Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 8, 2025

AN INFANT CRY INTERPRETER (BABBLE BOT)

Fatima Yaqoob[1], Laiba Shahid[2], Mahnoor Khalid[3], Sobia Riaz[4], Aasma Khalid[*5]

[1,2,3]*Dept. of Computer Sciences. The University of Faisalabad, Pakistan*
[4,*5]*Lecturer, Dept. of Computer Sciences. The University of Faisalabad, Pakistan*

[1]fatimayaqoob138@gmail.com,[2]laibashahid710@gmail.com, [3]mahnoork4112@gmail.com
[4]sobiariaz.CS@tuf.edu.pk, [*5]aasmakhalid.CS@tuf.edu.pk

DOI: https://doi.org/10.5281/zenodo.16835633

Keywords
LSTM, XGBoost, Deep Learning, Machine Learning

Article History
Received: 11 May, 2025
Accepted: 17 July, 2025
Published: 13 August, 2025

Corresponding Author: *
Aasma Khalid

Abstract

Babble Bot for Infant Communication. Infant crying is a key way by which babies communicate their needs, like hunger, discomfort, tiredness, burping, or pain. Innovating a system that identifies when a baby is crying and identifies the spe- cific reason behind it is included in the goal. The methodologies involved the collection of a dataset combined with algorithms for feature extraction and classification. LSTM model will be used for cry identification and the XGBoost model for reason finding. The cry detection system can tell when a baby is crying and ignore other sounds. Sends real-time alerts to caregivers and works well in different settings. This project is significant because it helps caregivers respond quickly and accurately to the needs of the baby. Knowing when a baby is crying allows caregivers to address things like hunger, discomfort, or pain immediately. This quick response improves baby comfort and reduces caregiver stress by giving them clear information on how to help. The cry detection system will be trained to ignore background noises such as talking, TV sounds, or other ambient sounds and focus only on baby cries. It will work in real time and send instant alerts to parents or caregivers through a mobile app or device. The system is designed to perform well in different environments such as homes, hospitals, or daycare centers. This project is important because it provides caregivers with clear and quick information. Knowing when and why a baby is crying helps them take the right action without delay. This fast response not only improves the comfort of the baby, but also reduces the stress and guesswork for parents, especially new or first-time parents. Furthermore, the Babble Bot can support healthcare staff in neonatal wards, helping to monitor multiple babies at once without missing any cries. Over time, the collected data can also be used for health tracking. For example, frequent pain-related cries might indicate a hidden health problem that needs medical attention. The system can also help detect unusual patterns in a baby's cries, which may help identify early signs of illness.Babble Bot aims to combine modern artificial intelligence with everyday parenting. It supports responsive care, improves infant well-being, and gives caregivers peace of mind. In the future, this system could be further developed with video monitoring, emotion recognition, or even suggestions for what action to take, making it a complete smart care assistant for infants.

INTRODUCTION

The target of the document is to outline the creation and implementation of a newborn's cry interpreter system. The mission of the system is to identify when a baby is crying and figure out why, like if the baby is hungry, uncomfortable, or in pain [1]. This will help caregivers meet the needs of children, improve care and reduce their anxiety [2]. The strategy aims to secure backing, resources, and authorization for the project by outlining its importance, goals, methodology, and anticipated results. Crying is how babies tell their needs and discomfort before they can speak. It helps parents know if their child is hungry, sick, tired, or in need of care and comfort. This communication is important for a child's health and development because it helps parents respond quickly and appropriately, creating a safe and loving environment. Understanding different types of cries can also help detect health problems early, leading to better care for the baby [3]. The program provides parents and caregivers with precious tools to help them better manage and respond to their infant's needs. This kind of system can be helpful not just at home, but also in hospitals, daycare centers, and places where many babies are being cared for. Nurses and caregivers can use it to monitor babies more effectively and give better care[12]. It can also help in areas where there are not enough doctors or baby experts. Another benefit is that the system can collect information about baby cries over time. This data can help researchers and doctors understand baby health better. For example, if a baby is crying in a way that shows pain or sickness, it might help doctors find problems earlier[10]. Also, for new or first-time parents, this system can be a great support. It can guide them through difficult moments and help them feel more confident in taking care of their baby.[11]. It can improve the bond between the parent and the baby and reduce the chances of the baby being left in distress. In the future, this technology could become part of smart homes, where baby cries are automatically detected and messages are sent to parents' phones or even to doctors if needed. It's a step toward smarter, safer, and more caring parenting using modern technology. In designing such a system, it is crucial to consider the real-time nature of baby monitoring. The system must be responsive, efficient, and capable of working continuously without delays or interruptions. It should be able to differentiate background noise from actual baby cries and provide accurate predictions even in noisy environments such as hospitals or busy households. Integrating machine learning and deep learning models like LSTM and XGBoost enhances the system's ability to classify audio with high precision. These models learn from audio features such as MFCCs, which mimic human auditory perception, making the system more aligned with how humans understand sound.

