `Controller` with the credentials.
5. The `Controller` validates the credentials.
   — **If valid :** The `Controller` returns a `login success` to the `user interface`, which displays the main menu and notifies the `Doctor` of the `successful connection`.
   — **If invalid :** The `Controller` returns a `login failed` to the `user interface`, which displays an error message and notifies the `Doctor` to `display error`.

Add Patient Sequence (`sd Add patient`) (See Figure 3.17) This diagram details how a doctor adds a new patient :

1. The `Doctor` initiates the `add patient` action via the `user interface`.
2. The `user interface` displays the addition form.
3. The `Doctor` enters the patient's details.
4. The `user interface` transmits these details to the `controller` via `add patient(patient details)`.
5. The `controller` instantiates a `patient` object via `create(patient details)`.
6. The `controller` saves the patient (`save patient`), presumably to the database.
7. The `controller` returns an `add patient success` to the `user interface`.
8. The `user interface` displays a success message.
9. The `Doctor` is notified that the `patient added`.

Analysis Sequence (`sd Analyse`) (See Figure 3.18) This diagram shows the workflow of a tweet analysis for a patient :

1. The `Doctor` initiates the `Analyze` action from the `user interface`.
2. The `user interface` allows for patient selection.
3. The `Doctor` selects a patient.

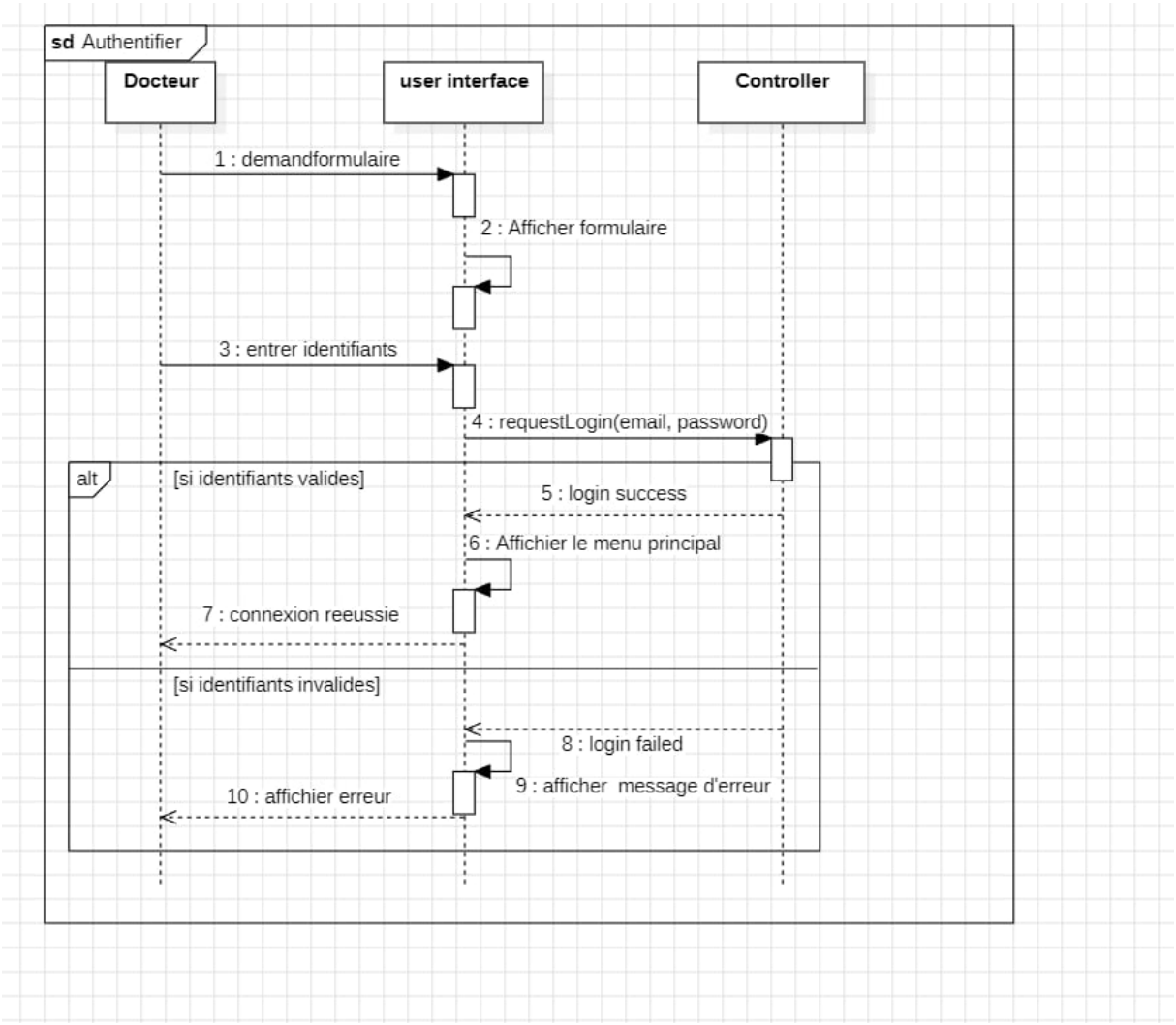

FIGURE 3.16 – Sequence Diagram : Doctor Authentication.

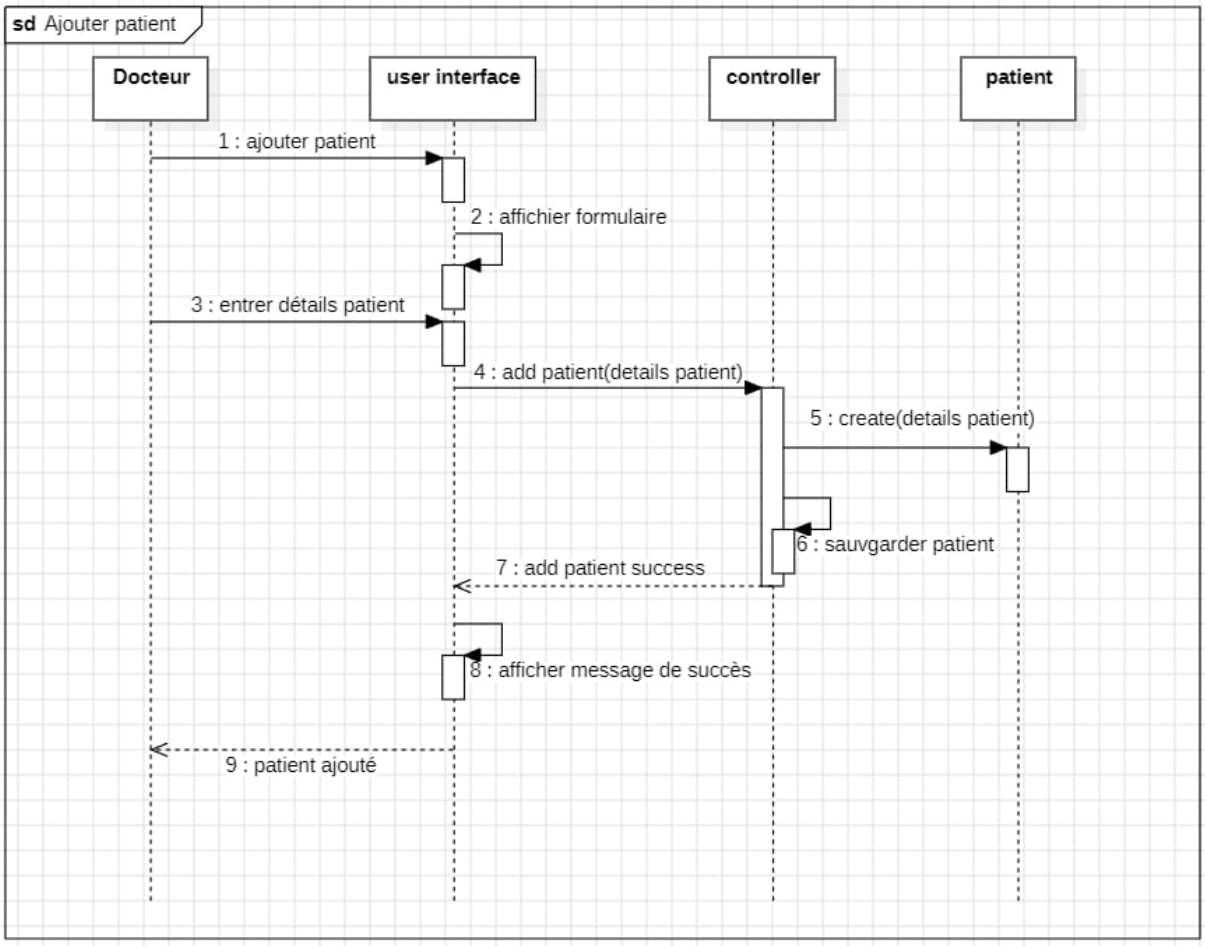

FIGURE 3.17 – Sequence Diagram : Adding a Patient.

