# Etiqa'a: An Android Mobile Application for Monitoring Teen's Private Messages on WhatsApp to Detect Harmful/Inappropriate Words in Arabic using Machine Learning

**Faiza Mohammed Usman Baran**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
faizaBaran@gmail.com

**Lama Saleh Abdullah Alzughaybi**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
lama.alzughaybi@gmail.com

**Manar Ahmed Saeed Bajafar**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
manar.bajafar@gmail.com (corresponding author)

**Maram Nasser Muslih Alsaedi**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
maramalsaedi@gmail.com

**Thraa Freed Hassan Serdar**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
tharaa.fs@gmail.com

**Olfat Meraj Nawab Mirza**
Computer Science Department, Umm Al-Qura University, Saudi Arabia
ommirza@uqu.edu.sa

*Received: 17 July 2023 | Revised: 21 August 2023 | Accepted: 30 August 2023*

 *DOI: https://doi.org/10.48084/etasr.6174*

**ABSTRACT**

**In today's world, social networks, such as WhatsApp, have become essential to daily life. An increasing number of Arab children use WhatsApp to communicate with others on a local and global scale, which has led to several negative consequences in their lives, including those associated with being bullied and harassed online. This study presents Etiqa'a, an application aiming to minimize risks and keep threats against minors from becoming a reality. Etiqa'a scans received WhatsApp messages which are then analyzed, and classified using a Logistic Regression (LR) machine learning model. The test results showed an accuracy of 81% in classifying messages as appropriate or inappropriate based on the text of the message. In the case of the latter, the application sends a detailed alert to parents.**

***Keywords-machine learning; Artificial Intelligence (AI); Natural Language Processing (NLP); WhatsApp; private message monitoring; Arabic text classification; message classification***

## I. INTRODUCTION

Today, more and more young people use and misuse technology. Unfortunately, criminals are focusing on tracking down vulnerable people, such as minors, to contact them through social media, using deception to lure their victims. If not identified and addressed in time, these risks can cause physical and psychological harm to a child. In this context, the

requirement for applications to alert about the presence of dangers becomes critical. Several applications help parents monitor private social media messages, but most of them lack Arabic language support.

The Modern Standard Arabic language (MSA), one of the numerous Arabic language formats, is used in official communications and is spoken in journalism and media. The Holy Quran, literature works, and old poems are written in another type of Arabic, known as Classical Arabic (CA). Another category is public dialects, which differ depending on location [1]. The Arabic language has 28 different alphabets, written from right to left, and the Arabic alphabet letters take on various forms depending on their location in a word. As Arabic is the fourth most used language on social networks [2], an application that can detect inappropriate messages is necessary. In this regard, the primary purpose of the proposed system was to create an application that can detect inappropriate messages in Arabic and after classifying the content as inappropriate, notify parents as soon as possible to save them from having to read the child's private messages and invading their privacy.

The proposed system was designed to classify inappropriate messages based on the Oxford Dictionary's definition: "inappropriate" is unsuitable behavior or language, such as sexual harassment or anything that causes damage or injury to a person, such as bullying. Such systems are significant because they aim to keep minors safe on social networks. The proposed system focuses on WhatsApp, but it is planned to improve and expand to detect inappropriate messages in Arabic on all social media platforms, including Twitter, Instagram, and Facebook. This system's application is named Etiqa'a because it can help parents avoid the evil or dangers that can happen through social media. The application monitors messages using Machine Learning (ML) and alerts parents when it discovers an inappropriate word. This application can serve Arab parents who have children aged from 7 to 16 years.

As social media bullying has become a hotly debated and vast topic, the Arab world is becoming more aware of cyberbullying. According to [3], approximately 60% of children from Arab Gulf countries openly admit the occurrence of cyberbullying among their peers. In [4], 20% of people aged 18-29 were reported to have been cyber harassed from the age of 15. In [5], 61.6% of 279 children in the age range of 12 to 19 years were reported to use WhatsApp, followed by Facebook (53%) and Instagram (52.3%). According to 60.4% of them, receiving bad messages and images was one of the most significant forms of bullying they experienced, while 45.6% experienced threatening and intimidating messages.

