12-2022

Protectbot: A Chatbot to Protect Children on Gaming Platforms

Anum Faraz
af1653@rit.edu

Protectbot:
A Chatbot to Protect Children on Gaming Platforms

by

Anum Faraz

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering

Department of Electrical Engineering and Computing Sciences

Rochester Institute of Technology, Dubai
December, 2022

Protectbot: A Chatbot to Protect Children on Gaming Platforms

by

Anum Faraz

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Electrical Engineering

Department of Electrical Engineering and Computing Sciences

Approved By:

Dr. Jinane Mounsef Date

Thesis Advisor/ Assistant Professor of Electrical Engineering - Department of Electrical Engineering and Computing

---

Dr. Ali Raza Date

Thesis Co-Advisor/ Associate Professor of Computer Sciences - Department of Electrical Engineering and Computing

---

Dr. Boutheina Tlili Date

Thesis Committee member/ Associate Professor of Electrical Engineering - Department of Electrical Engineering and Computing

Acknowledgments

I would like to express my gratitude to my advisor Dr. Jinane Mounsef for her invaluable supervision, support and tutelage during my master's degree. Her efforts and guidance encouraged me at every stage of my research. Besides my advisor, I would like to thank the rest of my thesis committee members: Dr. Ali Raza, Professor in Computing Sciences at RIT Dubai and Dr. Boutheina Tlili, Professor in Electrical Engineering at RIT Dubai for their insightful comments and feedback which allowed me to widen my research from various perspectives. I would like to thank Dr. Sandra Willis, Director of Policy & Advocacy in the Global Mental Health Laboratory, Columbia University, New York, USA for her guidance and support to broaden my research by adding a physiological perspective to my research. I would like to thank Fardin Ahsan for helping me with the coding of my project.

Finally, I would like to thank my parents, my husband and my son. It would have been impossible to finish my research without their motivation and support over the past few years.

Abstract

Online gaming no longer has limited access, as it has become available to a high percentage of children in recent years. Consequently, children are exposed to multifaceted threats, such as cyberbullying, grooming, and sexting. The online gaming industry is taking concerted measures to create a safe environment for children to play and interact with, such efforts remain inadequate and fragmented. Different approaches utilizing machine learning (ML) techniques to detect child predatory behavior have been designed to provide potential detection and protection in this context. After analyzing the available AI tools and solutions it was observed that the available solutions are limited to the identification of predatory behavior in chat logs which is not enough to avert the multifaceted threats. In this thesis, we developed a chatbot *Protectbot* to interact with the suspect on the gaming platform. *Protectbot* is using the dialogue generative pre-trained transformer (DialoGPT) model which is based on Generative Pre-trained Transformer 2 (GPT-2). To analyze the behavior of the suspect we developed a text classifier based on natural language processing which can classify the chats as predatory and non-predatory chats. The developed classifier is trained and tested on Pan 12 dataset. To convert the text into numerical vectors we utilized

fastText. The best results are obtained by using non-linear SVM on sentence vectors obtained from fastText. We got a recall of 0.99 and $F_{0.5}$-score of 0.99 which is better than the state-of-the-art methods. We also built a new dataset containing 71 predatory full chats retrieved from Perverted Justice (PJ). Using sentence vectors generated by fastText and K- nearest neighbour (KNN) classifier, 66 chats out of 71 were correctly classified as predatory chats.

Contents

List of Tables

List of Figures

1. Introduction

1.1 Motivation

The rapid growth of technology has remarkably transformed the way people connect with each other. The Internet is becoming a crucial source of information and entertainment. Social media, instant messaging, and audio/video calling platforms have become major sources of communication. Among the 4.95 billion Internet users, 1 in 3 are under 18 years of age and often use the Internet without the supervision of an adult [1], [2]. With the rapidly increasing use of the Internet among children and adolescents, it has become more important to provide them with a safe and secure environment. Children can face various threats while being involved in different online social activities that could involve exposure to violent content. Child harassment and pornographic content, cyberbullying, child victimization, abuse, grooming, sexting, and pedophilia are also among the common and serious threats children can face while socializing online with strangers or even with peers [3].

According to [4], more than 90% of children in the US play online games. This figure increases to 97% among children aged 12-17. Online gaming is considered a source of learning that aims to enhance children's cognitive abilities [5]. These games provide a

useful means of building leadership qualities in children [5] and enhance teamwork skills [6]. Moreover, multiple games are developed to help children in educational fields, such as learning science and mathematics [7–9]. Online gaming is also found to be associated with positive outcomes, such as enhanced social relationships. However, problematic outcomes are also associated with excessive online gaming, such as negative emotions and attitudes, low self-esteem, loneliness, anxiety, poor academic performance, and maladaptive coping strategies [10]. Notably, mobile game addiction is associated with social anxiety, depression, and loneliness, with male adolescents reporting the highest social anxiety when gaming excessively [11]. Few studies have examined the relationship between game addiction and mental health outcomes, due to a lack of standardized instruments required to measure this new type of behavioral addiction. Gaming platforms provide public chat rooms, private chat rooms, group chat rooms and in-game chatting for online gamers to interact with each other. The same chat rooms on these platforms can also pose a considerable risk to children.

A survey of 10-17 years old children in the US showed that 56% of child abuse incidents occurred on social networking sites, while 11% occurred in online video chat rooms and 6% in game chat rooms or gaming sites [12]. Age-inappropriate games that include sexual, abusive and self-harm content are also a source of threat to children, knowing that the content could be a shared media over the chat. Gaming platforms provide parental control features to monitor the online content presented to children along with a variety of options to limit the contact with gamers that use the chatting features. Parental controls do not only provide an option to sway the children from being exposed to inappropriate content, but they also provide an opportunity for parents to restrict the socialization with strangers. However, the effectiveness of the parental controls remains debatable due to a

deficiency of detailed knowledge and understanding of the control features. The lack of parental awareness about the control tools is a serious factor that hinders children's online safety [13], [14].

Policymakers and stakeholders are working along with governments to ensure effective vulnerable child protection online by placing protective measures and developing interactive user-friendly tools to enhance the knowledge of children and parents about online threats and the ways to combat them [15–18]. Nevertheless, several challenges must be overcome to keep children safe online whilst enabling them to benefit from digital engagement opportunities.

Artificial Intelligence (AI) is widely used in gaming platforms through different applications. Most research in the literature is related to the development of AI agents to play games and compete with humans as opponents [19–24]. The detection of predatory behavior on gaming chat platforms using AI tools is a growing field. Renowned AI techniques such as supervised learning require labelled datasets that are not widely available. However, raw chat logs are accessible and public. The authors in [25–28] picked chat logs from selected games, such as MovieStarPlanet [25], Online Battle Arena [26], World of Tanks [27], and Dota Ragnarok [28], to detect predatory behavior in online gaming. Nonetheless, most research [29–52] has not been conducted in the context of online gaming but rather on chat logs related to social media websites, instant messaging applications, and public chat rooms. Despite the increasing number of cases of cyberbullying and child sexual grooming in the gaming environment, solutions to children's safety on gaming platforms using AI tools have not been adequately addressed in the literature. The main focus of the research remains restricted to the detection of predatory behavior in chat logs. This solution is not enough to protect the child from predators, as it is only a step toward

building a mechanism to protect children online.

In this work, a chatbot *Protectbot* is designed to provide a protection mechanism for children from predatory behavior in any gaming environment. The basic task of *Protectbot* is to act like a child and identify the predators by chatting with the suspects. The *Protectbot* is using the pre-trained model Dialogue Generative pre-trained Transformer (DialoGPT). The DialoGPT is based on Generative Pretrained Transformer (GPT2). The *Protectbot* is equipped with the text classifier to detect predatory behavior in chats. The developed text classifier is trained and tested on Pan 12 dataset [53]. The text classifier showed the best results as compared to the proposed models in the literature. The accuracy, recall, F_1-score and $F_{0.5}$-score of 0.99 are achieved by the proposed model. An integrated framework is developed to identify the child predators in gaming platforms and generate reports for Law Enforcement Agencies (LEA) as shown in Figure 1.1.

1.2 Purpose and Contributions

The purpose of this project is to develop an integrated framework to provide a mechanism to protect children on gaming platforms. The main contributions of our work are multifold and include:

- Examining the different aspects of child safety by highlighting the existing threats to children in gaming environments on the one hand, and the existing protection mechanisms provided by the industry, researchers, international and national law makers, and regulators.
- A substantial survey of the different AI tools applied in the gaming environment including the detection of child predatory behavior.

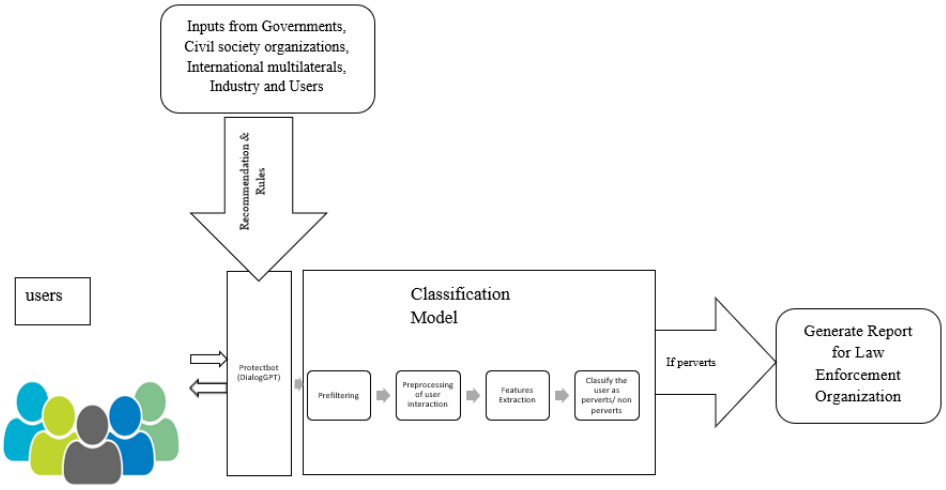

Figure 1.1: Protectbot-Framework

- Highlighting the need to leverage AI technology to identify the pervasiveness, type, and risks associated with predatory behavior in online gaming and its potential effects on children's and adolescents' mental health and overall protection.
- A dataset containing 71 predatory chats is developed. The chats were retrieved from the website PJ.
- A chatbot *Protectbot* is proposed to identify the predators by analyzing their behavior in chat logs. *Protectbot* is integrated with the Natural Language Processing (NLP) model to classify the chat.
- The text classifier used in *Protectbot* is trained and tested on the publicly available dataset Pan 12. The testing of the model shows that our NLP model is showing better performance than the models present in the literature.