4. The `user interface` sends an analysis request (`requestAnalysis`) to the `controller` with the patient's ID.
5. The `controller` retrieves the patient's tweets (`retrieve tweets`).
6. The `controller` performs the analysis (`perform analysis`) on these tweets (this is where the Machine Learning model intervenes).
7. The `controller` creates an `analysis` object with the result (`create(analysis result)`).
8. The `controller` saves the analysis (`save analysis`).
9. The `controller` returns the analysis result (`analysis result`) to the `user interface`.
10. The `user interface` displays the result.
11. The `Doctor` views the `displayed results`.

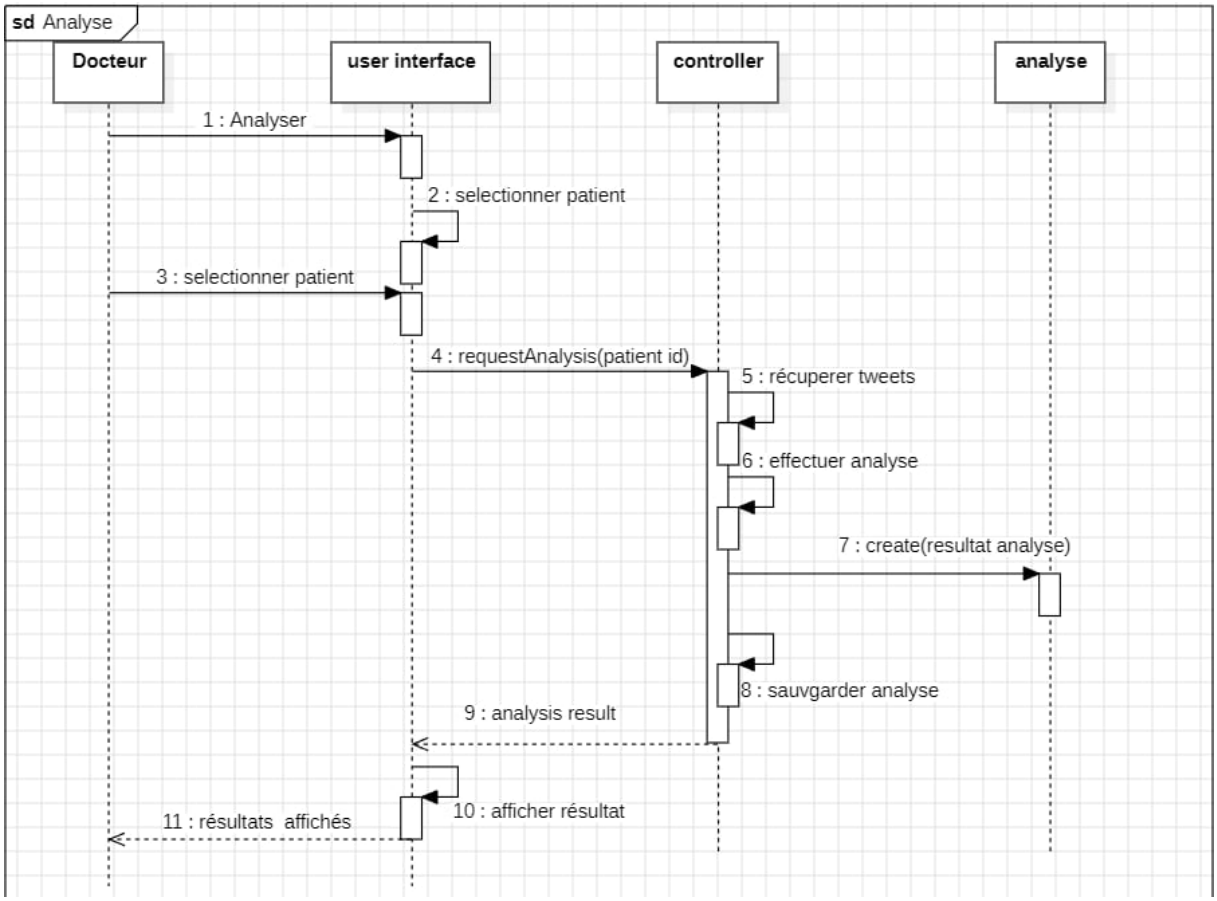

FIGURE 3.18 – Sequence Diagram : Patient Tweet Analysis.

3.6.4 Implicit Architectural Choices

The diagrams suggest a layered architecture, approaching a Model-View-Controller (MVC) pattern or similar :

— **User Interface (View) :** Manages information presentation and interactions with the doctor.

— **Controller :** Receives requests from the user interface, orchestrates business logic (calling analysis services, data management), and updates the view.
— **Model :** Represented by the data classes (`Doctor`, `Patient`, `Analysis`, `History`) and associated business logic (not explicitly detailed in the sequence diagrams, but including tweet retrieval and execution of the analysis model).

Data persistence is managed by the controller or a dedicated data access layer, interacting with a database (MongoDB as mentioned in the project description).

3.7 Conclusion

This chapter has detailed the proposed approaches for developing "MindInsight," an intelligent system designed for mental health recognition from tweets. We began by outlining the primary objective : to create a system capable of classifying tweets into distinct mental health-related categories (Normal, Stress, Depression, Suicidal, Anxiety) to aid mental health professionals. A significant portion of the chapter was dedicated to the data preprocessing pipeline, a critical phase for handling the noisy and informal nature of Twitter data. This involved several steps, including noise removal, normalization, tokenization, stop-word elimination, stemming, emoji resolution, and abbreviation expansion, all aimed at preparing the textual data for effective model training. Following preprocessing, we explored various predictive models. While classical machine learning models and other deep learning architectures like LSTM, BiLSTM, and XLNet were considered, the core analytical engine relies on a fine-tuned BERT-base-uncased model, chosen for its robust contextual understanding capabilities. The development of a user-friendly web application was then described, outlining the technologies used (React.js, Angular, FastAPI, MongoDB) and the system workflow. This application serves as the primary interface for doctors to manage patients, initiate analyses, and visualize the results, thereby bridging the gap between complex AI analysis and practical clinical use. Finally, the chapter presented the system's design through UML diagrams, including use cases, class structures, and key interaction sequences for authentication, patient addition, and tweet analysis. These design artifacts illustrate the system's architecture and the interactions between its core components, suggesting a layered approach akin to the MVC pattern.

Chapitre 4

Experiments and discussion

4.1 Introduction

The Experiments and Discussion chapter is a pivotal section of this research, where we present the outcomes of the conducted experiments and critically analyze the results. This chapter aims to assess the performance and effectiveness of the model or system under investigation, based on the methodologies outlined in the previous chapters. Specifically, it highlights the evaluation metrics, the challenges encountered, and the insights gained from analyzing the experimental results.