This study conducted an online survey to better identify the target age group and see if parents can look through their children's devices, answered by a total of 280 parents. The survey included questions such as whether they would use the Etiqa'a application and their opinion on whether it was ethical. Due to the results of the survey, shown in Figure 1, the target age group starts at the age of 7 to protect young children using WhatsApp. As 56.92% of parents said they do not have access to their children's devices if they are older than 17, the scope range was limited to minors up to the age of 16, excluding children younger than 7 due to their low use of WhatsApp (22%), and including ages from 7 to 16 due to their use of WhatsApp and parents having access to their devices. Teens aged 17 years and above were excluded.

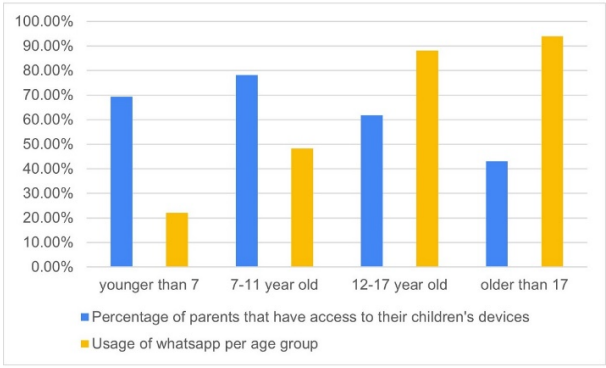


Fig. 1. Minors' accessible devices and use of WhatsApp.

Many parental control applications exist on the market, such as Bark, Qustodio, and AirDroid Parental Control. Although they all share the same idea of monitoring messages, many differences exist between them and the Etiqa'a application. Applications such as Qustodio and AirDroid do not have smart monitoring, which means that they enable the parent to see all messages and not just the inappropriate ones. Bark allows the parent only to see inappropriate messages, which is what Etiqa'a also does, protect children's privacy by not allowing the parent to see all the messages. In contrast, applications like Qustodio and AirDroid do not protect children's privacy. Another thing that Bark and Etiqa'a share is that they alert the parent if they find any suspicious content, while applications like Qustodio and AirDroid do not have an alerting feature. Bark has an alerting feature, but it does not alert in real-time, while Etiqa'a alerts the parent as soon as the child receives a suspicious message. The other applications do not support the Arabic language and need a subscription to be used, in contrast to Etiqa'a, which supports Arabic and is planned to be freely available without requiring any subscriptions. Table I compares Etiqa'a with these systems, showing their drawbacks.

TABLE I. COMPARISON BETWEEN ETIQA'A AND CURRENT SYSTEMS

| | Bark | Qustodio | AirDroid | Etiqa'a |
|---|---|---|---|---|
| **Messages Monitoring** | ✓ | ✓ | ✓ | ✓ |
| **Smart Monitoring** | ✓ | ✕ | ✕ | ✓ |
| **Suspicious Content Alert** | ✓ | ✕ | ✕ | ✓ |
| **Protect Children Privacy** | ✓ | ✕ | ✕ | ✓ |
| **Arabic Language Support** | ✕ | ✕ | ✕ | ✓ |
| **Free** | ✕ | ✕ | ✕ | ✓ |
| **Real-Time** | ✕ | ✕ | ✕ | ✓ |

In [6], a child protection application was suggested to protect children from online threats by analyzing children's online activities, monitoring the use of dangerous vocabulary, and informing parents of the dangers, but it has not been implemented yet. In [7], the difficulty and inaccuracies of

processing data on social networks due to short content were discussed. This study also discussed the massive flow of data that is constantly added, providing some successful solutions to overcome the problems of short text and large data flow. In [8], some of the challenges in Arabic Natural Language Processing (ANLP) were presented, providing some solutions that help overcome them. In [9], ML was used with Arabic social media content, reviewing and comparing the most commonly used ANLP tools with AML software to determine the best. In [10], bullying texts in YouTube comments were investigated using three ML algorithms: Complement Naïve Bayes (CNB), Multinomial Naïve Bayes (MNB), and Linear Regression (LR).

## II. METHODOLOGY

The Etiqa'a system was developed in two phases. Phase 1 requires a model that can process and classify Arabic text messages into appropriate or inappropriate labels according to their content. Figure 2 shows the process required to establish the model. Phase 2 consists of the development process of the mobile application, creating the database, exporting the trained model and vectorizer in a file, restoring them to test the model on new data, integrating the server with the client side, and lastly implementing tests for the application. The Android Virtual Machine (AVM) was used for simulations in Phase 2.