1.3 Research Methodology

To investigate the available solutions present in the literature, a survey was conducted to identify relevant research that addresses child safety in gaming environments or chatting platforms using Machine learning (ML) algorithms and AI tools. The survey considers only the work published between 2017 and 2021 using academic portals that can access research databases, including IEEE Explorer, Elsevier, Springer, ACM, Cambridge, Wiley, ProQuest, and Sage. We use the following keywords to search the portals without filters: "cyberbullying and AI", "cyberbullying and ML", "child predators, AI and games", "cyber threats", "cyber predators", "child paedophiles", "child pedophiles", "detect child predators", "predatory behavior and games", "child pedophilia and online games", and "child safety online games".

The research focuses on the literature that describes the threats to online gaming children and presents a solution to detect and combat child predators on gaming platforms and chatting platforms available as part of social media.

Initially, 1073 references were found based on a keyword search. After filtering out newspapers, magazine articles and book chapters, 550 papers were selected, which were then filtered based on their titles and keywords, keeping 200 relevant papers. The abstracts of these papers were analyzed to filter out 100 papers; the remaining 100 papers were reviewed in detail, leaving 29 papers that fulfilled the required criteria. The second stage of the search included a comprehensive review of relevant cross-references that added five more papers to the previous ones, resulting in a total of 34 papers reviewed. Table 1.1 provides a summary of the datasets and threats covered in the selected papers.

In addition to the review of ML and AI techniques for enhancing children's online safety, this work presents an overview of the existing laws, policies and regulations to

protect children and recommendations provided by International Telecommunication Union (ITU) and UNICEF to stakeholders, including children, parents, caretakers, educators, industry, and policymakers. This work also presents a review of tools and solutions proposed by the industry and organizations to enhance Child online protection (COP).

After surveying the existing solutions to protect children from predators we developed an integrated platform to protect children from online predators as shown in Figure 1.1. We developed a chatbot *Protectbot* to chat with suspects on any gaming platform. We also developed an improved text classifier; using the developed classifier the chats are classified into two categories i.e. predatory or non-predatory. We trained and tested the text classifier on the only publicly available dataset Pan 12 [53]. The developed text classifiers showed an accuracy, recall, F_1-score and $F_{0.5}$-score of 0.99 which is better than the performance shown by proposed models in the literature. If predatory behavior is detected in chats a report is generated for LEA.

1.4 Structure of the thesis

Chapter 2 provides an overview of the threats to children on gaming platforms along with the current international laws and policies to protect children on gaming platforms. The recommendations to stakeholders provided by government multilateral and government organizations are also included in Chapter 2. Chapter 3 includes the related work used in the design of *Protectbot*. This Chapter also includes the theoretical concepts related to NLP, ML used in the *Protectbot*. Chapter 3 also includes a detailed description of the integration of chatbots with the gaming platform. Chapter 4 explains the *Protectbot* architecture, design and its working in detail. The results obtained from the classifier

Table 1.1: Studies with datasets and identified threats.

Ref.	Year	Datasets					Covered Threats			
		PJ	**PAN12**	**PAN13**	**Collected from gaming sources**	**Combination of datasets/other sources**	**Cyberbullying**	**Sexual threats**	**Cyberpredators**	**Other**
[53]	2012		✓					✓		
[29]	2013	✓						✓		
[25]	2015				✓				✓	
[26]	2015				✓					✓
[30]	2017							✓		
[31]	2017						✓			
[32]	2017			✓				✓		
[33]	2017			✓				✓		
[27]	2018				✓		✓			
[34]	2019					✓		✓		
[35]	2019			✓				✓		
[36]	2019					✓			✓	
[28]	2019				✓		✓			
[37]	2019					✓		✓		
[38]	2019					✓		✓		
[39]	2019	✓						✓		
[40]	2019		✓						✓	
[41]	2019					✓	✓			
[42]	2020					✓		✓		
[43]	2020					✓		✓		
[44]	2020		✓					✓		
[45]	2021		✓				✓			
[46]	2021		✓					✓		
[47]	2021		✓					✓		
[48]	2021					✓				✓
[49]	2021		✓					✓		
[50]	2021		✓					✓		
[51]	2021		✓					✓		
[52]	2022		✓						✓	

and its comparison with the literature are also presented in Chapter 4. Finally, Chapter 5 concludes the thesis along with future work and recommendations.

2. Background

2.1 Introduction

This Chapter provides the background research which motivated us to propose a chatbot *Protectbot*. This Chapter provides an overview of the threats faced by children in online gaming environments. The current international laws and policies to protect children on gaming platforms are also discussed in this Chapter. This Chapter also summarizes the available solutions developed by industry, government multilaterals, parents and caretakers, to protect children in gaming platforms. The limitations of the available solutions are also included which shows the need for an integrated solution to protect online children. The summary of the several recommendations to all stakeholders provided by government multilaterals and government organizations is also included. This Chapter highlights the lack of protection mechanisms to protect children from online threats on gaming platforms which provided motivation to develop a solution that could protect children on any gaming platform. The material covered in this Chapter has been published in [54].

2.2 Online Threats to Children

2.2.1 Threat Types

Online gaming is a popular leisure activity for children, but it also poses many threats. The digital environment provides great opportunities for children to learn in all disciplines, but at the same time, it poses a multitude of threats from organizations, adults, and peers. Online threats to children are broadly classified into three categories: content, contact, and conduct risks [55].

Content Risk

Content risk includes exposure to inappropriate content such as adult, violent, extremist, and gory content. The assurance of content related to self-harm, self-abuse, destructive, and racist ideas is also considered a content risk. Exposure to incomplete and inaccurate information is another way to affect children's understanding of the world around them.

A research by UNICEF explored the consequences of exposure to game content and its potential effects on children's social relationships, education, physical activity, mental well-being, and psychological or developmental challenges such as depression, social anxiety, stress and excessive play [56]. Despite the volume of research, to date, the results from studies on the effects of online gaming on children's well-being, whether positive or negative, have been mixed [57–60]. Evidence of the impact of online gaming on children has been oriented toward exploring issues or building initial theories, but it is not robust or reliable enough to inform policy decisions or best practice recommendations. This is the case not only for studies examining the influence of online gaming on children but also for research exploring the influence of digital technologies more broadly [61–64].

Contact Risk

Children can face a broad range of contact threats from their adults and peers. Contact risk includes harassment, exclusion, defamation, victimization, child pedophilia, and grooming. A summary of the contact risks is presented below.

- Bullying is a common threat that children face online or in real life. The effects of bullying have been widely studied. Depression, anxiety, panic disorders, distributed personality, suicidality, criminality, and illicit drug misuse are common effects reported by people who face online or physical bullying during childhood [65]. In [66], the study showed a positive relationship between peer cyberbullying and suicidal ideation among young children and adolescents.
- Children with disabilities are more prone to experiencing victimization, including bullying, harassment exclusion, and discrimination.
- Defamation of a child, such as sharing images and/or videos or sharing altered images and/or videos of a child, can put him or her in complete devastation.
- Children can also be targeted, groomed and sexually abused by adults pretending to be someone they are not.

In May 2019, in California, a man was sentenced to 14 years in prison to force an 11-year-old girl to produce child pornography [67]. He approached the girl through the Clash of Clans game. A similar case was reported in suburban Seattle, where a man was caught by the LEA for blackmailing three boys and forcing them to share inappropriate photos. He was posing as a teenager and approached the victims through Minecraft and League of Legends [67]. In a recent event in May 2022, a 17-year-old boy committed suicide after being scammed by a man posing as a girl [68]. According to a CNN report, the scammer shared a nude photo and asked the victim to share his photo. After receiving

the photo from the victim, the scammer started to extort him to send him money, but the victim was unable to arrange the money. He then committed suicide and his family came to know about the whole situation through the suicidal note.

Conduct Risk

Conduct risk includes the children behaving as perpetrators. Children can play a role in victimizing their peers. This includes harassment, bullying, sexting, exclusion, shaming, and the generation of inappropriate content by the child [69]. The different scenarios of conduct risk can be summarized as follows.

- Online bullying is more damaging than real-life bullying because it can spread in less time and the shared content or images are available for a longer period; hence, it is harder for the victim to overcome the embarrassing situation.
- Children are responsible for plagiarism, such as uploading pictures of others, without their consent.
- Children can use disrespectful names to harass or bully their peers.
- A very common behavior observed in adolescents is sexting, which involves sharing sexualized images and/or text via messages. The outcome of sexting is wide, ranging from positive and accepted to negative and unwanted [70]. The photos produced for sexting can be distributed to a wider audience, often leading to embarrassment, harassment, and placing adolescents in vulnerable positions [71].

2.2.2 Current Laws to Protect Children

The Electronic code of federal regulation (eCFR) is a compilation of the material published by the Code of Federal Regulations (CFR) and the Federal Register amendments produced

by the National Archives and Records Administration's Office of the Federal Register (OFR) and the Government Publishing Office [72]. The regulations of eCFR related to children's usage of online platforms can be summarized as follows.

- It is unlawful for the operator of any platform to collect the personal information of a child without providing a written notice on the website.
- Parents should be informed of the collection of personal data and a verifiable parental consent is required prior to the collection, use, and disclosure of the data.
- The notice should clearly mention the information type that is collected and the way it is used by mentioning the disclosure practices.
- It is the responsibility of the operator to use the available technology to verify that consent has been provided by the parents.
- Once the verification process is complete, the information collected for parents identification should be deleted by the operator immediately from the company's record.
- Parental consent is mandatory for the approval of transactions made using the platform's online payment system.
- The operator is required to provide a reasonable platform for the parents to review the collected personal information of the child in order for them to allow or deny further use of the information.
- Parents should be given the opportunity to refuse or permit the operator to delete or use the provided information.
- The operator is not allowed to condition a child's participation in games by requiring the child to provide additional personal information necessary to participate in any activity.

- The operator is required to protect the confidentiality, security, and integrity of the children's personal information and to ensure that the information shared by a third party takes care of the integrity of the personal data.