The experiments were designed to test the core hypotheses of the research, examining the system's ability to detect panic in tweets accurately. By leveraging a variety of classification models and evaluation techniques, the experiments provide valuable insight into the strengths and limitations of the developed approach. Additionally, this section discusses the implications of the findings in relation to existing methods in the field, shedding light on the potential for improvement and future work.

4.2 Realization

4.2.1 Experimental Setup

Data Collection

The dataset used for this experiment consists of tweets that have been labeled with the presence or absence of panic. These labeled tweets were collected from a variety of public sources that include real-time updates, emergency situations, and general conversations where panic-related content may appear. The tweets were pre-processed to ensure that they are suitable for use in training and testing the model.

Tweet Content : Each tweet contains a short textual message, which could include signs of panic or anxiety expressed by the user.

Labels : The dataset is labeled as binary (1 for panic-related tweets, 0 for non-panic tweets). Labels were manually annotated or automatically generated based on predefined keywords related to panic or distress.

Data Split : The dataset was split into training (0.8) and testing (0.2) sets to ensure that the model generalizes well to unseen data.

Preprocessing

The preprocessing of tweet data is crucial in preparing the text for model training. The following steps were applied to the raw tweet data :

Text Cleaning : URLs, mentions (@user), hashtags, and other non-informative characters were removed.

Tokenization : The text was split into individual words or tokens using tokenization techniques. This allows the model to analyze each word independently.

Stopword Removal : Common words that do not contribute significant meaning (such as "and", "the", etc.) were removed.

Lemmatization : Words were reduced to their base or root form using lemmatization (e.g., "running" -> "run").

Vectorization : Tweets were transformed into numerical representations using techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec or GloVe). These techniques allow the model to process text data as numerical input.

4.2.2 result

The project aimed to classify tweets into five distinct mental health-related categories : 'Normal,' 'Stressed,' 'anxiety,' 'depression,' and 'potential suicide post.' Analysis of the dataset, as visualized in the figure3.2 chart, reveals a class imbalance. 'Normal' (over 10,000 instances) and 'anxiety' (around 9,500 instances) are the most prevalent classes, followed by 'Stressed' (around 7,500). In contrast, 'depression' (around 1,500 instances) and 'potential suicide post' (around 1,400 instances) represent considerably smaller minority classes. This imbalance is a critical factor, as models can become biased towards majority classes, potentially impacting performance on less frequent but crucial categories like 'potential suicide post.' To prepare the textual data for modeling, a comprehensive preprocessing pipeline was likely employed. This would typically involve tokenization (WordPiece for BERT/XLNet, or word-level for LSTMs), crucial for breaking down text into manageable units. Stemming or lemmatization might have been used, particularly for LSTM/BLSTM, to reduce words to their root forms, helping to consolidate vocabulary and improve generalization. Handling emojis (e.g., converting to textual representations or removing them) and expanding common abbreviations would further standardize the input, reducing noise and ambiguity. These preprocessing steps are vital as they directly influence the quality of features the models learn from ; cleaner, more consistent input generally leads to better performance by allowing models to focus on meaningful linguistic patterns rather than superficial variations.

The models were trained over a specific number of epochs using a defined train-test split (an 80/20 split) to evaluate their generalization capabilities on unseen data. Among the models evaluated – BERT, XLNet, BLSTM, and LSTM – BERT demonstrably achieved the best results on both the training data (inferred by its strong test performance, assuming no significant overfitting) and, crucially, on the test set. BERT yielded an impressive overall accuracy of 0.91 and a weighted F1-score of 0.91. The provided confusion matrix for BERT further substantiates its superior performance : it shows strong diagonal dominance, correctly classifying a high number of instances for 'Normal' (1961), 'Stressed' (1370), and 'anxiety' (1780). While some confusion exists, particularly for 'depression' which is sometimes misclassified as 'anxiety' (27 instances) or 'Normal' (25 instances), BERT still correctly identifies 222 'depression' posts and, importantly, 244 'potential suicide posts' with minimal misclassifications into other categories for this critical class. This indicates BERT's robust ability to discern nuanced differences in language indicative of these mental states.

In comparison, XLNet achieved a commendable accuracy of 0.88, while both BLSTM and

LSTM models reached an accuracy of 0.84. While these are solid performances, BERT's architectural advantages, such as its transformer-based attention mechanism and extensive pre-training on vast corpora, likely contribute to its enhanced understanding of context and subtle linguistic cues. The meticulous data preprocessing, especially effective tokenization, provides BERT with a rich and well-structured input, allowing its powerful architecture to excel, making it the most effective model for this challenging and important classification task.

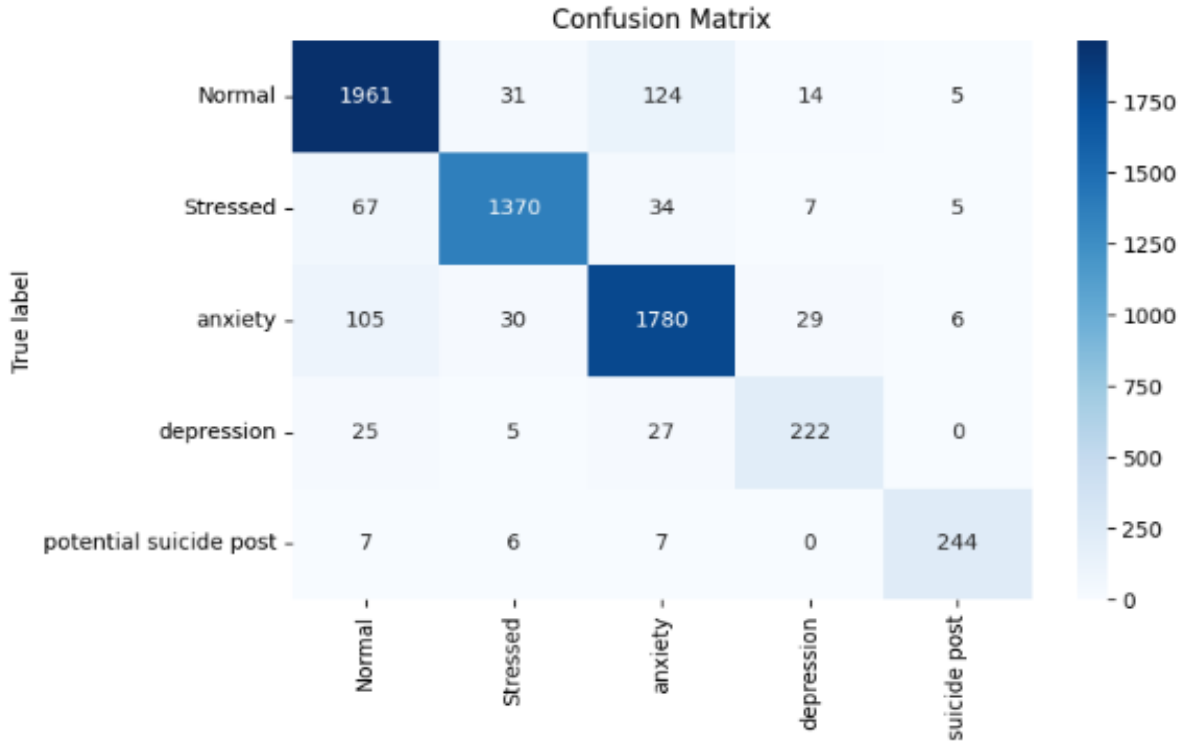

FIGURE 4.1 – Confusion Matrix Bert.

Classification Report:

	precision	recall	f1-score	support
Normal	0.91	0.92	0.91	2135
Stressed	0.95	0.92	0.94	1483
anxiety	0.90	0.91	0.91	1950
depression	0.82	0.80	0.81	279
potential suicide post	0.94	0.92	0.93	264
accuracy			0.91	6111
macro avg	0.90	0.90	0.90	6111
weighted avg	0.91	0.91	0.91	6111

FIGURE 4.2 – Bert result.