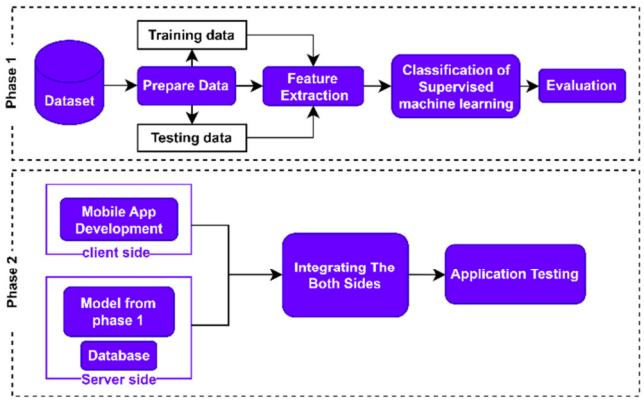


Fig. 2. Development phases of the Etiqa'a system.

### A. Dataset

The following datasets were used to detect inappropriate Arabic words: the Instagram Cyberbullying dataset [11], the Twitter Dangerous dataset [12], the Multi Platforms Offensive Language dataset [13], and the Twitter Offensive dataset [14]. As all these datasets had different labeling, all labels such as "toxic", "bullying", "offensive", "vulgar", "hate speech", and "dangerous" were unified under the label "NOT_APROP". On the other hand, all labels such as "positive", "nonoffensive", "clean", and "safe" were unified under the "APROP" label. After unifying the datasets and their labels, a single dataset was produced with 64K+ rows and two columns, one containing the text (message) and the other having the label indicating whether it is appropriate or not.

### B. Data Preprocessing

Preprocessing was used to prepare the data for training and classification [15], which consisted of cleaning the data, ANLP, and special feature extraction.

#### 1) Data Cleaning

The data were cleaned, including correcting grammatical errors, and removing duplicate or empty rows, non-arabic characters and emojis, and extra characters. In addition, misspelled words were corrected, and extended Urdu and Persian letters were mapped into normal (for example, all ؤۇۈۋۉۋ were mapped to و).

#### 2) Data Processing Using ANLP

After cleaning the data, data processing for the Arabic language was involved, which included tokenization, normalization, stemming or lemmatizing, and stop-word removal [15]. Stop words are a group of frequently used words that include determiners, conjunctions, and prepositions. These words are not worth a classification and were removed before training the model [16]. Six different stemmers and lemmatizer libraries were tested on the data to find which would yield the best result: CAMeL_lem, CAMeL_stem, tashaphyne, qalsadi, farasa_Stem, and ARLSTem.

#### 3) Feature Extraction

Because of the different nature of the data and the desire to find the best results, two different techniques were tested for feature extraction: The TF-IDF/Bigram TfidfVectorizer( ngram_range = ( 1, 2)) and the Count Vectorizer/Unigram CountVectorizer(ngram_range = ( 1, 1)) classes.

### C. Machine Learning (ML)

The ML model was built using a supervised method for classification purposes, which requires labeled data to be trained on to classify future data correctly with more accurate results.

#### 1) Algorithms Selection

The following algorithms were selected and tested to build a model according to relevant studies conducted on Arabic language models [15, 17]: Naïve Bayes classifier, Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and Decision Tree (DT).

#### 2) Model Training and Testing

Modules from the Scikit_Learn library were used to train and test the models. The training score for most models was above 90%. Each algorithm was tested using the two different vectorizers in four different dataset conditions:

- Extracted dataset features by lemmatizing text and maintaining the stop-words.
- Extracted dataset features by stemming text and maintaining the stop-words.
- Extracted dataset features by lemmatizing text and removing stop-words.

- Extracted dataset features by stemming text and removing stop-words.

Tables II and III show the accuracy scores obtained by training the model on the different algorithms mentioned.