Moreover, this system aligns with the growing demand for personalized and assistive healthcare technologies. As populations grow and the need for remote healthcare solutions increases, such innovations provide scalable and cost-effective support. Integration with wearable devices or IoT systems can further increase its effectiveness by allowing continuous monitoring and quick interventions. With proper privacy safe- guards and ethical design, this cry detection and interpreta- tion tool has the potential to not only revolutionize infant care but also contribute to early detection of developmental delays or medical issues. Ultimately, this system can play a transformative role in strengthening child health services and empowering parents with knowledge and confidence. To ensure widespread adoption and usability, the system must be developed with accessibility and user-friendliness in mind. A simple and intuitive user interface, customisable alert settings, and support for multiple languages can make the system more inclusive for diverse users around the world. Addition- ally, incorporating feedback mechanisms where caregivers can confirm the system's predictions will allow for continuous learning and improvement of the model. This human-in-the- loop approach not only boosts accuracy over time but also helps build trust between users and the technology. As AI in healthcare evolves, systems like this can serve as a foundation for broader innovations in emotion-aware and context-aware infant care tools. In addition to its practical benefits, the system also opens new avenues for academic and clinical research. By collecting and analyzing large datasets of infant cries, researchers can uncover patterns linked to specific medical conditions, emotional states, or developmental stages. This data can be invaluable for pediatric studies and could con- tribute to early diagnosis of disorders such as colic, autism spectrum disorders, or hearing impairments. Collaborations between technologists, pediatricians, and psychologists can further enhance the system's capabilities, ensuring it evolves into a scientifically grounded and medically useful tool. Thus, beyond immediate caregiving support, the system contributes to a broader vision of data-driven child healthcare innovation.

LITERATURE REVIEW

As crying of infants is a major part in the life of the infants, parents and caregivers force many researchers and developers to work for the welfare of infants while also parents and caregivers get relaxed. Researchers contribute to this field in various ways; some apply machine learning techniques, while others use deep learning approaches [4]. Some achieve low accuracy, whereas others attain high accuracy. Everyone has their own way of feature extraction techniques. The paper ”How to use machine learning to detect if a baby is crying ” of year 2023 reveals that there is a system that uses a machine learning technique Support Vector Classifier (SVC) to detect if a baby is crying. The dataset consists of 4 classes; crying, laughing, noise and silence. Mel-frequency Cepstral coefficients (MFCC) methods were involved. The number of MFCC was 13. All the four classes were balanced. The system can detect only the newborn is weeping or not. The training accuracy was 100% which seems that model was over fit [5]. The paper ”Infant cry classification by MFCC features extraction with MLP and CNN structures” of year 2023 describes that there is a system that detects the reason of infants crying after taking the infant crying as input. It uses SVM, MLP and CNN algorithms to classify different types of sounds of crying. The data set consists of five classes; hungry, tired, discomfort, burping and belly pain. 19 MFCC features were used for model training. The dataset consists of only 315 audio files. The accuracy discussed in this model was 89%.They include the future work that accuracy should be improved [6]. In response to the previous paper a paper ”AI FOR INFANT WELL-BEING” reveals that they have used machine learning techniques like random forest to increase the performance of the system that predicts the reason after taking input of infant cry. But the dataset used was imbalanced that can bias towards the majority class. Accuracy was 94% [7].

Building on previous research, this system aims to overcome the limitations found in earlier studies by using more advanced models and a better approach to data handling. While some systems only detect if a baby is crying or not, our project not only identifies the presence of a cry but also determines the specific reason behind it. We plan to use a combination of LSTM and XGBoost models, which are known for their high performance in time-based and classification tasks. Unlike some previous studies that used small or imbalanced datasets, we are focusing on collecting a well-balanced and larger dataset to improve the system's accuracy and fairness. By combining powerful feature extraction techniques like MFCC with strong models, our goal is to develop a system that works accurately in real-world settings, reduces overfitting, and supports both parents and healthcare professionals. This contribution can help bridge the gap in current research by offering a more complete and practical infant cry interpretation system.

After discussing and reviewing several research papers on infant care and monitoring systems reveals that there are some systems that can detect whether the infant is weeping or not. On the other hand, there are some systems that can detect why the baby is crying. This system can only work well when input to the system is infant crying.[11] So there is a need for the integration of these two systems that continuously detect first the baby is crying or not, and if the baby or infant is crying then detect the reason and send alert to caregivers and parents. Another crucial aspect where this project stands out is its focus on real-time processing and adaptability in various environments. Unlike some previous systems that were limited to offline or lab-based testing, our approach integrates real- time audio input—both live and uploaded—with immediate classification and alert functionalities. This ensures that parents and caregivers receive timely notifications, especially in noisy or dynamic surroundings, such as hospitals or daycare centers. Additionally, by enabling email alert updates and continuous monitoring loops, the system is designed to adapt to ongoing infant behavior, ensuring reliability and responsiveness under different real-world scenarios.