XLNet Classification Report (Best Result: 0.88):

	precision	recall	f1-score
Normal	0.88	0.89	0.88
Stressed	0.92	0.89	0.90
anxiety	0.87	0.88	0.87
depression	0.79	0.77	0.78
potential suicide post	0.91	0.89	0.90
accuracy			0.88
macro avg	0.87	0.86	0.87
weighted avg	0.88	0.88	0.88

FIGURE 4.3 – Xlnet result.

BLSTM Classification Report (Best Result: 0.84):

	precision	recall	f1-score
Normal	0.84	0.85	0.84
Stressed	0.88	0.85	0.86
anxiety	0.83	0.84	0.83
depression	0.75	0.73	0.74
potential suicide post	0.87	0.85	0.86
accuracy			0.84
macro avg	0.83	0.82	0.83
weighted avg	0.84	0.84	0.84

FIGURE 4.4 – Blstm result.

LSTM Classification Report (Best Result: 0.84):

	precision	recall	f1-score
Normal	0.84	0.85	0.84
Stressed	0.88	0.85	0.86
anxiety	0.83	0.84	0.83
depression	0.75	0.73	0.74
potential suicide post	0.87	0.85	0.86
accuracy			0.84
macro avg	0.83	0.82	0.83
weighted avg	0.84	0.84	0.84

FIGURE 4.5 – Lstm result.

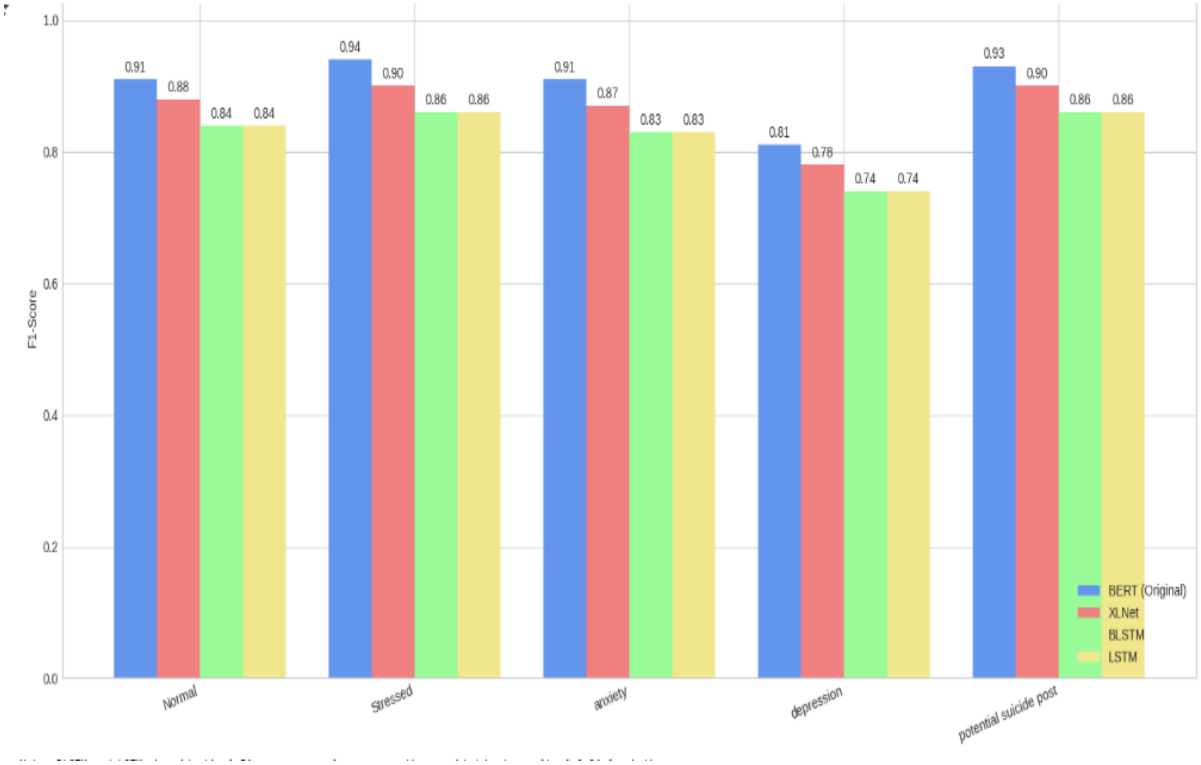

FIGURE 4.6 – Models result comparing.

4.3 application : Mindinsight

4.3.1 Landing Page

The main landing page of the MindInsight application. It introduces the service's purpose (analyzing patient psychological states via social media), includes calls to action ('Commencer maintenant', 'En savoir plus'), and provides navigation links for Contact, Login ('Connexion'), and Sign Up ('Inscription').

FIGURE 4.7 – Landing Page

4.3.2 Dashboard - Analyses en cours

: This image illustrates the 'Analyses en cours' section when no analyses are currently being processed or have been recently completed for the logged-in doctor. It displays the message 'Aucune analyse en cours' (No ongoing analysis)

FIGURE 4.8 – Analyses in Process .

4.3.3 Analyses en cours

Screenshot of the 'Analyses en cours' (Ongoing Analyses) section of the doctor's dashboard. It lists analyses initiated for patients, displaying the patient's name, Twitter

handle, analysis date, status (e.g., 'Terminé' - Finished), and provides an option ('Voir l'analyse') to view the detailed report.

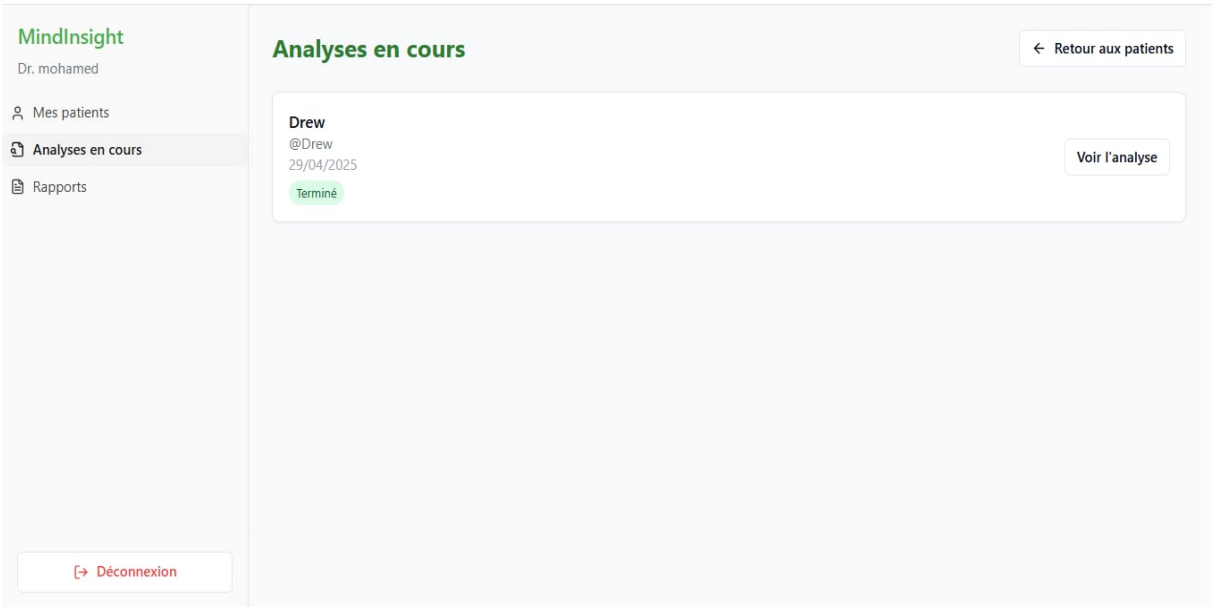

FIGURE 4.9 – Exemple of analyse finished.