TABLE II. ACCURACY SCORE FOR TRAINING MODELS USING TF-IDF

| Using TF-IDF Vectorizer | | | | | | |
|---|---|---|---|---|---|---|
| | ***NB*** | ***LR*** | ***SVM*** | ***KNN*** | ***RF*** | ***DT*** |
| **Lemmatized text with stop-words** | 77.8 | 76.6 | 77.2 | 72.2 | 76.8 | 70.3 |
| **Stemmed text with stop-words** | 78.4 | 77.1 | 77.8 | 72.2 | 75.2 | 68.5 |
| **Lemmatized text without stop-words** | 75.2 | 75.7 | 76.1 | 71.1 | 74.4 | 70.3 |
| **Stemmed text without stop-words** | 75.3 | 75.6 | 75.4 | 71.5 | 74.8 | 70.7 |

TABLE III. ACCURACY SCORE FOR TRAINING MODELS USING COUNT VECTORIZER

| Using Count Vectorizer | | | | | | |
|---|---|---|---|---|---|---|
| | ***NB*** | ***LR*** | ***SVM*** | ***KNN*** | ***RF*** | ***DT*** |
| **Lemmatized text with stop-words** | 76.5 | 79.2 | 76.5 | 54.8 | 76.5 | 71.1 |
| **Stemmed text with stop-words** | 76.4 | 78.9 | 76.3 | 54.8 | 74.6 | 68 |
| **Lemmatized text without stop-words** | 75 | 75.7 | 74 | 52.7 | 73.1 | 71.2 |
| **Stemmed text without stop-words** | 74.8 | 75.5 | 74 | 53 | 72.5 | 70.6 |

### *D. Mobile Application Development*

The application was developed using Flutter, a framework based on the Dart language created and supported by Google, using Visual Studio Code [18]. The application is used on the parent's and the child's device. At first, the interfaces were designed and then the services that each parent can access from his device were implemented, such as creating an account and logging in, using his email and password to use the application and its services such as adding a child, receiving an alert, getting advice, and adjusting account settings. Then, the services that the parent can access from the child's device were implemented, such as logging in, where the parent can enter his email and password and then give permission to allow the system to monitor notifications of the child's device to be processed later.

The notifications of Android applications can be read to gain access to WhatsApp messages using the Flutter notification_listener. Firebase Cloud Messaging was used to send an alert to the parent's device, which is a Google free cloud service that enables application developers to deliver messages and notifications to users across a variety of platforms, including Android, iOS, and web applications. FCM enables software developers to push application notifications to end users through an Application Programming Interface (API). FCM can send messages to apps in three different ways: directly to a single device, to a group of devices, or to devices that have subscribed to a topic [19].

### *E. Database*

The Etiqa'a system database structure was modeled using the ER-Diagram in Chen notation. Figure 3 shows its relational schema, which consists of 4 tables. The tables were created using PHPMyAdmin [20].

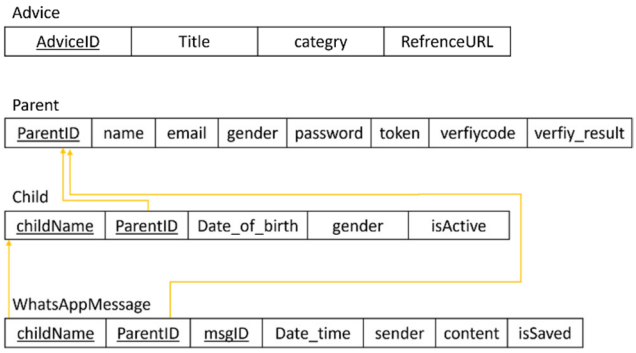


Fig. 3. Relational Schema of Etiqa'a system.

### *F. Integrating the Server Side to the Client Side*

Since Flutter cannot handle PHP, JSON [21] was used to integrate the database with the application, which is an interchange format for lightweight data. JSON was used for data exchange between the backend (PHP) and the front end (Flutter). Flask was used to integrate the model into the application, which is a popular microweb framework for Python API development that does not need special tools or libraries and aims to keep the core basic but extensible. Flask provides the fundamental components for development, such as request handling, routing, etc. In addition, it can be used to quickly and easily deploy ML models [22]. Once the ML model is exported, it can be turned into an API and send requests to it via a Flutter application.

### *G. Testing*

The application was tested to detect and resolve any problems and ensure that it was ready and free of errors that could impair its function or effectiveness.