Furthermore, this project lays the groundwork for future ex- tensions such as multilingual cry analysis, integration with mo- bile health apps, or even smart devices like baby monitors and IoT-based nursery systems. With continuous improvements in dataset collection and model training, the system can evolve to not only detect more subtle emotional cues but also contribute to longitudinal health tracking. These advancements can lead to early detection of neurological or developmental issues, ultimately expanding the role of technology in pediatric care. By combining technical excellence with a deep understanding of infant well-being, this research aspires to bring meaningful impact both in domestic and clinical contexts.

Requirement Analysis

The Babble Bot system aims to detect infant crying and determine the underlying reasons behind them using machine learning and deep learning techniques. To achieve this, both **functional** and **non-functional** requirements must be consid- ered during the design and development of the system.

Functional Requirements

The Babble Bot system is developed to perform a set of key functions that support real-time infant monitoring and assistance for caregivers.

The functional requirements include:

Real-Time Monitoring:
The system must continuously analyze live audio input to detect baby sounds in real time.

Cry Detection:
It should be able to distinguish baby cries from other sounds such as background noise, laughter, or silence.

Cry Reason Identification:
Once a cry is detected, the system must classify the reason behind the cry—such as hunger, discomfort, tired- ness, burping, or belly pain.

Confidence Score Display:
The system should provide the confidence percentage for the identified reason, helping caregivers gauge the certainty of the prediction[10].

Alert Notification:
Immediate alerts must be sent to caregivers via email when a cry and its reason are detected[9].

Email Update Option:
The system should allow users to update or change the email address where alerts are sent.

Non-Functional Requirements

These define system qualities and constraints:

Accuracy:
The system should provide high prediction accuracy, with minimal false positives or negatives.

Performance:
It must perform real-time processing with low latency to ensure timely alerts[13].

Scalability:
The system should be scalable to include additional cry categories or languages in the future[14].

Usability: The user interface (if any) should be simple and intuitive for caregivers with minimal technical knowl- edge.

Reliability:
The system must remain stable over long periods and function correctly even in noisy environments.

Portability:
The solution should be able to be deployed on mobile or embedded devices, allowing parents to use it conveniently at home or on the go[15].

Security and Privacy:
Audio data should be handled securely, respecting the privacy of babies and their families.

The first major requirement is **Cry and Reason Recog- nition**, which is the foundation of the Babble Bot sys- tem[16]. This feature leverages trained Machine Learning (ML) and Deep Learning (DL) models to detect baby cries and determine the underlying cause, such as hunger, discomfort, sleepiness, or the need for attention. The system uses labeled datasets comprising different types of baby sounds for training and testing. Feature extraction techniques such as MFCC (Mel Frequency Cepstral Coef- ficients) are applied to convert raw audio into meaningful patterns that the models can learn from. To ensure the accuracy of this component, a variety of cry and non- cry sounds are used during the testing phase. The goal is to assess the model's ability to distinguish between different emotional tones and provide reliable predictions with confidence scores.

The second essential requirement is **Microphone Access**, which allows the system to capture live audio from the environment. This is implemented using device-specific libraries or web-based APIs that request permission from the user to access the built-in microphone. The system ensures compatibility with various platforms, includ- ing mobile devices and desktops. Microphone access is critical for feeding real- time data to the classification engine, enabling continuous monitoring. During testing, different devices are used to confirm that microphone permissions are handled correctly and that the system successfully receives and processes audio input without delay or interruption. This step guarantees that the system is responsive and functions seamlessly across different environments[17].

The third requirement, **Well-Defined System Interface**, focuses on the user experience (UX) and system usability. A clear and intuitive interface is designed using modern UI components, such as buttons, icons, status indicators, and responsive layouts. This ensures that users, especially caregivers or parents, can easily understand and operate the system. The interface includes visual feedback, such as confidence levels of cry detection, audio activity indicators, and alert notifications. Usability testing is performed with real users to gather feedback on the interface design, navigation flow, and overall comfort[18]. The goal is to make the system not only functional but also accessible and user-friendly, even for those without technical knowledge.

Lastly, the **Real-Time Recognition and Alert System** plays a vital role in timely monitoring and response. This feature is implemented by developing a background service that continuously records audio, processes it through the recognition model, and generates alerts (such as emails or mobile notifications) when a baby's cry is detected. The real-time nature of this feature ensures that no critical moment is missed, making it ideal for situations where constant supervision isn't possible[19]. To test this component, scenarios are simulated where baby cries occur randomly, and the system's ability to detect them and send alerts promptly is evaluated[20]. This ensures that the system maintains high performance and reliability during prolonged operation.