4.3.4 Database collection

This image displays the structure of a document within the analyses collection in the MongoDB database. It details how analysis results are stored, including patient/doctor identifiers, overall summary statistics (Normal, Stressed, Anxiety, Depression, Potential Suicide Post counts), analysis date, and a detailed list of individual tweet predictions with their text, metadata, and predicted emotional state.


```
> db.analyses.find().pretty()
{
        "_id" : ObjectId("68112254552263e512c64d62"),
        "patient_id" : ObjectId("68112254552263e512c64d61"),
        "doctor_id" : ObjectId("681121f9552263e512c64d60"),
        "twitter_username" : "Drew",
        "analysis_date" : ISODate("2025-04-29T19:02:44.521Z"),
        "analysis_data" : {
                "username" : "Drew",
                "tweets_analyzed" : 10,
                "overall_summary" : {
                        "Normal" : 90,
                        "Stressed" : 0,
                        "Anxiety" : 10,
                        "Depression" : 0,
                        "Potential Suicide Post" : 0
                },
                "predictions" : [
                        {
                                "id" : NumberLong("1659776054599336000"),
                                "text" : "RT @PropertyBrother: Ready, set, reno! We give college football fans Tenny and
Lyle the ultimate party hosting space tonight on the season…",
                                "created_at" : "2023-05-20 04:20:23+00:00",
                                "likes" : 0,
                                "retweets" : 14,
                                "predicted_state" : "Normal",
                                "probabilities" : {
```


FIGURE 4.10 – saving patient Analyses in mongodb

4.3.5 Database data save

This screenshot shows example documents from the patients collection in the MongoDB database. It demonstrates how patient information is stored, including a unique identifier , the associated doctor's ID , the patient's name , their Twitter username , and the creation date .

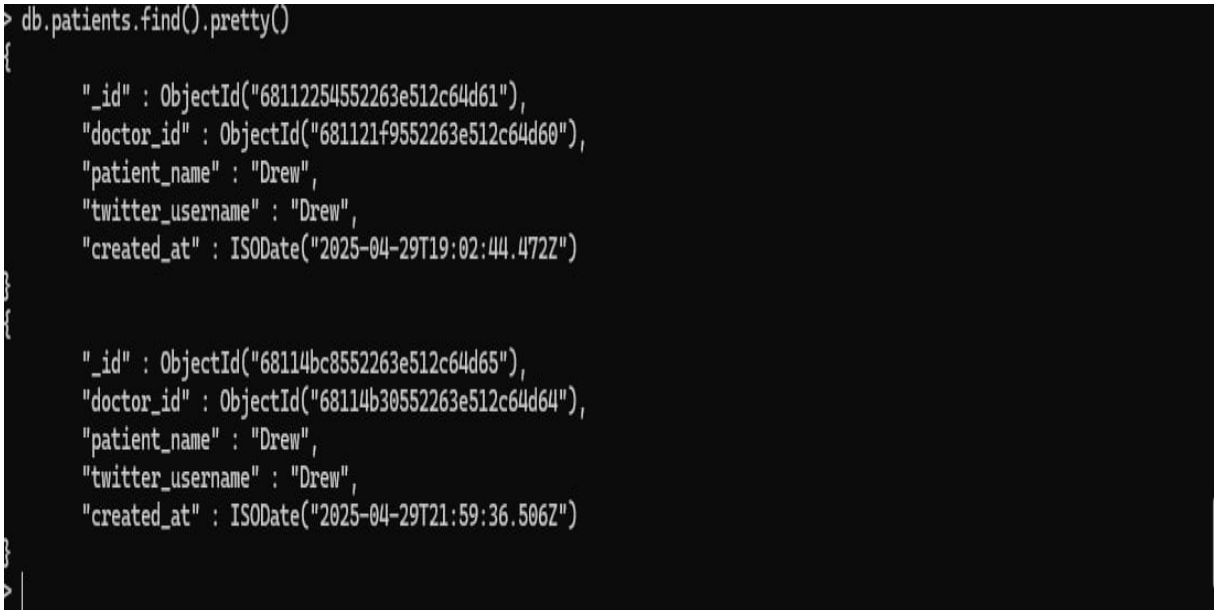

FIGURE 4.11 – creat collection in mongodb.

4.3.6 Analysis Detail

This image displays the 'Tweets Analysés' (Analyzed Tweets) section within a patient's analysis report. It lists individual tweets processed by the system, showing the tweet text, its predicted emotional state ('État prédit'), associated metadata (likes, retweets, date), and the detailed probability breakdown across different categories.

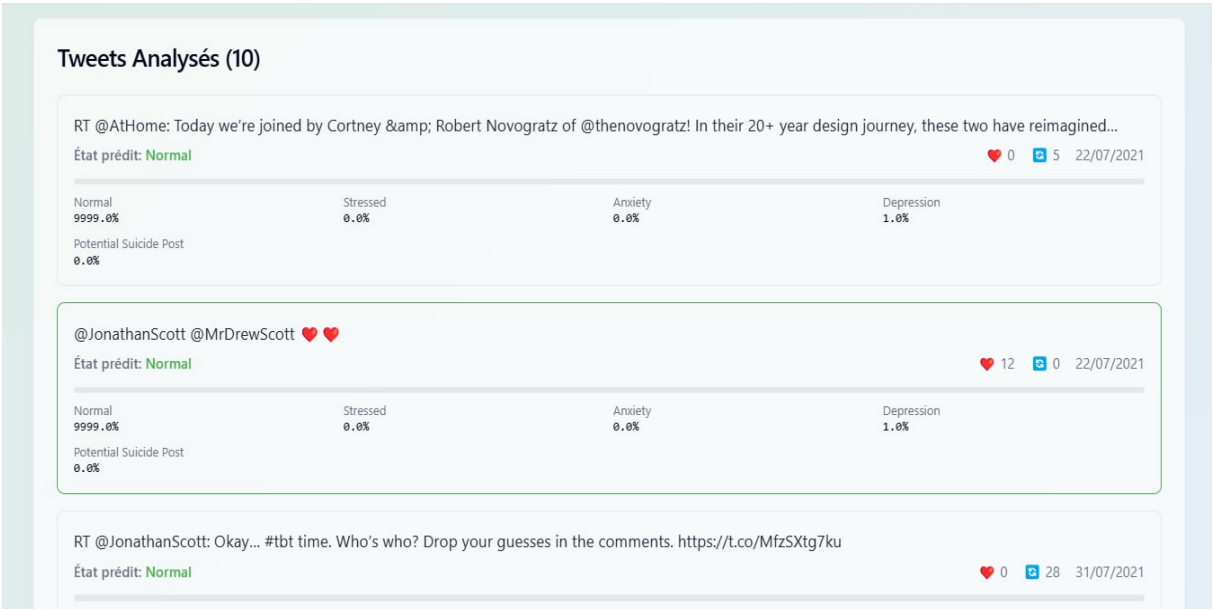

FIGURE 4.12 – Example of analysed tweets .

4.3.7 Dashboard - Rapports

Screenshot of the 'Rapports' (Reports) section of the doctor's dashboard. This view is shown when there are no generated reports available for the logged-in doctor, indicated by the message 'Aucun rapport disponible' (No report available).