- Unit testing was performed to isolate the code and ensure that it worked as intended. Each application function was tested and confirmed to be error-free.
- Integration testing was performed to expose any flaws when integrating components interacting with each other. The application's components were tested and confirmed to work perfectly together with no flaws or errors, such as integrating the ML model and the application.
- System testing was performed to evaluate the functionality of the application. The entire application was tested from the beginning to the end to confirm that it was bug-free, worked as expected, and that all the requirements were met.
- Acceptance testing was performed to assess the system's compliance with the requirements and verify if it met the

required criteria for use by end users. End users participated in the testing process and provided feedback on the application. Users were observed during this test, and notes were taken about their performance and behavior while using the application.

## III. RESULTS AND DISCUSSION

### A. ANLP Results

After testing the stemmers/lemmatizers, it was found that the CAMeL stemmer and lemmatizer worked the best for the given data.

TABLE IV. RESULTS OF STEMMERS AND LEMMATIZERS

| cleaned_text | CAMel_lem | CAMel_stem | tashaphyne |
|---|---|---|---|
| وجع بشكلك الله ياخذك | وجع شكل الله أخذ | وجع شكل الله أخذ | جع شكل له ياخذ |
| السلام عليكم مساء الخير كيف الحآل | سلام على مساء خير كيف حال | سلام علي مساء خير كيف حال | سلام على مساء خير كيف حال |
| ه رؤعة يأ زين الزين بسم الله عليك | ه رؤعه يا زين زين سم الله على | ه رؤعه يا زين زين سم الله علي | ه رؤع يا زي زين سم له على |
| لما الكلاب تهوهو عليك | ما كلب تهوهو على | لما كلاب تهوهو علي | لما كلاب هوهو على |
| روحي لبريده تلقين اشباه كثير بس محد زيكم مشفوح الله يفشلكم بس | روح بريد ألقى شبه كثير بس محد زي مشفوح الله فشل بس | روح بريد لقي أشباه كثير بس محد زي مشفوح الله فشل بس | روح بريد لق شبا ثير بس محد زي مشفوح له فشل بس |
| **qalsadi** | **farasa_stem** | **ARLSTem** | |
| وجع شكل الله ياخذك | وجع شكل الله ياخذك | وجع بشكل الل اخذ | |
| سلام على مساء خير كيف حال | سلام على مساء خير كيف حال | سلام علي مساء خير كيف حال | |
| ه رؤعه يا زين زين سم الله على | ه روع يا زين زين بسم الله على | ه رؤع يا زين زين بسم الل علي | |
| لما كلاب تهوهو على | ما كلب تهوهو على | لما كلاب هوهو علي | |
| روح بريد تلقي اشباه كثير بس محد زي مشفوح الله فشل بس | روح بريد تلقى اشباه كثير بس محد زي مشفوح الله فشل بس | روحي لبريد تلق اشب كثير بس محد زيكم مشفوح الل فشل بس | |

### B. Model Evaluation

The best accuracy score was obtained by the LR algorithm at 79.2% using the Count Vectorizer trained on lemmatized text with stop-words data, while the worst score was for the K-Nearest Neighbor with 52.7% accuracy.

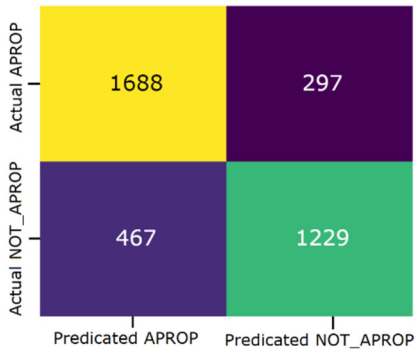


Fig. 4. Confusion matrix of the LR model.

To determine how well the proposed model operated and its highest accuracy results, the model was trained in different training sets, which required breaking the dataset into many distinct segments using the k-fold cross-validation technique. The best result was obtained with LR on lemmatized data with stop words, using a count vectorizer with an 81.2% accuracy score. The F1-score obtained from the k-fold cross-validation technique for the LR algorithm was 81.5%.

### C. User Interfaces

The principles considered when designing the application interfaces were: visibility, feedback, constraints, consistency, and affordance. When a user initially launches the application, a brief description appears, and then an interface appears with options for the user to enter the application either from the parent's or the child's device. The parent can log in or create a new account. If the login is performed from the parent's device option, the user can move to an interface that enables him to add a child, as shown in Figure 5 (the colored icon means the child's account is activated, and the black and white icon is the opposite). After adding the child, a user interface will appear containing the child's information through which he can modify and delete the child's account.