2.3 Existing Protection Mechanisms for Children's Online Gaming

2.3.1 Children's Rights and Gaming Environment

The multitude of risks faced by children in the gaming environment also poses a threat to the infringement of their rights. UNICEF prioritizes engagement with the information and communication technology (ICT) industry and works in the following areas to increase children's safe usage of the Internet and the associated technologies by tackling different issues, such as the transmission of children's online sexual abuse images, exposure to inappropriate content or contact, and violation of the child's privacy [12]. UNICEF is also working with corporate partners that harness ICT to provide children with opportunities to become engaged digital citizens and use ICT platforms for learning, sharing, and communicating [15], [17].

UNICEF presented the children's rights related to the positive and negative impacts of online gaming in a discussion paper [12]. Children's rights related to online gaming should be taken care of by all stakeholders to protect the child from risk. Many of UNICEF children's rights can be associated with the gaming industry: acting in the best interest of the child (Article 3), a parental guidance consistent with the child's evolving capacities (Article 5), the right to leisure, play and culture (Article 31), the protection of a child

from sexual abuse (Article 34), parents' primary responsibility for the upbringing and development of the child (Article 18), children's right to non-discrimination (Article 2) and freedom of association (Article 15), respect for the views of the child (Article 12), the children's right to freedom of expression (Article 13), the protection of privacy and personal information (Article 16), the protection of the child from all types of exploitations (Article 36), and the right to education (Articles 28 and 29). All the protection mechanisms developed for the children must uphold their rights while online.

2.3.2 Stakeholders and their Roles

This section summarizes the guidelines and effective tools created by international and national organizations to enhance the COP for relevant stakeholders including children, parents, caretakers, educators, industry, and policymakers. In November 2008, the ITU launched the COP initiative as a multi-stakeholder global initiative to create a safe and empowering online experience for children [15]. The COP guidelines have served national government entities, civil society organizations, industry, and many other stakeholders in their children's online protection efforts. Moreover, the COP's initiative attained further endorsement and validation during the 2018 Plenipotentiary Conference of the ITU held in Dubai. The multitude of stakeholders also framed the protection of children online within the framework of the United Nations Convention on the rights of children and other human rights treaties. To address the exploding and transforming threats to online children, the COP-updated guidelines were launched in June 2020.

Children

Children are key consumers in a gaming environment. Therefore, they need to be educated on their rights and the protection of their rights. The ITU launched an online safety course with Sangophone (Sango), where a child online protection mascot equips children with the knowledge they need to know about their rights and responsibilities when they are online [73]. The five episodes of this course have been launched to address the multiple issues a child can face online, such as inappropriate content, sharing of personal information, the vulnerability of different threats while using social media applications and gaming environment, downloading, and in-store purchase of games, etc. The ITU created three different resources to guide children in different age groups. A storybook with questions was developed for children under nine to provide them with an understanding of their rights and safety online. A workbook containing educational activities was designed for children aged 9-12 years. Through these activities, children can learn about their rights and online risks. A social media campaign was created for children aged 13-18 to help them learn how to manage risks online.

Parents/Caretakers & Educators

Parents, caretakers, and educators are responsible for the wellbeing of children. Therefore, they must play a positive role in protecting children online. The ITU provides recommendations to parents/caretakers and educators to understand children's vulnerabilities and the best protective measures to safeguard them [74]. These guidelines include all the major threats a child can face online and how parents/educators can help children by providing the right and complete information needed to protect them online.

The ITU recommends the parents to:

- Have a discussion with their children about the vulnerabilities and mechanisms available to protect against them by joining the children in their online activities.
- Monitor all devices used by their children including their mobile phones, laptops, tablets, gaming consoles, fitness trackers, smart televisions, and applications used on any of these devices.
- Install firewall and antivirus software on all the devices. Parental controls and filtering are useful tools; however, children's privacy should also be considered.
- Control their children's access to age-appropriate websites, set rules such as screen time, and teach their children about their privacy issues.
- Create a positive environment so that children can express their problems and opinions. Many websites may not ask parents' permission for their children to join a website or a platform.
- Be aware of the minimum age requirements for their children to use these platforms.
- Be aware of the unauthorized access to the debit or credit cards through their children's account by controlling the use of cards and other payment mechanisms.
- Be aware of reporting a person or inappropriate content on any platform their children use.
- Talk to their children about any advertisement that might be misleading and inappropriate.
- Educate their children about the threats related to their relationship with strangers.
- Be aware of the people their children are chatting with or meeting with online.
- Teach the children about the privacy issues and managing their personal information online.
- Inform their children that photos can reveal a lot of personal information and thus,

explain the risks associated with uploading photos or any other confidential content.

- Tell the children about obtaining their parents' consent before sharing any information or photos of their family or friends.

Educators were also provided with guidelines to protect children from online threats. The ITU recommends that teachers and other relevant school staff:

- Ensure that all devices are password-protected and that the antivirus and firewalls are updated.
- Communicate a clear policy about how technology can be used to students and their parents.
- Acquire parents' consent when taking photos of children and sharing them on social media platforms.
- Ensure that inappropriate content is filtered and monitored via the Internet network provided by the school.
- Raise awareness of the importance of the digital footprint and online reputation.
- Understand the importance of professional online communication with students, parents, and other stakeholders.
- Have knowledge of risks and vulnerabilities students can be exposed to when they are online.

Industry

ITU guidelines are helpful in creating a connected framework for the COP to create harmony among all stakeholders. To accomplish this, the industry is also provided with guidelines to play a role in the COP. The industry includes Internet service providers, social networks, messaging and gaming platforms, hardware and software manufacturers,

companies providing digital media and several services, such as streaming, digital file storage and cloud-based. The ITU recommendations to the industry require the following actions in collaboration with international and national governments and law enforcement organizations:

- Identify, prevent, and mitigate the adverse impacts of ICT on children's and adolescents' rights by developing child protection, safeguarding policies, and integrating risks and opportunities into company-wide policy commitments [75].
- Play a role in combatting Child Sexual Abuse Material (CSAM) and prohibiting the uploading or sharing of content that violates the rights of any party.
- Provide users with a comprehensive way to report any inappropriate content and take prompt actions against them in accordance with the international and national government and law enforcement organizations.
- Be responsible for actively monitoring any content hosted on the company's server on a regular basis using tools, such as hash scanning of known children's abuse images, image identification software and URL blocking to oversee CSAM.
- Make sure to provide a safe and enjoyable digital environment for children and adolescents by providing age-appropriate content to children of different age groups, enhancing the existing parental control features and developing new tools using technological advances to help the COP.
- Create an efficient framework for the awareness of customers related to spam, data theft and inappropriate contact, such as bullying and grooming, and educate them on the procedures to combat them.
- Invest in research and develop tools or educating material to enhance children's, parents', caretakers', and educators' knowledge about the children's rights and

protection mechanisms provided by the industry itself, international and national governments, policymakers, and other law enforcement agencies.

The European Commission suggested that the industry be self-regulatory, as self-regulations allow the latter to create their own system by which they can deal with the challenges a child can face online [76].

Policymakers

The ITU highlights the need for policy frameworks to address all harms against children in the digital environment but at the same time, this should not unduly restrict children's rights. The national-level recommendation provided by the ITU is summarized as follows [77].

- It is important to note that any illegal act against children in the real world is illegal online. Therefore, framing legal regulations for online data protection and privacy rules for children is necessary.
- Self-regulatory or co-regulatory policy development is required along with the full regulatory framework.
- A mechanism should be established and promoted to report any illegal content as well as reporting user issues or concerns.
- Research is required to engage all stakeholders to determine their opinions, ideas, experiences, difficulties, and opportunities for COP.
- Digital literacy features should be a part of the national school curriculum that is applicable to children of different age groups.
- Educational resources should be developed to reflect cultural norms and laws and to enhance the COP.
- National awareness campaigns are needed to highlight the COP's related issues.

- The understanding of tools, applications and settings that help COP should be evaluated and improved.

2.3.3 Tools for awareness of end consumers

The ITU in collaboration with the National Cybersecurity Authority (NCA) of the Kingdom of Saudi Arabia launched an ITU global program on children's online protection in December 2020 [78]. The work stream of this project was divided into two stages. In the first stage, cyber-skill development is provided to train children, adolescents, parents, and educators. It also intends to develop a game and an application for children of different age groups to understand the COP guidelines by the end of 2022. The translation of the COP guidelines into national languages and the use of the Sango tool are also included in the scope of this first stage [73]. The second phase of this project aims to provide national strategy development on COP and capacity building for ICT professionals and government international and national stakeholders. This phase will be completed by 2024.

UNICEF and the Global Partnership to End Violence Against Children launched an AI-based game, which is a social-emotional learning tool that teaches children the skills needed to stay safe and protected online [17]. This tool was designed for 5-10 years old children to learn the skills necessary to protect themselves. It also provides parents and teachers with guidelines on how to teach their children about online safety.

In the UAE, the Ministry of Community Development, in collaboration with the Telecommunication Regulatory Authority, launched a digital platform kidX that provides an interactive environment using games and virtual reality technologies to raise children's and adolescents' knowledge about online safety [16].

Finally, the European Union (EU) developed an interactive platform to enhance parents'

and children's awareness by creating different scenarios related to online grooming, sexual exploitation, and domestic violence [18].

2.3.4 Tools for safety of end consumers

Age-Appropriate Rating

Age-appropriate ratings provide guidelines to consumers including children, parents/caretakers and the industry regarding the age group for which a game is appropriate. Established in 2013, the International Age Rating Coalition (IARC) provides a globally standardized age classification process for digital games and mobile applications [79]. IARC is administered by games rating authorities including the Australian Classification Board from Australia [80], the Classificacao Indicativa (Classind) from Brazil [81], the Game Rating and Administration Committee (GRAC) from the Republic of Korea [82], the Entertainment Software Rating Board (ESRB) from North America [83], the Pan European Game Information (PEGI) from Europe [84] and the Unterhaltungssoftware Selbstkontrolle (USK) from Germany [85]. The IARC system was established in close collaboration with rating authorities, game developers, and game retailers. The rating contains a three-part categorization that suggests age appropriateness, content descriptors that indicate the content type that may have triggered a particular rating, and interactive elements that advise about several risks, such as sharing the user's location with other users, or the fact that personal information may be shared with third parties. A free of cost application can be downloaded from Google Play or the Apple Store to check the rating of any video game and obtain insight into the game content.