PROPOSED FRAMEWORK

Babble Bot is a smart tool that helps detect parents when a baby is crying and what might be the issue behind it, even if there's a lot of noise around. It uses a special program to tell apart different sounds like cries, laughs, and background noise. By studying many examples of baby cries, the system will learn to understand the unique features of a cry. When the system hears a baby cry, it can tell if it's hungry, tired, burping, belly pain or uncomfortable by analyzing the sound. It then send alerts to caregivers, helping them respond accurately. This will make it easier to take care for a baby, ensuring they get the care they need even when there's a lot going on. A system that detects and understands the cries of the baby is precious because it helps parents respond quickly and keeps the baby safe. It will also help physicians by providing information about the health and behavior of the infant [8].

The table provides a side-by-side overview of the **categories used in two separate models** involved in an infant sound classification system: the **Cry Model** and the **Reason Model**. It does not show the relationship between these models but rather lists the distinct output classes that each model is trained to recognize.

On the left side, the **Cry Model** categories include four types of sound classifications: Crying, Laughing, Noise, and

Silence. These represent the general types of audio the system might detect when monitoring an infant. The Cry Model is responsible for determining which of these categories an incoming audio clip belongs to.

Categories of Cry Model	Categories of Reason Model
Crying	Hungry
Laughing	Discomfort
Noise	Uncomfortable (e.g., wet diaper)
Silence	Belly pain
—	Burping

Fig. 1. Categories.

On the right side, the **Reason Model** categories are listed: Hungry, Discomfort, Uncomfortable (e.g., wet diaper), Belly pain, and Burping. These are the potential reasons behind a baby's cry and are the output classes of a separate model that analyzes audio features only when a cry has been identified.

The LSTM model is used for cry identification and XG-Boost model for reason finding. The dataset for the cry model is obtained from Kaggle. The dataset for reason is from donating a cry corpus that is also used in many researches but the problem is that it is an imbalanced dataset which can predict towards the majority class. It is balanced by data augmentation techniques like adding background noise in audios so that all classes will have an equal number of audios.

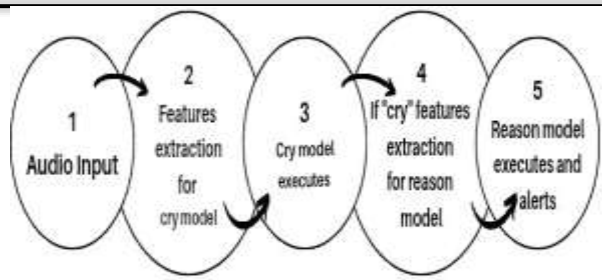


Fig. 2. Proposed Framework.

In the **proposed framework**, incoming audio is first passed through a pre-processing step to remove noise and enhance cry signals. Then, the LSTM model checks if the sound is a baby cry. If it is, the cry is passed to the XGBoost classifier, which analyzes it and predicts the possible reason. Finally, the result is sent to the caregiver through an app or notification system, giving messages like "Baby may be hungry" or "Check for discomfort."

Babble Bot is designed to work in **real-world environments** like homes, hospitals, and nurseries. It is helpful for caregivers who may be away from the baby or looking after multiple children. It can be added to **baby monitors**, **smartphones**, or even **wearable devices**. Over time, the system can also track patterns in the baby's cries and help identify recurring health or emotional issues.

In the **future**, the system can be improved further by using **larger datasets**, **multilingual support**, or even combining with **video monitoring** and **facial recognition** to understand the baby's emotions better. The feedback from parents can also be used to improve the system, making it more adaptive and personalized.

Overall, Babble Bot offers a **complete and intelligent solution** to support infant care. It reduces stress for caregivers, improves baby comfort, and opens the door to smarter, AI- based parenting support tools.

In conclusion, **Babble Bot presents a comprehensive and intelligent solution for infant care** in the modern age. By combining advanced audio processing, machine learning, and real-time alert systems, it reduces the stress and uncertainty faced by caregivers. It enables timely and informed responses to a baby's needs, ensuring greater comfort, safety, and emo- tional well-being. Beyond just recognizing cries, Babble Bot paves the way for a new generation of **AI-powered parenting tools** that support personalized care, early issue detection, and overall peace of mind for families. Its adaptability and scalability make it a promising addition to the future of smart childcare technology.