Fig. 5. Children's list interface.

The user should then log into the child's device with the same parent account information and select a specific child from his children to allow the application to access the device's notifications, as shown in Figure 6.

Fig. 6. Permission window in child's device.

When an inappropriate message is received on the child's device, the application will detect it immediately and send an alert to the parent's device, whether inside or outside the application, as shown in Figure 7. Alerts will appear for all children on the parent's home page for two weeks, while he can view alerts for a specific child if he wishes, as shown in Figure 8. As shown in Figure 9, when clicking on a specific alert, more details will appear, such as the full content, the sender's name, the time of the message, and whether he wants the alert to be stored in the alert history.

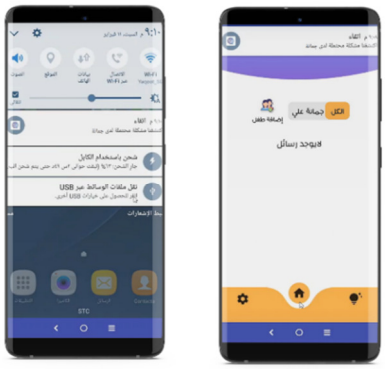

Fig. 7. Notifications pop up inside and outside the application.

Fig. 8. The homepage interface on the parent's device.

Fig. 9. Alert details interface.

The application also provides articles to get advice that could help the parent deal with or address the issue. In addition, the user can view and modify his account information, children's list, and alerts history, as well as a help center for the parent to explain how the application works and frequently asked questions and their answers.

### *D. Application Test Results*

The application was tested by nine different parents aged above 20 years. The tests were carried out in person, and the application was downloaded to user devices, who were then asked to use it. After the users were done using the application, a questionnaire was given to provide their feedback. 55.6% of the application users were technology experts, 33.3% considered their level in technology medium, and 11.1% considered themselves beginners. The users were asked to use the Etiqa'a application without limitation. Parents set up their accounts on their and their children's devices without problems, and then the parents sent messages to their children's WhatsApp accounts to test the application. The results showed that 25 of 28 messages (89.3%) were correctly classified, while 3 of 28 (10.7%) messages were misclassified, as shown in Table V. These misclassification problems were caused by the following reasons:

- The model was trained on a small and limited dataset.
- The tools that can correct Arabic dialect misspelled words.

All the parents reported that they received alert notifications about inappropriate messages, 88.8% of them found the application clear and easy to use, while 11.1% found it difficult to understand some of the instructions and phrases of the interface. Most parents suggested that the model should be taught more inappropriate words to improve its performance and found it valuable and very useful in protecting their children. Overall, all users were satisfied with the application and its services.

## IV. CONCLUSION

This study presented an application called Etiqa'a that analyzes WhatsApp messages using an ML model to classify them as appropriate or inappropriate. If a message is found inappropriate, the application sends an alert to the parent informing him of the message content and the sender. Four different datasets were combined, from different social media platforms, into one. The dataset content was then labeled as appropriate (APROP) or inappropriate (NOT_APROP). The dataset was cleaned, normalized, and tokenized using different stemmers and lemmatizes to find the best one to improve classification performance, which was provided by the CAMeL lemmatizer and stemmer. Data features were extracted using two methods, Count Vectorizer and Frequency-Inverse Document Frequency (TF-IDF), and the data were tested using six different algorithms. Each algorithm was tested using two different vectorizers under four different datasets. The results showed that the best results were achieved using the CAMeL lemmatizer without removing stop words for feature extraction. LR yielded the best results, with an accuracy of 81.2% and an F1-score of 81.5%.