A developer must submit the deployed game along with the questionnaire available on the IARC website. Based on the information provided, the IARC assigns age ratings

and content descriptors in accordance with regional rating authorities and the interactive elements are assigned universally. The IARC rating authorities check the ratings assigned to the game to ensure the accuracy of the age rating if needed, and corrections are implemented by the storefronts. Developers need to submit an IARC report when the game is submitted to a publisher. The rating system is not only used by game publishers and retailers but also by parents who are the key consumers. The rating system provides guidelines to the parents by describing age ratings and content descriptors. The parental controls also used the age rating to filter games for the different age groups of children.

The Entertainment Software Association (ESA) conducted a survey on the video game industry in July 2021, where approximately 4000 participants from the US took part [86]. According to the survey, 86% of parents were aware of the ESRB rating and 76% were using it to protect their children online. The survey, which only involved the US population, should be extended to other countries to provide more comprehensive insight into the usage of age-appropriate ratings.

Gaming Platforms and Parental Controls

In this section, selected gaming platforms are presented along with the parental control features provided by these platforms. Table 2.1 summarizes the gaming platforms with their chat features and parental controls. Each of these platforms, along with the features provided, are described below.

Roblox can be accessed through its website on desktop computers [87]. For mobile users, the application is available in play stores. It provides a large variety of games from different genres. There is no restriction on age to make an account or play a game, but for children aged under 13 years, the game limits the user's account to a restricted view of the

Table 2.1: Popular gaming platforms and their features.

Platforms	Platform type	Age restriction	Voice messages	Video chat	Content sharing	Chat filtering	Chat restriction	Monthly spend restriction	Customized blocking	Screen time limit	Additional parental control features
Roblox	App/ Website	No Restriction	Yes	No	No	Yes	Yes	Yes	Yes	No	N/A
Steam	App	13	Yes	No	Yes	Yes	Yes	No	Yes	No	N/A
PS5	Console/ App	18, under 18 can use PS through family account	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Xbox	Console/ App	18, under 18 can use *Xbox* through family account	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Nintendo	Console/ Website	13, under 13 can use *Nintendo* through parent account	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

website/application. It also provides basic filtration of games, which is not suitable for players under 13. An age verification process is also introduced in which users need to upload any identity card along with their photo, but this is just an optional requirement. *Roblox* provides multiple chat features including chats with friends, group chat rooms, and chats with unknown people while playing a game. Chatting communication is written only and does not include voice messaging. As there is no age restriction for playing a game, children as young as five years can have a chat with an unknown person. Game developers claim to have a combination of chat filters that are both human-controlled and machine-moderated to filter inappropriate content, words associated with bullying and harassment, and personal information shared in the chat. *Roblox* provides a wide variety of parental control options. To use them, parents must generate a pin to secure the settings of the child's account. Parents/caregivers can link their account to the child's account and control or restrict the platform's available features. For example, a parent can place restrictions on persons who can message their children, chat with them in the application and chat with them in the games. Multiple options are provided to a parent including "no one can chat", "only friends can chat", and "anyone can chat". A purchase notification can also be enabled, and parents can receive a message whenever their children buy an item. A

monthly spending limit can also restrict children from purchasing items from the game store. *Roblox* also provides an option to enable "Account Restriction" that blocks all the chats and prevent anyone from searching the account. Only a suitable pre-approved list of games and content is available for the restricted account.

Steam provides a wide variety of games that a user can play using the application on a desktop as well as on mobile devices [88]. An account is required to play a game and a minimum age of 13 years is required to create an account. However, parents can create a "Family view" account to enable any family member to access the game. The "Family view" option in the *Steam* platform provides options to restrict the accessible features that enable parents to choose the games that would be visible in the "Family View" for up to 10 different accounts. A parent can also disable access to the *Steam* store, chat rooms, friends list and the online profile of the primary user (parent). Content sharing is also available such as screenshots, video clips, game play broadcasts, and the user-generated data. *Steam* also provides users with the capability to chat with friends/group, voice message, and share content including photos and videos. All these features can be blocked in the "Family view" mode. In addition, *Steam* provides a wide variety of content filtering options for blocking mature content, frequent violence, nudity, sexual content, and adult content using the settings. Any related content game can be blocked or made inaccessible. *Steam* also blocks strong profanity and slurs in chatting platforms. Additional words or tags associated with games that need to be blocked in chats can be specified in the settings. These content and chat filtering options are applied to the primary and secondary accounts used in the "Family View".

Play Station 5 (PS5) is a popular console developed by Sony for playing games [89]. To use the console, the user must create an account with a minimum required age of 18.

Parents can link children under 18 years old to their own account. *PS5* provides multiple chat features including chats with friends, group chat rooms, chats with unknown people while playing games, voice messaging, and video chats. The web browsing option is also available for children who can access any content online. Content sharing, such as screenshots, video clips, game play broadcasts, and user-generated data, is another feature of *PS5*. *PS5* provides a wide variety of parental control features ranging from content filtering to restricted communication. In content filtering, the application requires the user to enter the age of the child in account settings. Content and games are filtered according to age. Manual filtering is also permitted such that parents can customize and create a list of games that the children can access. Multiple chat options are provided to a parent to choose from, ranging from no chat features, chats limited to only friends, or open chats, whereby anyone can chat. In addition, *PS5* provides a feature that blocks the content created by other users. The daily screen time, monthly spending limit, and web browsing can be restricted by the parents. After the time or spending limit has been reached, parents receive notifications via their registered email.

Xbox is a gaming console developed by Microsoft [90]. Like *PS5*, *Xbox* provides a wide variety of games for children and adults. An account is required to play games at a minimum age of 18 years. Younger children can link their accounts to their parents to play games in *Xbox*. A wide range of chat features is available, such as chats with friends, group chat rooms, chats with unknown people while playing games, voice messaging, and video chats. Content-sharing features are also available. *Xbox* provides a Family Setting application to manage parental controls, which can be installed on the mobile phone. This application is available in the Apple and Google Play stores. Thank to this application, the parents can set up accounts for their children. Several options are available, such as

screen time limits, content filtering, and spending limits. Parents can approve or block any user who sends a friend request. Contacts, chatting with friends, group chats or chats with unknown people, and any item purchased can be restricted by parents. Moreover, *Xbox* generates a weekly or monthly report for all activities performed by the children, which is sent to the parents. *Xbox* also provides an opportunity to locate family members through the Family Setting application.

Nintendo provides a series of switches with a variety of games for all age groups [91]. The most appealing feature of the *Nintendo* Switch is the ease of carrying the device that can be used in travel because of its light weight. Switches can also be connected to TVs or computers to enjoy playing on large screens. The minimum age required to create an account for this platform is 13 years. However, parents can link their accounts to their children's younger than 13 years. *Nintendo* Switch provides multiple chat features including chats with friends, chats with unknown people in games, group chat rooms, and voice chats. *Nintendo* provides a *Nintendo* Switch Parental Controls mobile application for parents to restrict and monitor the activities of linked accounts. Many parental control options are available, such as setting screen time limits, content filtering, and spending limits for the *Nintendo* store. Chats with friends, group chats, chats with unknown people, item purchases, Internet browsing, and friends' request approval can all be restricted by parents.

Industry's Protection Mechanisms

The industry has also initiated actions to improve existing child protection mechanisms. For instance, Millicom, a leading provider of cable and mobile services dedicated to emerging markets in Latin America and Africa, partnered with UNICEF to map out the risks and

opportunities faced by the telecommunications sector with respect to children's rights [92]. The partnership aimed to develop guidelines and tools for telecommunications companies to assess how their policies and processes might affect children's rights. Microsoft's *photoDNA* is a software devoted to tracking child sexual content by assigning a DNA to each photo. Since its launch, billions of photos have been analyzed [93]. Microsoft also developed a program to support national governments in establishing initiatives and action plans for the COP [94]. Safaricom has built upon children's rights in business principles by developing their own children's rights and business policy [95]. They highlight the importance of respecting children's rights and introducing business cases to do so. In partnership with Chicos.net, Disney's Amigos Conectados Project offers teachers, parents, and children in Latin America the digital literacy and citizenship skills necessary to fully engage in the digital future [96]. Thorn, a non-profit entity that drives technological innovation to fight the sexual exploitation of children, developed a solution to help companies identify tools and practices that can help prevent their platforms from being used for child sexual exploitation [97]. LEGO collaborated with a key supplier in India to develop and implement training on child rights as a part of the LEGO Academy [98]. Lego's supplier guidelines help suppliers, including those providing digital marketing or product development services, to apply these guidelines to everyday business operations.

The gaming industry is looking into finding appropriate ways to control different kinds of risks, as discussed above, by improving parental control features and content filtering tools. However, these tools are insufficient to rectify multifaceted threats. In terms of content filtering, gaming platforms cannot completely filter inappropriate content. Available tools are limited in providing safety for gamers and result in disabling chat features, which are essential for online multiplayer games. Meanwhile, there is no enforcement

from a regularity body to provide safety to children on the gaming platforms and invoke regulations to be followed by the industry.

Lately, 48% of the parents have used parental control features in the US [13] and 46% in the UK [14] to control the exposure of a child to inappropriate content, which indicates that more than half of the population in the US and UK are not even aware of the usage of parental controls. Therefore, more than 50% of the children's population in the US and the UK is vulnerable to a multitude of risks. Although international and national governments, policymakers, and the gaming industry are trying to create a safe environment for children, the threats related to online child victimization cases are increasing day by day, which shows the lack of research and policies that ensure the safety of a child online and specifically, in the gaming ecosystem. More advanced and efficient solutions are needed to enable children to play online while being safeguarded against predators' threats without sacrificing the ability to communicate with other players.

3. Related Work

In this Chapter, the theoretical concepts used in this project are described. Section 3.1 provides a detailed background of NLP. The NLP pipeline and its components for example data acquisition methods, data cleaning, preprocessing, feature engineering and modeling are described in the detail. Section 3.2 provides the detailed architecture of chatbots. The integration of the chatbot with the network is also included in section 3.2. This Chapter builds the knowledge required to understand the proposed solution *Protectbot*.