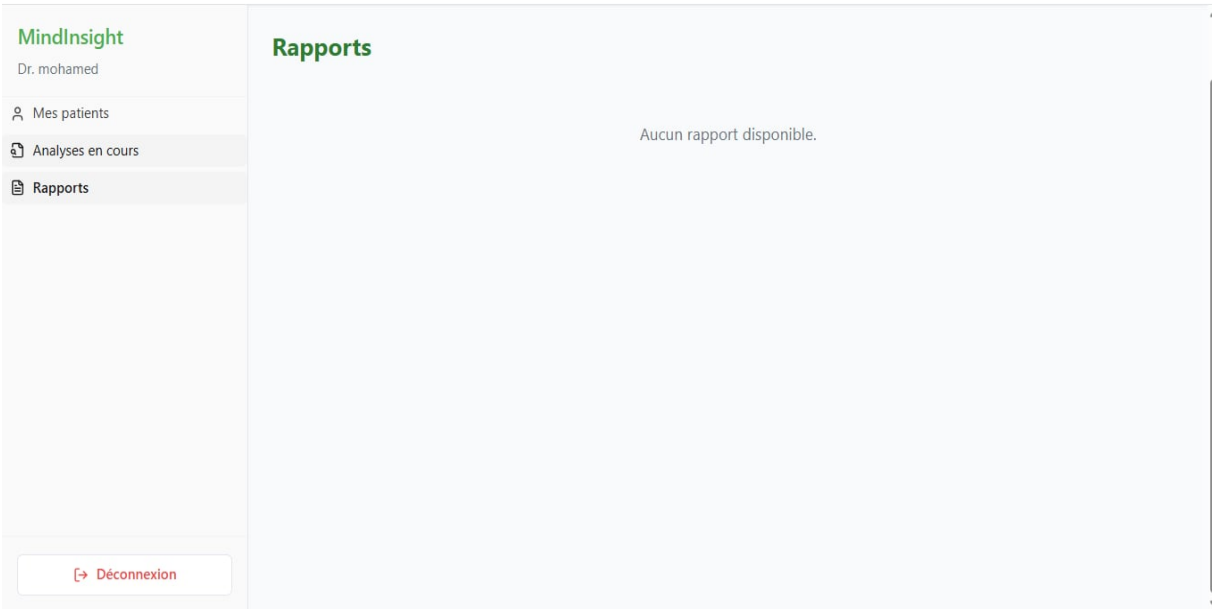

FIGURE 4.13 – Example of rapport for doctor .

4.3.8 Dashboard - Mes patients - Add Analysis

This image shows the 'Mes patients' (My Patients) section, specifically the interface for initiating a new analysis. The doctor can enter the patient's name ('Nom du patient')

and their Twitter profile ('Profil Twitter') and then click 'Analyser' (Analyze) to start the process. The lower part indicates that no patients have been analyzed yet for this doctor.

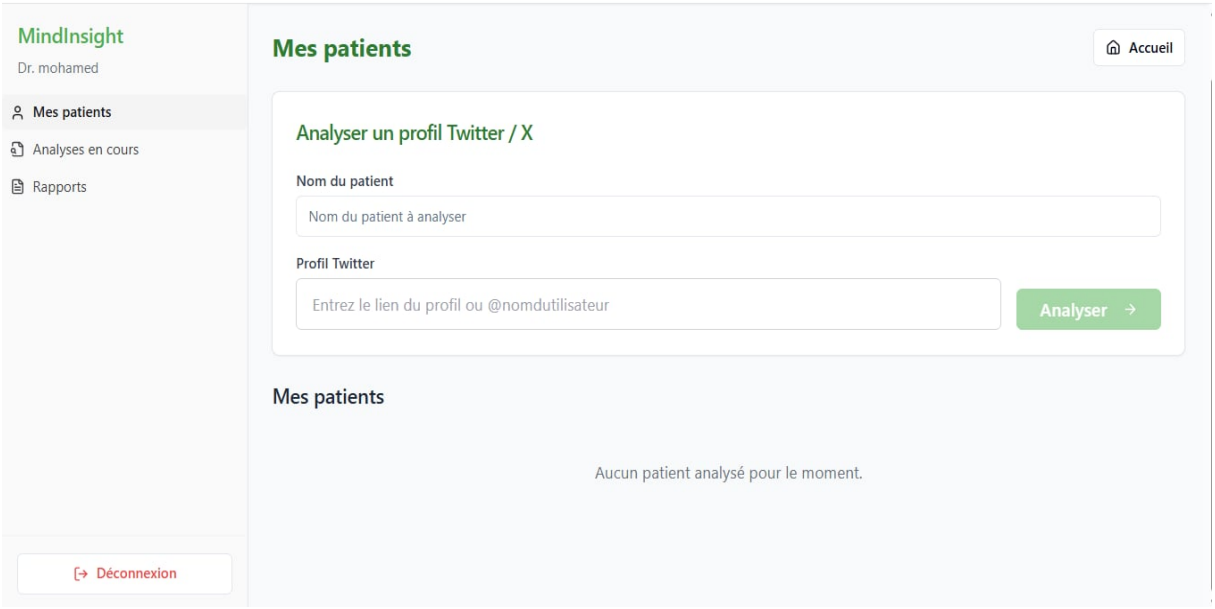

FIGURE 4.14 – detailed rapport for doctor.

4.3.9 Analysis Detail - Overview

This screenshot presents the main analysis report view for a patient ('Analyse de Drew'). It features key visualizations : a pie chart showing the 'Distribution des États Émotionnels' (Distribution of Emotional States) and a line graph illustrating the 'Évolution du Sentiment' (Sentiment Evolution) over time.

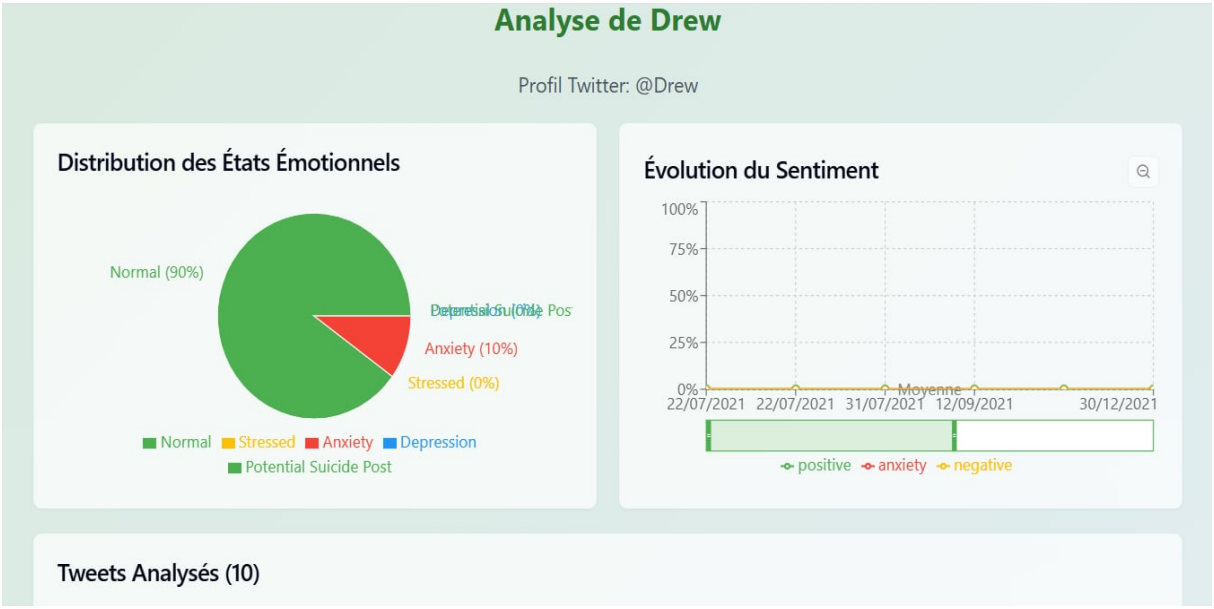

FIGURE 4.15 – result of various model.

4.3.10 Login - Password Error

The application's login ('Connexion') interface. This screenshot specifically highlights the password input field validation, displaying an error message ('Le mot de passe doit contenir au moins 6 caractères' - Password must contain at least 6 characters) when the entered password does not meet the length requirement.