TABLE V. USER ACCEPTANCE TEST

| User | Sent Sentence | Model's Classification | Was the message classified correctly? | Problem Justification |
|---|---|---|---|---|
| 1 | عمه في وجهك ياحمار ياوصخ | NOT_APROP | Yes | - |
| | كلب جحش حيوان غبي ثور | NOT_APROP | Yes | - |
| | اتوطا في بطنك | APROP | No | Training data size was small |
| 2 | السلام عليكم ورحمة الله وبركاته | APROP | Yes | - |
| | مساء الخير لمى | APROP | Yes | - |
| | وينك يا زق ليش ما تجين | NOT_APROP | Yes | - |
| | فاتك نص عمرك جنى الحيوانة اليوم جات | NOT_APROP | Yes | - |
| | يا حيوانه | NOT_APROP | Yes | - |
| 3 | السلام عليكم كيفك | APROP | Yes | - |
| | لو انك ولد امك و ابوك قابلني بعد المدرسة وتشوف ايش حسوي فيك | NOT_APROP | Yes | - |
| | والله اقتلك | NOT_APROP | Yes | - |
| 4 | ياكلبة | APROP | No | There are no good tools for correcting misspellings for Arabic dialects and training data were spelled correctly |
| | وريني صدرك واعطيك اللي تبين | NOT_APROP | Yes | - |
| | روحي انتحري محد يبغاك يا منبوذة | NOT_APROP | Yes | - |
| 5 | يا غبي | NOT_APROP | Yes | - |
| | ياحبيبي | APROP | Yes | - |
| | سمعت انك كلمت اخويا ع اللي صار امس | APROP | Yes | - |
| | لو سمحت لا تتدخل مرة ثانية ولا قسم بالله اقتلك | NOT_APROP | Yes | - |
| 6 | اذبحك | NOT_APROP | Yes | - |
| | غبيه ومحد يبغاك يامنبوذة | NOT_APROP | Yes | - |
| | فسخي وصوري لي وحاجيب لك | NOT_APROP | Yes | - |
| 7 | يا حقيره | NOT_APROP | Yes | - |
| | زباله انقلعي | NOT_APROP | Yes | - |
| | لا ترسلي شي | APROP | Yes | - |
| 8 | كل شي سيء في الحياه منك | APROP | No | Limited training data |
| | الله يلعنك | NOT_APROP | Yes | - |
| | شكلك المعفن ينحسنا | NOT_APROP | Yes | - |
| 9 | حلاة اللي يغتصبك ويقتلك بعدها يا سلام | NOT_APROP | Yes | - |

The application works differently on the parent's or the child's device. Parents can log into the application using an account. After the parent has set up the application on a child's devices, Etiqa'a monitors the notifications using the Flutter notification_listener plugin. Only WhatsApp messenger notifications are filtered and sent to the ML model to be analyzed and classified as appropriate or inappropriate. The ML model and the application were connected using the Flask Python API via a Flutter application. Each WhatsApp message read from the child's device is sent to the model to be cleaned, perform ANLP, and predict whether it is an appropriate or inappropriate message. If the ML model finds the message inappropriate, it stores it in the database, and sends a notification via FCM to the parent device to inform him about the content of the message.

The application was implemented for the Android platform. However, it can be ported to the iOS platform as well. In future work, the system will be able to monitor other social media applications and improve the message extraction method to overcome drawbacks, such as being unable to see the receiver's (child's) messages. The model can be further improved to recognize all inappropriate/harmful words and sentences, making the classification of messages more accurate by adding more specific categories, such as suicidal, bullying, and sexual harassment, and detecting not only inappropriate messages but also photos and voice messages. Another future objective is to make the application usable for everyone, including people with physical/visual disabilities.

## REFERENCES

[1] S. Larabi Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, and I. Abunadi, "Arabic Natural Language Processing and Machine Learning-Based Systems," *IEEE Access*, vol. 7, pp. 7011–7020, 2019, https://doi.org/10.1109/ACCESS.2018.2890076.

[2] O. Oueslati, E. Cambria, M. B. HajHmida, and H. Ounelli, "A review of sentiment analysis research in Arabic language," *Future Generation Computer Systems*, vol. 112, pp. 408–430, Nov. 2020, https://doi.org/10.1016/j.future.2020.05.034.

[3] "Cyber Safety Report - Research into the online behaviour of Arab youth and the risks they face," ICDL Arabia, 2015.

[4] F. A. Moafa, K. Ahmad, W. M. Al-Rahmi, N. Yahaya, Y. B. Kamin, and M. Alamri, "Cyber harassment prevention through user behavior analysis online in kingdom of Saudi Arabia (KSA)," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 6, pp. 1732–1746, Mar. 2018.