3.1 Natural Language Processing

NLP is a branch of computer science that aims to make it possible for machines to comprehend language in the same "natural" way that people do [99]. Humans are interacting with machines for decades by writing programs to perform different tasks. However, these programs are written in languages such as C, C++, Python, and Java that machines can understand not in the natural language. On the other hand, NLP enables machines to understand natural human language and learn the way humans communicate with each other. This is more beneficial for humans as it provides ease of use. For example, the personal voice assistants found in the phones such as Alexa and Siri are a product of NLP.

These applications are designed in a way that they not only understand the human natural language but they can also act upon the commands given to them. Other examples of technologies based on NLP include translation of text in different languages, summarization, text generation, speech recognition, and text sentiment analysis.

The relationship of different fields with NLP is shown in Figure 3.1 [100]. AI is a broad field that is concerned with achieving human-like intelligence using machines. ML, Deep Learning (DL) and NLP are subfields of AI . However, ML, DL and NLP are overlapping fields. ML deals with improving computer algorithms through experience and data. DL is a subfield of ML that usually uses deep neural networks to perform any task. NLP uses ML and DL algorithms to deal with tasks related to natural language. But it is important to note that some methods used in NLP do not overlap with ML which includes some statistical methods such as counting words.

Some commonly used application areas of NLP are mentioned below:

- Automatic text summarization: Text summarization means the text is provided to the NLP model and it will provide the summary of the provided text.
- Translation: Translation of the text from one language to another is very useful in many contexts, for example, Google Translate is providing this feature for the past fifteen years.
- Sentiment Analysis: The process of identifying the emotions in a text is known as sentiment analysis also known as opinion mining. This tool is very useful in understanding customer reviews and social media posts.
- Information extraction: Some applications require the extraction of specific information to process the data. The process of extracting particular terms or information from textual data is known as information extraction.

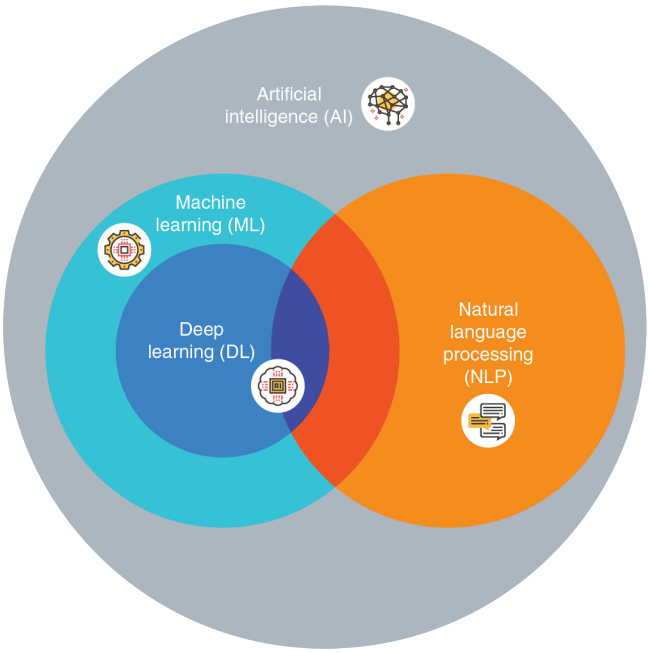

Figure 3.1: The relationship among different fields: AI, ML, DL, and NLP [100]

- Grammar checking: Grammar checking software are able to identify grammar, punctuation and spelling mistakes. These softwares are utilizing NLP to understand grammatical mistakes and correct them by making appropriate suggestions.
- Chatbot: Chatbots are designed to converse with humans via speech or text. Chatbots are able to generate human-like speech which is an excellent application of NLP.
- Personal voice assistants: Siri, Alexa, Cortana and Google Assistant leverage NLP techniques to understand and generate responses to the commands given to them.

In order to build an NLP application a step-by-step procedure is followed which is known as the NLP pipeline shown in Figure 3.2. These steps are commonly used in every NLP project, so it is important to understand each step before building any NLP project. The important stages in the pipeline are as follows [101]:

1. Data acquisition

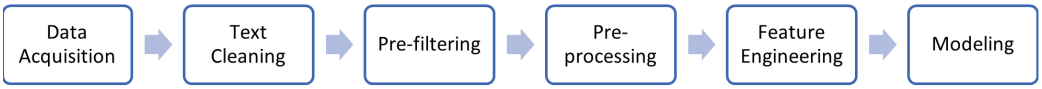

Figure 3.2: NLP pipeline

2. Text cleaning
3. Pre-filtering
4. Pre-processing
5. Feature engineering
6. Modeling

3.1.1 Data acquisition

The performance of an ML model is highly dependent on the data used to train and test the model. Data is the key component to building a high-performance ML model. For most industrial projects it is very important to train the model on datasets with thousands or even millions of data points. There are many public platforms that provide datasets to use for the research purpose such as Kaggle, UCI Machine Learning Repository, Google Dataset Search, Data.Gov, Datahub.io, Earth Data, CERN Open Data Portal, Global Health Observatory Data Repository, BFI film industry statistics, NYC Taxi Trip Data, FBI Crime Data Explorer and many more. If a public dataset is available relevant to the model then it could be utilized.

If the public datasets are not available then data is collected from different available resources for building an NLP or ML model. Different techniques are utilized during the

data collection process such as data scraping and industry intervention [101]. The data collection is an exhausting process. It can take anywhere between three to six months to collect a decent-sized, comprehensive dataset. Usually, data collection and cleaning can take more time than building the actual NLP/ ML model. Sometimes instead of collecting new data, data augmentation is utilized to increase the size of the dataset.

3.1.2 Text cleaning

Extraction of data from the raw data is an essential step before using the data to build any NLP model. The removal of non-textual information such as metadata and markups is important to get the original required data. Data cleaning may take several steps to get the cleaned processed data. For example, the data obtained from the website is usually in HTML format. Extracting the text from HTML requires several cleaning steps to generate the required format of data this is called HTML parsing. If the raw data is in PDF format it may require different steps to get the original text.

Another important cleaning step is converting the Unicode characters such as symbols, emojis and other graphical characters into a binary representation that is machine-readable. This process is known as text encoding. Ignoring the Unicode characters can cause several errors later in the pipeline.

Spelling correction of the dataset is another important text-cleaning step. The spelling mistake can hinder the linguistic understating of the data. It is important to correct the maximum number of possible words to maximize the performance of the classifier.

3.1.3 Pre-processing

After cleaning the data, the dataset contains plain text. All NLP algorithm typically works at the sentence level and expects a separation of words. So we need to split the text into words and sentences before proceeding further in the pipeline. In many applications converting the text into lowercase is another important requirement. Such kind of processing of text is done in the pre-processing step of the NLP pipeline. Some commonly used pre-processing techniques are tokenization, stop word removal, stemming, lemmatization, removal of digits and punctuation, lowercasing, POS tagging etc. Let's describe the pre-processing techniques mentioned above.

Tokenization

As mentioned earlier most of the NLP algorithm works by splitting the text into words or sentences. The process of splitting the text is known as tokenization. Depending upon the demand of the data and the results one wants to achieve, tokens can be of any length from sentences to phrases to words to sub-words.

Stop words removal

Stop words are commonly occurring words in the language that do not contribute to the meaning and information provided in a sentence such as is, am, are, in, of etc. They occur in large numbers in any document and can mislead a classifier in later steps of the NLP pipeline. So they are usually removed in pre-processing step.

Stemming

Stemming is the process of transforming a word into its stem word or base word. It is a rule-based and naïve process of converting any word into its stem word. Stemming removes the suffixes and reduces the word to base form for example after stemming the word buses and cars reduce to bus and car. The stemming can produce words that are not linguistically correct for example revolution is converted into *revolut* which is not a correct word and formality is converted into *formaliti* which is not the dictionary word.

Lemmatization

Lemmatization is the same as stemming but lemmatization always generates a dictionary word that is linguistically correct and it can reduce the words which were not reduced by stemming as the word *better* remains the same after stemming but lemmatization reduces better to good.

3.1.4 Feature Engineering

The machines can only understand numerical value so the pre-processed text needs to be converted into numerical value before feeding it into a classifier. Feature engineering refers to the set of methods that will convert the text into numerical vectors so the data can be fed into a classifier in later stages of the pipeline. Feature engineering is an important step for any machine learning algorithm, if the poor features are fed into a classifier the accuracy will be poor. Feature extraction is a commonly used method to get meaningful data from any format of data for example text, images, videos or speech. However, the representation of text into numerical vectors is usually more complex than the representation of images, videos and speech. There are many approaches available to extract useful features from the

text.

Vector Space Model

The text units such as characters, words, phrases, sentences, paragraphs and documents, must be converted into numerical vectors this process is known as Vector Space Model (VSM). It is a simple algebraic model used for representing any text in numerical vectors. VSM is fundamental to many information-retrieval operations, from scoring documents on a query to document classification and document clustering. These vectors are identifiers such as index numbers in a corpus vocabulary. In this setting, the most common way to calculate the similarity between two phrases or words or paragraphs is using cosine similarity. The cosine of the angle between their corresponding vectors. Given two vectors, A and B, each with n components, the similarity between them is computed as follows [101]:

$$similarity = cos(\theta) = \frac{A.B}{||A||^2||B||^2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.1}$$

where A_i and B_i are the ith components of vector A and B respectively.

Some of the commonly used techniques are described below.

Bag of Words

Bag of words (BoW) is a classical text representation technique that has been used commonly in NLP, especially in text classification problems. In this method, the text is represented as a bag of words without considering the order and context of a sentence. It is assumed that the text belonging to a given class in the dataset is characterized by a unique

set of bags. If two text pieces have nearly the same words, then they belong to the same bag. Thus, by analyzing the words present in a piece of text, one can identify the bag it belongs to. Consider the following example to understand the BoW technique.

Sentence 1: Harry loves to play football.

Sentence 2: He also likes to cook.

Sentence 3: He is a good guy.

Based on these sentences, the BoW list is as follows:

BoW_list= “Harry”, “loves”, “to ”, “play”, “football”, “He”, “also”, “likes”, “cook”, “is”, “a”, “good”, “guy”

Using this BoW, the representation of each sentence will be as follows:

Sentence 1: [1,1,1,1,1,0,0,0,0,0,0,0,0]

Sentence 2: [0,0,1,0,0,1,1,1,1,0,0,0,0]

Sentence 3: [0,0,0,0,0,1,0,0,0,1,1,1,1]

The text is converted into vectors that could be fed to any ML/NLP algorithm. The main advantage of this algorithm is its simple to understand and implement but the size of the vector increase as the vocabulary or dataset increases. For a large dataset, this algorithm is computationally expensive. Another drawback of this method is it can not tackle the out of vocabulary words. This method also does not care about the order of the word and the context of the word in a sentence.