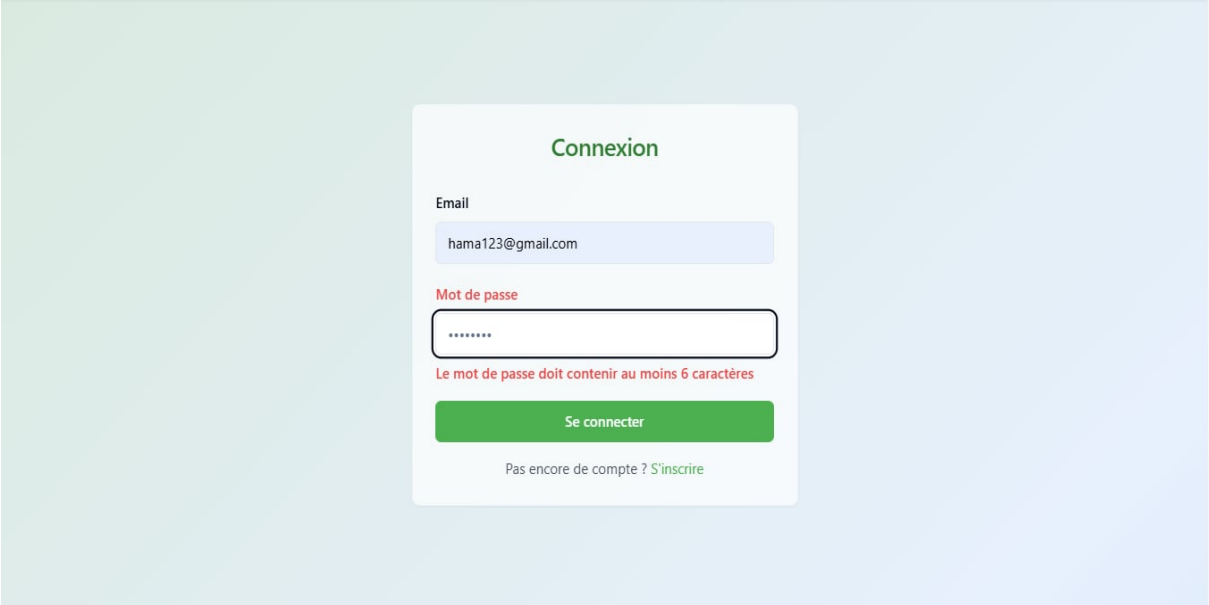

FIGURE 4.16 – Login test.

4.3.11 Backend Application Initialization

The backend application, developed using the FastAPI framework and served by Uvicorn, undergoes a sequential initialization process upon execution, as detailed in the server logs (see Appendix/Figure X for full log). The Uvicorn server starts, binding to http ://127.0.0.1 :8000 and enabling auto-reload for development efficiency. Critical configurations are established, including the connection parameters for the MongoDB database . Essential services are then initialized : the Twitter client connects, and the core sentiment analysis component, the BERT model and its associated tokenizer, are loaded into memory, utilizing the CPU for processing. Although a warning regarding newly initialized classification weights is noted, the model loading completes successfully.

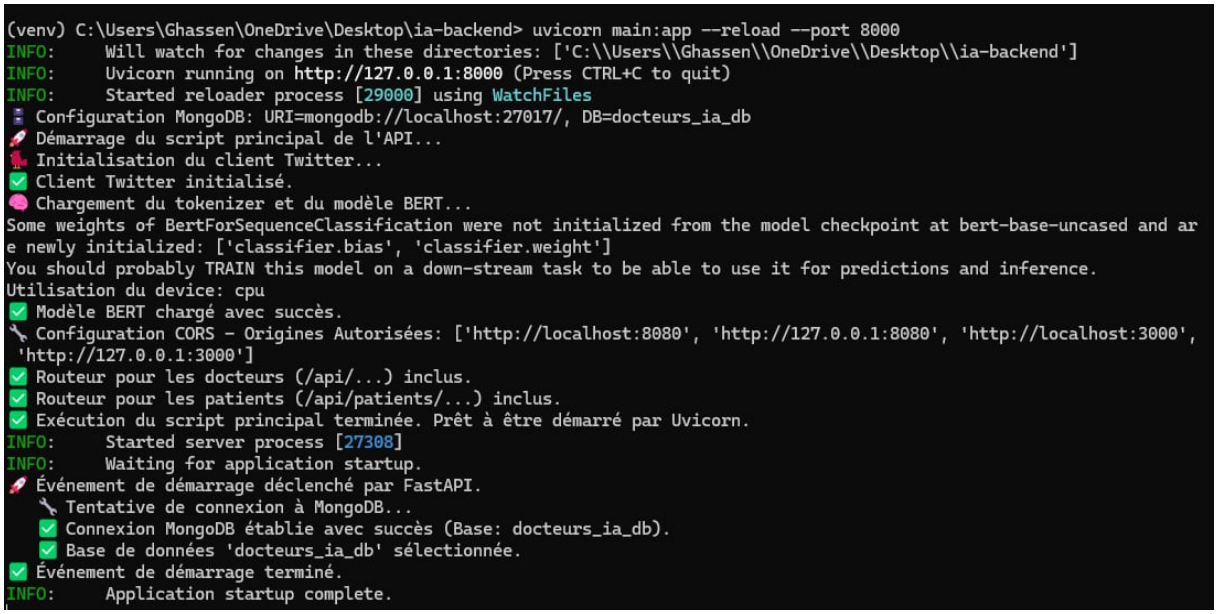

FIGURE 4.17 – Backend Application Initialization.

4.4 Conclusion

The system successfully integrates a user-friendly frontend with a robust FastAPI backend and MongoDB database, effectively managing user authentication, patient data, and analysis results. The core analytical engine, leveraging a fine-tuned bert-base-uncased model, demonstrated strong performance with an overall accuracy of 0.91 and high F1-scores for key categories like 'Stressed' and 'Potential Suicide Post'. While evaluation results are promising and validate the technical feasibility, MindInsight currently serves as a functional prototype. Looking forward, several perspectives emerge for enhancing the platform's utility and impact. Firstly, model refinement is crucial ; further fine-tuning, potentially utilizing larger datasets or exploring alternative architectures like RoBERTa, could improve accuracy, particularly for nuanced categories like 'Depression', and reduce the observed confusion between related states. Secondly, the most critical next step is clinical validation. Collaborating with mental health professionals in controlled studies (following strict ethical guidelines and obtaining informed consent) is essential to assess the tool's real-world effectiveness, usability, and integration into clinical workflows. Thirdly, feature enhancement could significantly increase value ; implementing a robust alerting system for high-risk predictions and developing more sophisticated visualizations for tracking sentiment evolution over time would provide more actionable insights. Furthermore, incorporating explainability techniques (XAI) could help clinicians understand why a particular prediction was made, fostering greater trust and interpretability. Finally, ongoing attention to ethical considerations, data privacy, and security protocols will be paramount as the system evolves, ensuring responsible use of sensitive patient information. Addressing these perspectives will be key to transitioning MindInsight from a promising prototype to a validated and impactful tool for mental healthcare support.