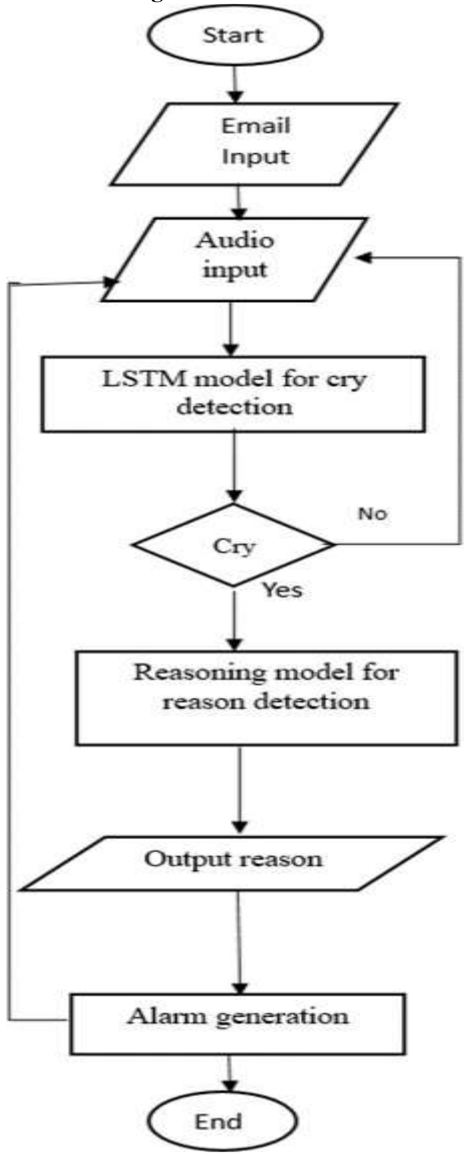
Bot system—a real-time baby cry detection and reasoning tool. The process starts with the user providing an **email input**, which is necessary for the system to send alerts or notifications later. After that, the system activates the **audio input** module, where it continuously listens for sounds using the device's microphone. The captured audio is then passed into an **LSTM (Long Short-Term Memory) model**, which is trained to recognize patterns in audio signals and determine whether the sound is a baby's cry.

If the LSTM model **detects no cry**, the system loops back to continue monitoring audio in real-time. However, if the model confirms a cry, the audio is forwarded to a **reasoning model**, such as an **XGBoost classifier**, which analyzes the cry further to identify the most probable reason behind it—this could be hunger, discomfort, sleepiness, or pain. The detected reason is then passed to the next step for interpretation and formatting. In the **"Output reason"** stage, the identified cause is prepared in a user-friendly message format (e.g., "Baby may be hungry").

Finally, this message triggers the **alarm generation** stage. The system sends an alert to the previously provided email address or through a mobile notification. This alert helps caregivers respond quickly and appropriately. After the alert is sent, the system returns to the initial monitoring stage to continue listening for new cries, ensuring continuous opera- tion. This flowchart efficiently models a loop-based, real-time AI system that integrates audio processing, cry classification, and actionable caregiver alerts.

ALGORITHM

Fig. 3. Flowchart.

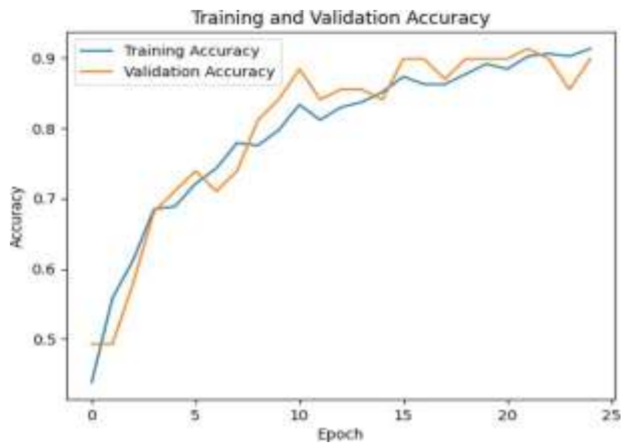


This flowchart outlines the working logic of the **Babble**

Cry Detection:

In the project there is a Multimodal Neural Network is formed for cry detection. The algorithm involves building a neural network architecture that takes audio features as inputs and learns to classify cry, laugh, noise and silence from audio samples. Regularization techniques are applied to avoid over fitting. This table outlines the architecture of a Sequential LSTM model used for classification.It includes two LSTM layers with 50 units each, followed by Dropout and a Dense layer.The model takes input in the shape of (timesteps, features).The output is a Softmax layer predicting across num_labels classes.The model uses the Adam optimizer and is trained for 25 epochs. The testing accuracy is 93 percent. The cry detection model used in the Babble Bot system is built using a **Sequential LSTM (Long Short- Term Memory)** architecture. LSTM is a type of deep learning model that is particularly effective for analyzing sequential data like audio signals. This model is designed to take in time-series input, making it well-suited for understanding baby cries that unfold over time. The model consists of **two LSTM layers**, each having **50 units**, which help capture the patterns in the audio sequences. To avoid overfitting and ensure better generalization to new, unseen data, a **Dropout layer** is included. This layer randomly deactivates some neurons during training, which prevents the model from relying too much on any single feature.

Fig. 5. Training and Validation accuracy.



Component	Details
Model Type	Sequential LSTM
Layers	Two LSTM layers (50 units each) + Dropout + Dense
Input Shape	(timesteps, features)
Output	Softmax layer with num_labels classes
Optimizer	Adam
Training Epochs	25

Fig. 4. Architecture.