One-Hot Encoding

One-Hot Encoding is very similar to the BoW. In one-hot encoding, each word w in a document is represented as a V-dimensional vector containing 0’s and 1’s. In a document wherever the word appears it is assigned 1 for it and all other words in that document are

assigned 0. Like BoW the length of the vector is proportional to the size of the vocabulary. Hence it is not suitable for large datasets. This method does not capture the context of the word and the similarity between different words.

Term Frequency-Inverse Document Frequency

In the above-mentioned methods, all the words are given equal importance which does not provide good features to feed in an ML classifier. To resolve this issue *Term frequency-inverse document frequency (TF-IDF)* was introduced. It aims to quantify the importance of a given word relative to other words in the document and in the dataset. It is a commonly used representation method for information-retrieval systems, for extracting relevant documents from a corpus for a given text query.

The intuition behind TF-IDF is that if a word w appears many times in a document d_i but does not occur much in the rest of the documents d_j in the corpus, then the word w must be of great importance to the document d_i. The importance of w should increase in proportion to its frequency in d_i, but at the same time, its importance should decrease in proportion to the word's frequency in other documents d_j in the corpus. Mathematically, this is captured using two quantities Term Frequency (TF) and Inverse Document Frequency (IDF).

TF measures how often a term or word occurs in a given document. Since different documents in the corpus may be of different lengths, a term may occur more often in a longer document as compared to a shorter document. To normalize these counts, we divide the number of occurrences by the length of the document. TF is defined by the following equation [101]:

$$TF(t,d) = \frac{\text{Number of occurrences of term t in a document d}}{\text{Total number of terms in the document d}} \tag{3.2}$$

IDF measures the importance of the term across a corpus. In calculating TF all terms are given equal importance. However, it is important to note that stop words like is, on, am, at etc., are not important but they occur frequently. To take account of such cases, IDF weighs down the terms that are very common across a corpus and weighs up the rare terms. IDF is defined by the following equation [101]:

$$IDF(t) = \frac{\text{Total number of documents in the corpus}}{\text{Number of documents with term t in them}} \tag{3.3}$$

The TF-IDF is calculated by the following expression:

$$TF - IDF = TF(t,d).IDF(t) \tag{3.4}$$

Like BoW and one-hot encoding, TF-IDF is also not able to determine the context of words and the similarity between different words. With the increase in the size of the dataset, the computational complexity increases. Like BoW and one-hot encoding, TF-IDF cannot handle any out of vocabulary word.

Word Embedding

Natural language processing prepares textual data for machine learning models [99]. The efficiency of the ML or NLP model is highly dependent on how the textual data is mapped from text to numerical values. Word embeddings are the representation of text in the form of real-value vectors. Word embeddings not only map the text data into numerical vectors but also capture the context and semantics of the words thus creating a relationship between

different words. Words having a similar meaning map to similar vectors and have similar representations. This helps the machine learning models to learn the meaning and context of different words. Word embedding maps the individual words to numerical vectors; so tokenization should be performed on the corpus before word embedding. Figure 3.3 shows the mapping the word embedding creates between similar context words.

Word embeddings use a variety of techniques to create a numerical representation and are the most popular way to represent a document's vocabulary. The main advantages of word embeddings are they capture the contextual, semantic and syntactic similarities and the relationships between different words which aid the ML model to learn natural language. Consider an online shopping website that wants to analyze the good and bad reviews of the products. It will be much harder to see each comment given by the users. Most of companies are using ML based applications to categorize the customer's feedback. Customer feedback consists of different adjectives describing the product, its usefulness and the missing features they want to see in any particular product. Suppose a company wants to categorize the feedback into two categories positive and negative. There are many adjectives that are classified as positive and many are classified as negative. So in this case ML/NLP model should be able to differentiate the positive and negative words and must have knowledge of different adjectives that could have similar meanings either positive or negative. Word embeddings are very useful in these scenarios. Consider these two reviews by two different customers:

Review 1: This product is great.

Review 2: This product is awesome.

The ML/NLP model needs to understand that great and awesome are positive words and have similar meanings. Thus similar meaning words should be clustered together.

There are different models to generate word embeddings. The most commonly used models are word2vec, Glove and Fasttext.

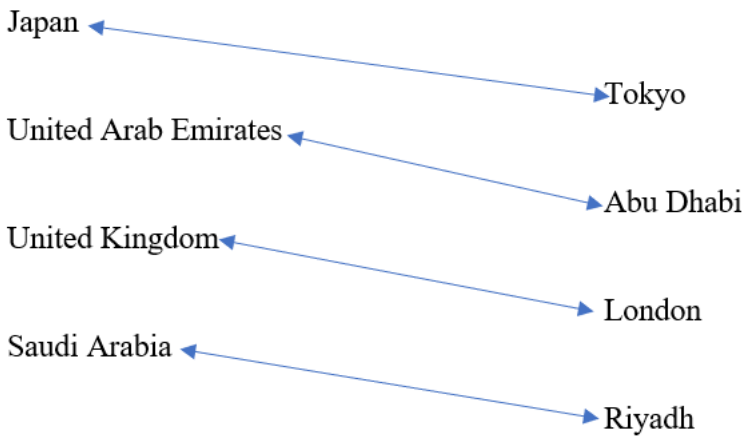

Figure 3.3: Examples of word embeddings

Word2Vec

Word2Vec is a two-layer Neural Network (NN). The input to the word2vec NN is text corpus and the NN will generate the numerical vectors as the output. These vectors are known as feature vectors for the input corpus. The basic idea of word2vec is transforming text data into the numerical vectors required to feed into an ML model. The aim of the word2vec is to understand the probability of words occurring together and to form clusters of word vectors that have a similar meaning. Like any other deep learning model, the efficiency of the model is highly dependent on the amount of data used to train the model. Hence if the model is trained on a large dataset it could generate a good understanding of natural language similar to how children understand language by listening to adults. Once the word2vec is trained it produces a vocabulary containing vectors for each word in the training corpus. It will also form connections between words and the vectors of similar

words will lie close together in the form of clusters to ensure the machine that these words have a similar meaning. For example, the vectors associated with words *boy* and *man* will be close to each other.

Word2vec trains a word against its neighboring words in the input textual data; there are two methods to do this training: Continuous bag of words (CBOW)and skip-gram.

- Continuous Bag of Words

  CBOW, as shown in Figure 3.5, is a language model that predicts the center *word* given the context words around that *word*. The language model is the statistical model that tries to give a probability distribution over sequences of words. For example, if a given sentence contains *w* words the language model will assign a probability to the whole sentence. The language model will always assign a higher probability to a *good* sentence and a lower probability to a *bad* sentence. The definition of *good* sentences to a language model are sentences that are semantically and syntactically correct. The sentences that are semantically or syntactically incorrect are given lower probabilities and are considered as *bad* words. For example, *The cat is eating food* will be given a higher probability than the sentence *the cat is the food*.

  CBOW learns a language model that tries to predict the center word from the words in its context. Let's consider a sentence to understand this model:

  I want eggs and a strawberry milkshake for my breakfast

  If we consider the milkshake as a center word and the context window size is 2 then the words a, strawberry, for, and my are context words. As mentioned earlier the word2vec models are neural networks so we need input-output pair to train any neural network. For the above-mentioned sentences these pairs are (a, milkshake),

(strawberry, milkshake), (for, milkshake), (my, milkshake). The proximity of the context words within the context window does not play any role; all the words in the proximity window are treated in the same manner while training. CBOW tries to do the above-explained procedure to every word in the corpus; it will consider each word of the corpus a target word and take input words from the context words and create input-output pairs to train the CBOW neural network. To run the above-mentioned procedure on a training corpus, a sliding window of size 2k+1 is run over the text corpus. For context window size 2, k will be assigned values 2. So the window size is $2*2+1=5$. The center word is the target word and k words on each side of the central word are context words. Total 2k input-output pairs are generated. To get the next data point the window is slid by one word each time as shown in Figure 3.4. The target central word is indicated by a red color and the context words are indicated by a box. Now that the training input-output pairs are ready, let's see the working of the CBOW shallow neural network. CBOW model has only one hidden layer as shown in Figure 3.5. Suppose the D-dimensions word embedding is used to train the model and V is the vocabulary of the trainset of a text corpus. The objective of this model is to learn the word embedding matrix E. Initially, the d-dimensional matrix is initialized with some random values. In the input layer shown in Figure 3.5, indices of the words in context are used to retrieve the corresponding rows from the embedding matrix E. The vectors retrieved are then added to get a single D-dim vector and passed to the next layer. The next layer simply takes this d vector and multiplies it with another matrix E'. This gives a $1x|V|$ vector, which is fed to a softmax function to get probability distribution over the vocabulary space. This distribution is compared with the label and uses back

propagation to update both the matrices E and E' accordingly. At the end of the training, E is the embedding matrix learned by CBOW.

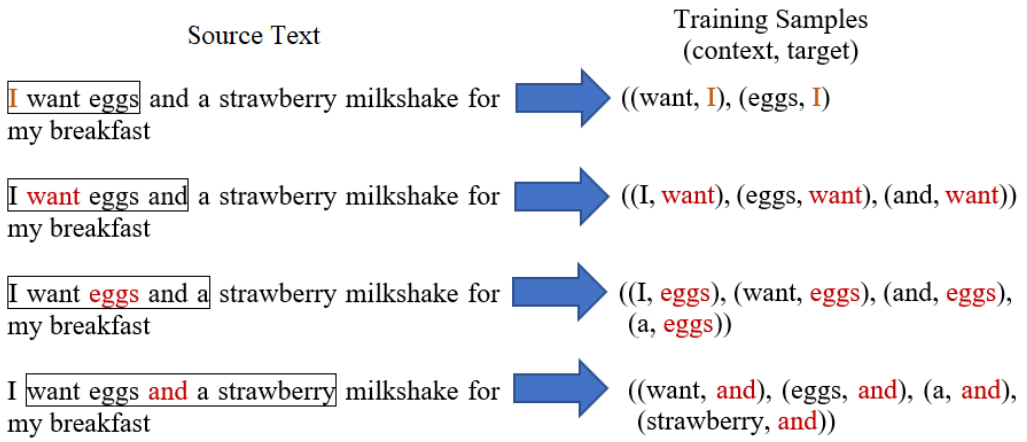

Figure 3.4: CBOW example

- SkipGram

  The SkipGram model is very similar to the CBOW with some structural changes. In SkipGram the model tries to predict the context words using the center word. For the above-mentioned example with context window size 2, using the center word *milkshake*, the algorithm tries to predict all the context words—" a", "strawberry", "for" and "my". This is skipGram algorithm running for one word. This process will repeat for every word in the corpus as shown in Figure 3.6.