Chapitre 5

General Conclusion and future works

5.1 conclusion

This project "MindInsight" an intelligent system designed for the recognition of mental health states (Normal, Stressed, Anxiety, Depression, and Potential Suicide Posts) from Twitter data. By leveraging advanced Natural Language Processing techniques, particularly a fine-tuned BERT-base-uncased model, the system demonstrated strong analytical capabilities, achieving an overall accuracy of 0.91 on the test dataset, with notable F1-scores for critical categories like 'Stressed' and 'Potential Suicide Post'.
The project encompassed the entire development lifecycle, from data collection and rigorous preprocessing (including noise removal, normalization, emoji/abbreviation resolution) to model training and evaluation. A key outcome was the creation of a secure and user-friendly web platform, built with a modern frontend (React/Angular), a robust FastAPI backend, and a MongoDB database. This platform serves as a dashboard for mental health professionals, enabling them to manage patient profiles (with consent), initiate tweet analyses, and monitor classified emotional states derived from patients' recent activity, supported by clear data visualizations.
MindInsight stands as a functional prototype, demonstrating significant potential in bridging AI capabilities with practical clinical applications. It highlights the feasibility of using social media data for early detection of emotional distress, aspiring to contribute to increased mental health awareness and explore digitally-supported interventions. While promising, the project acknowledges the need for further refinement and validation to transition from a prototype to a clinically impactful tool, always prioritizing ethical considerations and data privacy.

5.2 Future Work

The development of **MindInsight** has laid a promising foundation, yet several avenues for future research and enhancement are envisioned to elevate its capabilities, accuracy, and real-world applicability. These key areas include :

5.2.1 Model Refinement and Enhancement

Improving the core analytical engine is paramount for increasing the system's reliability and a nuanced understanding of mental health expressions :

[leftmargin=*, itemsep=2pt]**Advanced Fine-Tuning and Pre-training :** Continued fine-tuning of the existing **BERT** model, potentially with domain-specific corpora, or exploring more advanced pre-trained models tailored for social media text or mental health language (e.g., further development on **MentalBERT** or **TwitBERT**). **Expanded and Diverse Datasets :** Incorporating larger,

more diverse, and longitudinally tracked datasets for training. This can help improve generalization across different demographics and linguistic styles, and potentially enable the model to capture evolving mental health states over time.

Exploration of Alternative Architectures : Investigating other state-of-the-art NLP architectures such as **RoBERTa**, **XLNet** variants, or transformer-based ensemble methods that might offer superior performance, particularly for nuanced categories like 'Depression' or in reducing confusion between closely related emotional states. **Improved Handling of Context and Nuance :** Developing more sophisticated methods for capturing sarcasm, irony, and subtle linguistic cues that are often prevalent in social media text and crucial for accurate mental health assessment.

5.2.2 Clinical Validation and Integration

To transition **MindInsight** from a research prototype to a clinically useful tool, rigorous validation and seamless integration are essential :

[leftmargin=*, itemsep=2pt]**Collaborative Clinical Studies :** Partnering with mental health professionals, psychologists, and psychiatrists to conduct controlled studies and clinical trials. This will be crucial for assessing the tool's real-world effectiveness, diagnostic concordance, usability within clinical settings, and its impact on patient outcomes. **Ethical Oversight and Informed Consent :** Ensuring all validation studies and future deployments adhere to strict ethical guidelines, institutional review board (IRB) approvals, and robust informed consent processes, particularly given the sensitive nature of mental health data. **Workflow Integration :** Designing and developing features that allow for seamless integration into existing electronic health record (EHR) systems or clinical decision support systems, minimizing disruption for healthcare providers.

5.2.3 Feature Enhancement and Platform Development

Expanding the functionality of the **MindInsight** platform can significantly increase its value to both clinicians and potentially patients :

[leftmargin=*, itemsep=2pt]**Robust Alerting System :** Implementing a sophisticated and configurable alerting mechanism to notify clinicians or designated personnel in real-time about high-risk predictions, such as severe distress or potential suicidal ideation, enabling timely intervention. **Advanced Data Visualizations and Trend Analysis :** Developing more interactive and insightful dashboards for tracking sentiment evolution, emotional patterns over time, and correlations with other factors, providing clinicians with a richer understanding of a patient's mental state trajectory. **Explainability (XAI) and Interpretability :** Incorporating eXplainable AI techniques (e.g., LIME, SHAP, attention visualization) to provide clinicians with insights into *why* the model made a particular prediction. This can foster greater trust, aid in clinical judgment, and help identify potential model biases. **Scalability and Performance Optimization :** Continuously monitoring and optimizing the system's backend infrastructure to ensure scalability, accommodating a growing number of users and the increasing volume of data processing, while maintaining rapid response times.

5.2.4 Ongoing Ethical, Privacy, and Bias Considerations

As the system evolves, a steadfast commitment to ethical principles and data protection is non-negotiable :

[leftmargin=*, itemsep=2pt]**Data Privacy and Security Enhancements :** Continuously reviewing and strengthening data anonymization, encryption, access control, and compliance with data protection regulations (e.g., GDPR, HIPAA if applicable) to safeguard sensitive patient information. **Bias Detection and Mitigation :** Proactively investigating and addressing potential biases in datasets, model algorithms, and predictions (e.g., demographic, cultural, or linguistic biases) to ensure fairness and equity in mental health assessment. **Responsible AI Governance :** Establishing clear governance frameworks for the development, deployment, and monitoring of **MindInsight**, ensuring its use aligns with responsible AI principles and societal values.

Addressing these multifaceted perspectives will be instrumental in transforming **MindInsight** from a promising research initiative into a validated, impactful, and ethically sound tool that can genuinely contribute to the early detection of mental health issues and support timely interventions in mental healthcare.

Bibliographie

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186).

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems, 30*.

[4] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet : Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems, 32*.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv :1907.11692*.

[6] Gkotsis, G., Oellrich, A., Velupillai, S., Liakata, M., Hubbard, T. J., Dobson, R. J., & Dutta, R. (2017). Characterisation of mental health conditions in social media using Informed Deep Learning. *Scientific Reports, 7*(1), 45141.

[7] De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, 7*(1), 128–137.

[8] Resnik, P., Armstrong, W., Claudino, L., Nguyen, T. (2015). The University of Maryland CLPsych 2015 shared task system. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology : From Linguistic Signal to Clinical Reality* (pp. 106–110).

[9] Benton, A., Coppersmith, G., & Dredze, M. (2017). Ethical quandaries of learning from public data for health research. *AI Magazine, 38*(3), 37–46.

[10] Chancellor, S., & De Choudhury, M. (2020). Methods in predictive techniques for mental health status on social media : a critical review. *NPJ Digital Medicine, 3*(1), 43.

[11] Cozma, I., Kousoulis, A. A., & Livanou, M. (2020). Social media and mental health : A literature review of the current research and future trends. *International Journal of Mental Health and Addiction, 18*(6), 1527–1546.

[12] Trotzek, M., Koitka, S., & Friedrich, C. M. (2018). Utilizing social media for a linguistic-based approach to identify and assess mental health conditions. In *Proceedings of the 5th Workshop on Computational Linguistics and Clinical Psychology : From Linguistic Signal to Clinical Reality* (pp. 97–106).

[13] Lykourentzou, I., Karka, A., Lymperopoulos, D., & Kermanidis, K. L. (2022). Early detection of mental health disorders from social media text using deep learning : A review. *Artificial Intelligence Review, 55*(8), 6155–6209.

[14] Ji, S., Yu, C. P., Fung, S. F., Pan, S., & Long, G. (2021). A survey on artificial intelligence for mental health. *ACM Computing Surveys (CSUR), 54*(10s), 1–39.

[15] Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence, 29*(3), 436–465.

[16] Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). A practical guide to sentiment analysis. In E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco (Eds.), *Socio-Affective Computing* (pp. 1–20). Springer, Cham.

[17] Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing : From unimodal analysis to multimodal fusion. *Information Fusion, 37*, 98–125.

[18] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R. (2000). *CRISP-DM 1.0 : Step-by-step data mining guide*. SPSS Inc. (Often cited as : The CRISP-DM consortium).

[19] Bender, E. M., Gebru, T., McMillan-Major, A., Shmitchell, S. (2021). On the Dangers of Stochastic Parrots : Can Language Models Be Too Big ? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610–623).

[20] Hovy, D., & Spruit, S. L. (2016). The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)* (pp. 591–598).