Following the LSTM layers, a **Dense layer** is added, which acts as a fully connected layer and plays a crucial role in final decision-making. The **input shape** expected by the model is in the form of **(timesteps, features)**, where ”timesteps” represent the number of segments or windows in the audio, and ”features” refer to the characteristics extracted from each segment, such as MFCCs. At the end of the model, a **Softmax layer** is used to generate probability scores for each class. This means the model will predict the likelihood of each possible cry type (e.g., hungry, tired, pain, etc.), and the highest-scoring class will be chosen as the output.

To optimize the learning process, the **Adam optimizer** is used. Adam is known for being efficient and effective, espe- cially for models involving a lot of parameters and complex data. The model is trained for **25 epochs**, meaning the entire dataset is passed through the model 25 times during training. This helps the model refine its understanding of the patterns in baby cries and improves its classification accuracy over time. The plot illustrates how well the LSTM model learns to identify baby cries during training and how well it performs on unseen validation data. The **blue line** represents the **training accuracy**, while the **orange line** represents the **validation ac- curacy**. At the beginning (around epoch 1), both training and validation accuracies are quite low—around 45–50%—which is expected since the model is just starting to learn. As the number of epochs increases, both lines rise steadily, indicating that the model is improving its understanding of the cry patterns.

Around **epoch 10 to 15**, the validation accuracy fluctuates a bit but generally stays high, even reaching around **90%**, which is a strong sign that the model is generalizing well to unseen data. Meanwhile, the training accuracy also continues to rise and eventually approaches **92%**, showing that the model is fitting the training data very well. The fact that both training and validation accuracies are close together suggests that the model is **not overfitting**, and it's performing well both during training and in practical use. This confirms that the Babble Bot's cry detection model is reliable and effective for real- time baby cry classification.

Fig. 6. Training and Validation loss

The graph shows the training and validation loss of the model over 25 epochs. Both losses steadily decrease, indi- cating that the model is learning effectively during training. The validation loss closely follows the training loss, which suggests that the model is generalizing well to unseen data and not overfitting. Although there is a slight fluctuation in validation loss after epoch 20, this is minor and expected in real-world scenarios. Overall, the loss values reach a low point near the end of training, showing that the model's performance is improving and making fewer errors. This indicates a well-trained and reliable model.

In addition to the steady decline in both training and valida- tion loss, the graph demonstrates that the learning process was smooth and stable. At the beginning (epoch 0), the loss values were high, which is expected as the model starts with random weights. As training progresses, both losses consistently drop, showing that the model is successfully learning from the data. The gap between training and validation loss remains small throughout most of the epochs, which is a positive sign indicating that the model is not memorizing the training data (overfitting), but instead learning patterns that work well on new, unseen data.

Around epochs 20 to 25, we observe a slight increase or fluctuation in the validation loss, while training loss continues to decrease. This could be an early sign of overfitting, but since the increase is small and temporary, it doesn't significantly affect the model's overall performance. Such variations are common and can be controlled with techniques like dropout, regularization, or early stopping. Overall, the graph suggests that the model is well-optimized, with good generalization ability and minimal error by the end of training.

For identifying the reasons behind the baby's cry, the model uses XGBoost for multi class classification using softmax. The model is saved using pickle. The reason de- tection uses machine learning algorithm XGBoost is used because as in dataset all the audios are baby's cry so to differentiate mix type of data, this algorithm is the best.From the audios MFCCs, chroma and spectral features are ex- tracted. The XGBoost model architecture shown uses the

Cry Reason Detection:

XGBoost Model Architecture

Component	Details
Model Type	XGBoost Classifier (xgb.train with DMatrix)
Objective Function	multi:softmax (Multiclass classification)
Number of Classes	*len* (np.unique(*y_train*)) (Depends on dataset — e.g.,5 for 5-class problem)—g
Evaluation Metric	merror (Multiclass classification error rate)
Maximum Boosting Rounds	100
Early Stopping	30 rounds (monitors validation loss)
Data Format	X_train, y_train
Validation Dataset	X_validation, y_validation
Test Dataset	X_test, y_test

Fig. 7. Architecture of XGBoost.

XGBoost Classifier trained with DMatrix and is config- ured for multiclass classification using the multi:softmax objective function. The number of output classes depends on the dataset, calculated using the unique labels in the training set. The evaluation metric employed is merror, which measures the multiclass classification error rate. The training process allows a maximum of 100 boosting rounds, with early stopping set to 30 rounds based on validation loss. The data is formatted as X_train, y_train, with X_validation, y_validation used for validation and X_test, y_test for final testing.

ADVANTAGES

Early Identification of Baby's Needs

By detecting crying and classifying the reason behind it (e.g., hunger, discomfort, belly pain), the system helps care- givers respond **quickly and accurately**, improving infant care and reducing stress for both baby and parents.

Non-Intrusive Monitoring

The system works by passively analyzing sound, meaning it does not require any physical contact or wearable sensors on the baby. This ensures **comfort and safety** while still providing constant monitoring.