  To train the skipGram the dataset is prepared by sliding a context window of size $2k+1$ over the whole corpus to get the set of $2k+1$ words. Unlike CBOW, this gives us 2k data points. A single data point consists of a pair (index of the center word, index of a target word). We then shift the window to the right on the corpus by one word and repeat the process. This way, we slide the window across the entire corpus to create the training set. This is shown in Figure 3.6. The training

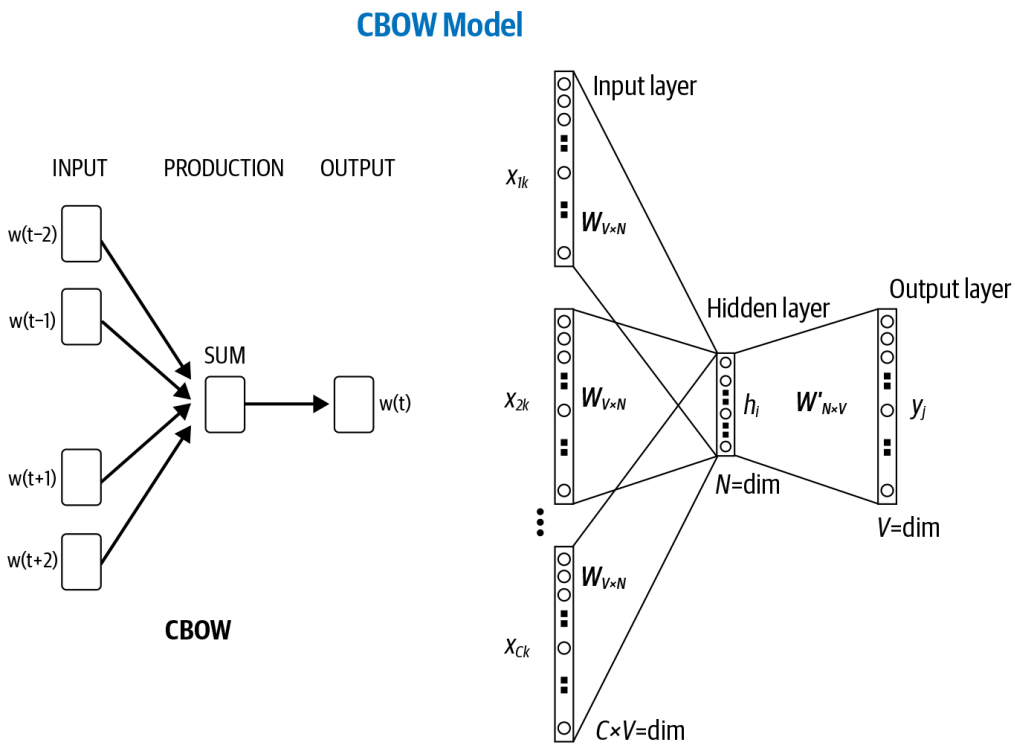

Figure 3.5: CBOW [101]

network for the SkipGram is shown in Figure 3.7, is very similar to the network used for CBOW with some changes for example in the input layer, the index of the word in the target is used to retrieve the corresponding row from the embedding matrix E. The retrieved vectors are then passed to the next hidden layer. The next layer takes this d vector and multiplies it with another matrix E'. This creates a $1x|V|$ vector, which is then fed to a softmax function to get probability distribution over the vocabulary space. This distribution is compared with the label and using back propagation update both the matrices E and E' accordingly. At the end of the training, E is the embedding matrix we wanted to learn.

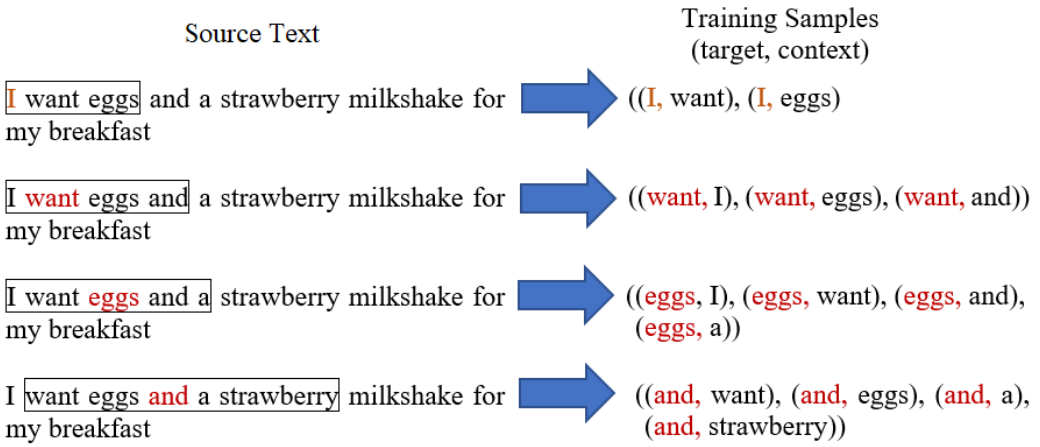

Figure 3.6: Skipgram example

fastText

fastText [102] is an open-source library developed by Facebook AI Research (FAIR) for learning word embeddings and word classifications. This model can generate numerical vectors for natural language words. FastText supports both CBOW and SkipGram models for training purposes. This model is based on the skipGram model where each word is represented as a bag of characters n-grams. Unlike skipGram, in fastText, the numerical vector is assigned to each character n-gram instead of a word and a word is represented as the sum of these numerical vectors assigned to the character n-gram. This model learns the word representation while taking into account morphology. By considering the subwords units, the word is represented by a sum of its character n-gram.

To understand the fastText model it is important to understand the mathematical background of skipGram. Given a word vocabulary of size W where a word is represented by its index $w \in 1, 2, \ldots, W$, the goal of fastText model is to learn a vectorial representation for each word w. Given a large training corpus represented as a sequence of words

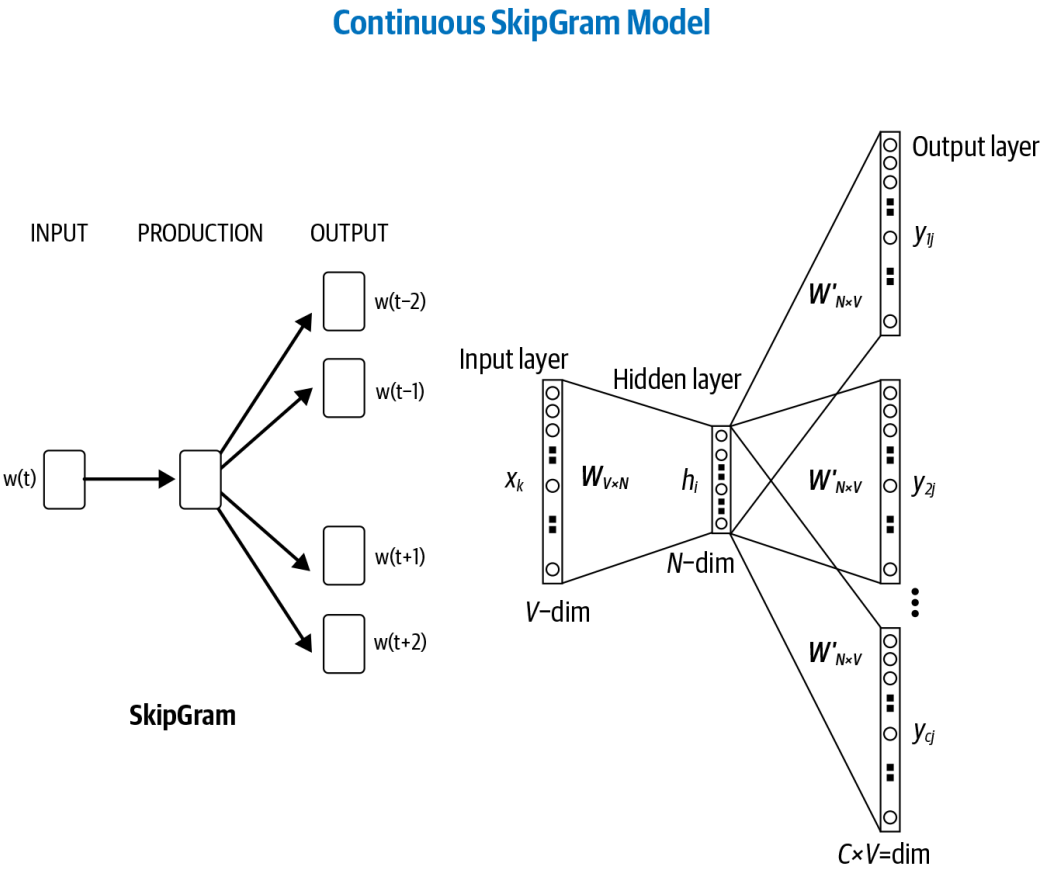

Figure 3.7: SkipGram [101]

$w_1, w_2, \ldots, w_T$, the objective of the skipgram model is to maximize the following log-likelihood:

$$\sum_{t=1}^{T} \sum_{c \in C_t} log p(w_c | w_t) \tag{3.5}$$

The prediction of context words can be mathematically presented as a set of independent binary classification tasks. The aim of classification is to predict the presence or absence of context words. Let's take an example of the word at index t, all the context

words are considered as positive examples and randomly chosen words from the dictionary not present in context words are considered negatives. For a chosen context position c, using the binary logistics loss, the negative log-likelihood can be obtained using the following equation [102]:

$$log(1+e^{-s(w_t,w_c)}) + \sum_{n \in N_{t,c}} log(1+e^{s(w_t,n)}) \tag{3.6}$$

where $N_{t,c}$ is a set of negative examples sampled from the vocabulary. By denoting the logistic loss function $l : x \rightarrow log(1+e^{-x})$, the objective can be rewritten as :

$$\sum_{t=1}^{T} [\sum_{c \in C_t} l(s(w_t,w_c)) + \sum_{n \in N_{t,c}} l(-s(w_t,n))] \tag{3.7}$$

The above-described model is the skipGram model with negative sampling. The skipGram model ignores the internal structure of the words and represents them by distinct vector representation for each word. In fastText, each word w is represented as a bag of character n-gram. A special boundary symbol $<$ and $>$ is added at the beginning and end of the word respectively, to distinguish prefixes and suffixes from other character sequences. The word w itself is also included in the set of n-grams. To explain this let us take an example of word *train* and $n = 2$, the representation of this word is shown below:

<train> = <t, tr, ra, ai, in, n> and <train>

In practice, all the n-grams for n greater or equal to 2 and smaller or equal to 6 are extracted as shown below:

<train> = For n=2 : <t, tr, ra, ai, in, n>, For n=3: <tr, tra, rai, ain, in>, For n=4: <tra, trai, rain, ain>, For n=5: <trai, train, rain>, Fro n=6: <train, train>, and the full word <train>.