[5] B. M. Fahmi, "Cyberbullying among Adolescents on Social Media Networks," *Egyptian Journal of Public Opinion Research*, vol. 20, no. 3, pp. 289–335, Jul. 2021, https://doi.org/10.21608/joa.2021.198148.

[6] H. Ameur, A. Rekik, S. Jamoussi, and A. B. Hamadou, "ChildProtect: A parental control application for tracking hostile surfing content,"

*Entertainment Computing*, vol. 44, Jan. 2023, Art. no. 100517, https://doi.org/10.1016/j.entcom.2022.100517.

[7] F. Kateb and J. Kalita, "Classifying Short Text in Social Media: Twitter as Case Study," *International Journal of Computer Applications*, vol. 111, no. 9, pp. 1–12, Feb. 2015, https://doi.org/10.5120/19563-1321.

[8] A. Farghaly and K. Shaalan, "Arabic Natural Language Processing: Challenges and Solutions," *ACM Transactions on Asian Language Information Processing*, vol. 8, no. 4, Sep. 2009, Art. no. 14, https://doi.org/10.1145/1644879.1644881.

[9] T. Kanan *et al.*, "A Review of Natural Language Processing and Machine Learning Tools Used to Analyze Arabic Social Media," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan, Apr. 2019, pp. 622–628, https://doi.org/10.1109/JEEIT.2019.8717369.

[10] T. Alsubait and D. Alfageh, "Comparison of Machine Learning Techniques for Cyberbullying Detection on YouTube Arabic Comments," *International Journal of Computer Science & Network Security*, vol. 21, no. 1, 2021, https://doi.org/10.22937/IJCSNS.2021. 21.1.1.

[11] R. ALBayari and S. Abdallah, "Instagram-Based Benchmark Dataset for Cyberbullying Detection in Arabic Text," *Data*, vol. 7, no. 7, Jul. 2022, Art. no. 83, https://doi.org/10.3390/data7070083.

[12] A. Alshehri, E. M. B. Nagoudi, and M. Abdul-Mageed, "Understanding and Detecting Dangerous Speech in Social Media." arXiv, May 04, 2020, https://doi.org/10.48550/arXiv.2005.06608.

[13] S. A. Chowdhury, H. Mubarak, A. Abdelali, S. Jung, B. J. Jansen, and J. Salminen, "A Multi-Platform Arabic News Comment Dataset for Offensive Language Detection," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, Feb. 2020, pp. 6203–6212.

[14] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, "Arabic Offensive Language on Twitter: Analysis and Experiments." arXiv, Mar. 09, 2021, https://doi.org/10.48550/arXiv.2004.02192.

[15] T. Alqurashi, "Arabic Sentiment Analysis for Twitter Data: A Systematic Literature Review," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10292–10300, Apr. 2023, https://doi.org/10.48084/etasr.5662.

[16] I. A. Kandhro, S. Z. Jumani, F. Ali, Z. U. Shaikh, M. A. Arain, and A. A. Shaikh, "Performance Analysis of Hyperparameters on a Sentiment Analysis Model," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6016–6020, Aug. 2020, https://doi.org/10.48084/etasr.3549.

[17] W. M. S. Yafooz, E. A. Hizam, and W. A. Alromema, "Arabic Sentiment Analysis on Chewing Khat Leaves using Machine Learning and Ensemble Methods," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6845–6848, Apr. 2021, https://doi.org/10.48084/etasr.4026.

[18] L. Cianci, "Best IDEs for Flutter in 2022," *LogRocket Blog*, Feb. 21, 2022. https://blog.logrocket.com/best-ides-flutter-2022/.

[19] "What is Firebase Cloud Messaging (FCM)? | Definition from TechTarget," *WhatIs.com*. https://www.techtarget.com/whatis/definition/ Firebase-Cloud-Messaging-FCM.

[20] M. Kofler, Ed., "phpMyAdmin," in *The Definitive Guide to MySQL5*, Berkeley, CA, USA: Apress, 2005, pp. 87–116.

[21] "JSON." https://www.json.org/json-en.html.

[22] "FastAPI vs Flask: Comparison Guide to Making a Better Decision." https://www.turing.com/kb/fastapi-vs-flask-a-detailed-comparison.