Real Time Alerts

It is integrated with a web application, the system can pro- vide **real-time notifications** or alerts to parents or caretakers when crying is detected and a specific reason is identified. This is especially helpful during sleep or when the caregiver is in another room.

Supports New or Busy Parents

For first-time parents or those managing multiple tasks, the system offers an intelligent assistant that **interprets a baby's cries**, helping them learn and respond more confidently without guessing the reason behind the cry.

Enhanced Pediatric Care

In hospital NICUs (Neonatal Intensive Care Units), such a system could assist nurses by providing **automated monitor- ing**, helping prioritize attention when babies cry due to pain, discomfort, or hungry.

SYSTEM TESTING

Manual testing is essential in the Babble Bot system to evaluate how well it functions in realistic scenarios. While automated tests can validate technical correctness, manual testing gives insight into how actual users will experience the system. It allows testers to interact with the graphical interface, observe how intuitive the buttons and flows are, and assess whether the predictions and alerts are clear and timely. This process is particularly useful in uncovering design flaws, usability barriers, and unexpected issues like delayed responses, mislabeling, or unresponsive buttons. Furthermore, it enables testing under varied conditions, such as different room acoustics, accents, or overlapping background sounds like music or television noise. This diversity ensures the system is inclusive and robust, especially for multilingual households or environments with variable noise levels.

System testing ensures that the complete system works as intended and meets all specified requirements. In the Babble Bot project, this phase tested whether live and uploaded audio could be handled correctly, whether predictions appeared on the interface, and whether alerts were successfully sent. It also examined how the system responded to real-time conditions, such as noisy environments and internet disruptions. Notably, all core features passed except for the scenario involving poor or no internet connectivity, where the system failed to maintain expected performance. This highlights the need for either offline functionality or more robust error-handling mecha- nisms—perhaps a fallback queue that temporarily stores alerts and sends them once connectivity is restored. System testing also reveals whether all subsystems (audio capture, prediction, UI, and alerting) function cohesively under extended and real- life operational conditions.

Unit testing was crucial in confirming that each core function of the Babble Bot system works reliably on its own. Components like MFCC feature extraction and model loading were tested individually to ensure they correctly handled valid inputs. Additionally, the models themselves—LSTM for cry detection and XGBoost for reasoning—were evaluated using test datasets to verify their performance and loading stability. Both models demonstrated strong predictive accuracy (above 90%), which builds confidence in the underlying AI. However, a flaw was identified in the email validation function: while it correctly accepted syntactically valid emails, it failed to verify their existence, allowing invalid or fake entries. This unit-level issue can be addressed by incorporating backend email validation services or two-step email confirmation to enhance reliability and avoid false notifications. Unit testing plays a foundational role in ensuring that no faulty logic passes into the integrated system.

Functional testing focuses on validating each user-facing feature to confirm it aligns with the expected requirements. All core features—cry detection, reason prediction, email alerts, form validation, and UI interactivity—were tested thoroughly. Input constraints were validated (e.g., disallowing names that start with digits, ensuring no empty fields), and the dynamic behavior of the email field (changing and storing updated input) was verified. All tests passed except one—again linked to the system accepting non-existent emails. This consistency in failure across unit and functional testing indicates a gap in user data verification that could affect user trust and system credibility. Aside from that, functional testing affirmed that the system's front-end and back-end interactions (such as clicking "Record" and receiving predictions) worked seamlessly, offer- ing a stable and intuitive experience to end users, especially caregivers.

Integration testing examines whether all modules interact as expected when combined into a full system. In Babble Bot, this involved connecting the audio recording module with the feature extraction pipeline, linking the LSTM output to the XGBoost reasoning model, and finally connecting the prediction output to the notification system. Most of these interactions performed successfully; however, the integration of live audio with the model and email system faced is- sues—mainly due to compatibility between audio formats and real-time model inference. Such issues can arise from inconsistent sample rates, encoding formats, or file types (e.g., mismatched .wav and .ogg formats). Moreover, the pending status of the email/SMS integration indicates an operational bottleneck that could delay critical alerts in real use. These challenges highlight the importance of continuous integration and testing to catch such mismatches early. By identifying these gaps and iterating improvements, integration testing ensures that the final system operates smoothly as a unified product, rather than just a collection of well-performing mod- ules.

The testing was mostly done by hand. Testers followed test cases and tried different ways to check if the software worked properly. Because the testing was manual, exact numbers like how many bugs were found or how much of the software was tested were not always written down. Still, testers made notes about what they saw, like how many problems they found, how much of the software they tested, and how long it took. This type of testing helped them look closely at how the software worked and how easy it was to use. They were able to find and fix many issues, making sure the software worked as expected. After testing, the team improved the test cases, increased the amount of testing, and focused on the parts that needed more checking.

Conclusion

In the conclusion, babble bot system uses machine learning and deep learning techniques to accurately identify and ana- lyze baby sounds. Real-time processing is used to give instant feedback based on live audio. This project helped learn more about handling audio data and programming. Overall, valuable experience is gained in creating a smart system with practical benefits for child care.The development of this project offered valuable learning opportunities in handling complex audio datasets, applying advanced preprocessing techniques, managing imbalanced data through augmentation, and working with powerful classification algorithms. Furthermore, it provided practical experience in integrating AI models into a real- world use case that can support parents, nurses, and even pediatricians. The Babble Bot system has the potential to evolve into a more advanced tool with future improvements, such as integrating mobile applications, expanding multilin- gual audio datasets, and incorporating health tracking features. Overall, the project delivers a meaningful contribution to the field of infant care technology and showcases how artificial intelligence can be leveraged to enhance everyday life.

Acknowledgment

We would like to express our sincere gratitude to Almighty Allah for giving us the strength and ability to successfully complete our Final Year Project.

We are deeply thankful to our parents for their continuous prayers, moral support, and encouragement throughout this journey.

Our special thanks go to our respected supervisor for their valuable guidance, patience, and constructive feedback, which played a crucial role in shaping our project. We are also grateful to all our teachers for sharing their knowledge and expertise, which helped us overcome challenges during the project development.

Lastly, we appreciate the collaborative efforts of all our team members, whose dedication and hard work made this project possible.

REFERENCES

Abbaskhah And H. Sedighi , ”Infant Cry Classification By Mfcc Feature Extraction With Mlp And Cnn Structures,” Biomedical Signal Processing And Control, 2023.

Bashiri And R. Hosseinkhani, ”Infant Crying Classification Using Genetic Algorithm And Neural Network,” Acta Medica Iranica, Pp. 531-539, 2020.

Maiti And C. Dutta, ”Ai For Infant Well-Being: Advanced Techniques In Cry Interpretation And Monitoring,” Journal Of Mechanics Of Continua And Mathematical Sciences , 2024.

Rosales-Pe´Rez, ”Classifying Infant Cry Patterns By The Genetic Se- Lection Of A Fuzzy Model,” Biomedical Signal Processing And Control, 2015.

Sinha, ”Ai Infant Listener: Deep Learning For Automatic Baby Cry Recognition And Analysis Using Iot,” Aip Publishing, Vol. 3075, 2024.

Belyk M, Brown S. The Origins Of The Vocal Brain In Humans. Neurosci Biobehav Rev. (2017) .

C.-Y. C. Chang And C.-W. Chuan-Wang Chang, ”Dag-Svm Based Infant Cry Classification System Using Sequential Forward Floating Fea- Ture Selection.,” Multidimensional Systems And Signal Processing, P. 961–976, 2017.

Chen Z. Structure And Function Of The Vocal Vocal Organs. Changchun,

Jilin Province: Modern Music (2020). P. 159–60

F. S. Matikolaie And C. Tadj, ”Machine Learning-Based Cry Diagnostic System For Identifing Septic Newborns,” Journal Of Voice, 2022.

Rani And P. Kumar, ”Baby Cry Classification Using Machine Learn- Ing,” International Journal Of Innovative Science And Research Technol- Ogy, Vol. 7, P. 677–681, 2022.

Rani And P. Kumar, ”Baby Cry Classification Using Machine Learn- Ing,” International Journal Of Innovative Science And Research Technol- Ogy, Vol. 7, P. 677–681, 2022.

Basavaiah, ”A Pragmatic Approach For Infant Cry Analysis Using Support Vector Machine And Random Forest Classifiers,” Wireless Pers Communication, P. 2269–2280, 2024.

Zhang, ”Baby Cry Recognition Based On Woa-Vmd And An Improved Dempster–Shafer Evidence Theory,” Computer Methods And Programs In Biomedicine, Vol. 245, 2024.

Jiang And Y. Yi, ”A Novel Infant Cry Recognition System Using Auditory Model-Based Robust Feature And Gmm-Ubm,” Concurrency And Computation Practice And Experience, P. 33, 2021.

Lagasse Ll, Neal Ar, Lester Bm. Assessment Of Infant Cry: Acoustic Cry Analysis And Parental Perception. Ment Retard Dev Disabil Res Rev ,2005 .

Paulsen F, Tillmann B. Structure, Function And Insertion Of The Human Vocal Folds. Adv Otorhinolaryngol. (2020)

Tao S. The Basic Structure Of The Singing Vocal Organs And The Principles Of Vocalization. Xi’an City, Shaanxi Province: The World Of Music (2015)

Wen K, Guo Z, Peng Y, Zhou X, Su C. Research On Infant Cry Classification Algorithm Based On Acoustic Feature Extraction. Inf Rec Mater,2021.

Wasz-Hockert O, Lind J, Vuorenkoski V, Partanen T, Valanne E, Illing- Worth R. The Infant Cry. A Spectrographic And Auditory Analysis.,1968.

Xu X, Zhang L, Qu Y, Mu D. Research Progress On Physiological And Pathological Factors Of Infant Cry. Chin J Child Health Care,2018.