Consider a corpus of size G with n-gram of characters. Given a word w, let us denote the set of n-gram appearing in w by G_w subset $1,\ldots,G$. The vector representation z_g is assigned to each n-gram g. A word is represented by the sum of the vector representation of its n-grams. Thus the scoring function s can be calculated as:

$$s(w,c) = \sum_{g \in G_w} z_g^T v_c \tag{3.8}$$

This model enables sharing the representations across words, thus allowing to learn the representation of out-of-vocabulary (OOV) words that are not part of the training corpus.

Three methods of word embeddings are discussed in this section word2vec, Glove and fastText. The word2vec and Glove are trained on words while fastText is trained on character n-grams of words. The word2vec and Glove are unable to generate the numerical vectors of the words that are not used in the training of the models. However, the fastText generates n-grams characters of words to generate numerical vectors of out-of-vocabulary words. Therefore this method provides better performance as compared to word2vec and Glove [103] The author in [103] shows the performance comparison of word2vec, Glove and fastText on the public dataset of news stories from UCI KDD Archie [104]. The results show that the fastText shows the best results as compared to the other word embedding methods. Another research [105] shows the performance comparison of the above mentioned three embedding techniques on publicly available two datasets. The results showed that fastText performs better than word2vec and Glove.

3.1.5 Modeling

Classification is the task of mapping the input to a discrete output. For example, given some information about the online shopping reviews of different reviewers; the classifier

should be able to classify the review into two categories, positive reviews or negative reviews. As mentioned in section 3.1.4 the classification models can only understand the numerical vectors. Once the numerical vectors are extracted from the dataset, the data is ready to be fed into a classifier. Several Classifiers are used for text classification purposes; some of the commonly used classification methods are discussed below:

K-Nearest Neighbour

The KNN algorithm is a non-parametric, supervised learning classifier, which uses proximity to make predictions about the grouping of an individual data point [106]. The main idea used in this algorithm is that similar class elements exist near to each other. KNN can be used for regression or classification problems but typically it is used for classification problems. For classification problems, the neighboring data point around a point c is considered, the distance of each neighboring point is calculated and the class label is assigned to c on the basis of a majority vote. This voting is also known as plurality voting. Regression problems use a similar idea as classification problems but in this case, the average of the k nearest neighbors is taken to make a prediction about a classification.

The value of k defines how many neighbors need to be considered to determine the classification of a query point c. For example, k=3 is the default value used by the algorithm, in this case, three neighboring points are considered to determine the class of a query point. The suitable value of k depends on the dataset used for the classification. If the dataset have more outliers or noise the larger values of k are more suitable than smaller values. The different values of k can lead to overfitting or underfitting. Lower values of k have high variance but low bias and higher values of k have low variance but high bias. If the data is clean and the same class data points are near to each other then the lower value of

k can serve the purpose. It is recommended to have an odd number for k to avoid ties in classification, and cross-validation can help to choose the optimal k for a particular dataset.

In order to determine which class data points are more close to the data point c, the distance between the neighboring points and point c is calculated before the classification can be made. These distance metrics help to form decision boundaries, which partition query points into different regions. Voronoi diagrams are most commonly used to visualize decision boundaries. While there are many distance metrics used to calculate the distance most commonly used distance metrics are mentioned below:

Euclidean distance is the most commonly used distance measure used only for real-valued feature vectors. It can be calculated using the formula mentioned below [107]:

$$d(c,x) = \sqrt{\sum_{i=1}^{n} (x-c)^2} \tag{3.9}$$

here x is the neighboring point and c is the query point.

Manhattan distance is also another popular distance metric, which measures the absolute value between two points. This is the shortest distance between two points which can be calculated by the following formula [107]:

$$d(c,x) = \sum_{i=1}^{n} |x-c| \tag{3.10}$$

Minkowski distance is the generalized form of Euclidean and Manhattan distance metrics shown in the below equation [107]:

$$d(c,x) = \sum_{i=1}^{n} |x-c|^{1/p} \tag{3.11}$$

The parameter, p allows for the creation of other distance metrics. Euclidean distance

is represented by this formula when p is equal to two, and Manhattan distance is denoted with p equal to one.

KNN is a lazy learning classification model as it only stores the training dataset no other processing step is applied. This also means that all the computation will occur at the time of model testing. KNN is known as a memory-based or instance-based learning method as it relies on memory to store all its training data. As a dataset grows, KNN becomes increasingly inefficient, compromising overall model performance. It is commonly used for simple recommendation systems, pattern recognition and data mining. The advantages of this algorithm include:

- This is the simplest algorithm to implement and understand but still, it can provide good results for classification tasks.
- All the training data is saved in the memory thus new training samples could be added at any stage seamlessly. The algorithm adjusts to account for any new data since all training data is stored in memory.
- This algorithm has very less hyperparameters. To use KNN the number of neighbors k and distance metric p needs to be defined.

On the other hand, the KNN also has the following limitations:

- As mentioned above the KNN is a lazy algorithm, it takes up more run time memory and data storage compared to other classifiers. Hence this algorithm is more costly in terms of memory usage and time.
- KNN does not perform well if the dataset contains data points of high- dimensionality.
- KNN is also more prone to overfitting. Lower values of k have high variance but low bias and higher values of k have low variance but high bias. If the data is not

clean and have outliners the performance would be highly affected and hence this classifier will not be able to generate good results.

- This algorithm is sensitive to missing or noisy data.

Naive Bayes

A Naive Bayes classifier is a probabilistic machine learning model based on the Bayes theorem. To understand Naive Bayes (NB) first we need to understand the Bayes theorem which is as follows:

$$P(A/B) = \frac{P(B/A).P(A)}{P(B)} \tag{3.12}$$

Where A and B are events and $P(B) \neq 0$.

This equation shows that the probability of occurrence of event A given that B event has occurred can be computed by the probability of occurrence of B given that A event has occurred. The basic intuition of Bayes theorem is that the occurrence of two events A and B is independent of each other.

The same idea is used in NB classifier that the presence of one particular feature does not affect the other features. The algorithm can be explained by using the following example.

The dataset contains different parameters to define the weather conditions of a day and based on the weather conditions a player can decide whether he/she should play golf on that or not as shown in Figure 3.8. The parameters defining the weather conditions outlook, temperature, humidity and windy are features that are used as input vectors and the response a player should play golf (yes/No) is the output of the classifier.

The basic assumption of NB is that features are independent of each other and contribute equally to the output of the classifier. For the above-mentioned dataset this assump-

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Figure 3.8: Dataset used to understand Naive Bayes [100]

tion can be understood as:

The humidity of a day does not depend upon the temperature of the day or the outlook of the day. So each feature can be treated independently. Each feature equally contributes to the forecast of a day to decide if it is suitable to play golf or not.

Now let's apply the Bayes theorem to the above mentioned dataset. Let X be the

feature vector containing all the parameters to define the weather of a day and y be the binary output of the classifier. The Bayes theorem can be rewritten as;

$$P(y/X) = \frac{P(X/y).P(y)}{P(X)} \tag{3.13}$$

Here $X = (x_1, x_2, x_3, x_4)$ and $y = (1, 0)$.

So if the input is given to the classifier the output will be generated based on the values given in the training dataset. For example, if the input features given to the classifier are "outlook= overcast", "temperature= hot", "humidity = high" and "windy = False" then the classifier should generate the output "Yes". Now as we assume that events are independent of each other so it can be written as:

$$P(y \cap X) = P(y).P(X) \tag{3.14}$$

Now eq 3.13 can be written as;

$$P(y/x_1, x_2, x_3, x_4) = \frac{P(x_1/y)P(x_2/y)P(x_3/y)P(x_4/y).P(y)}{P(x_1)P(x_2)P(x_3)P(x_4)} \tag{3.15}$$

$$P(y/x_1, x_2, x_3, x_4) = \frac{P(y).\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)P(x_3)P(x_4)} \tag{3.16}$$

Here $P(y) and P(x_1) \ldots P(x_4)$ are constants so

$$P(y/x_1, x_2, x_3, x_4) \propto \prod_{i=1}^{n} P(x_i|y) \tag{3.17}$$

Now, we need to create a classifier model. For this, we find the probability of the given set of inputs for all possible values of the class variable y and pick up the output with maximum probability. This can be expressed mathematically as:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.18}$$

There are many types of NB classifiers as described below:

Multinomial Naive Bayes

In Multinomial Naive Bayes (MNB) the feature vector is represented using the multinomial distribution. This is mostly used for document classification problems for example if we want to know the category of a particular document among different categories like sports, politics etc.

Bernoulli Naive Bayes

This is similar to MNB but the difference is that the features are represented by the binary value of 0 or 1. For example, if we want to check the presence of a word in the document, then we can utilize Bernoulli NB.

Gaussian Naive Bayes

In Gaussian Naive Bayes, continuous values associated with each feature are distributed according to a Gaussian distribution or normal distribution. Gaussian distribution is a symmetrical bell-shaped curve about the mean of the feature values as shown in Figure 3.9. The probability of occurrence of the features is assumed to be Gaussian, hence, conditional probability is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{3.19}$$

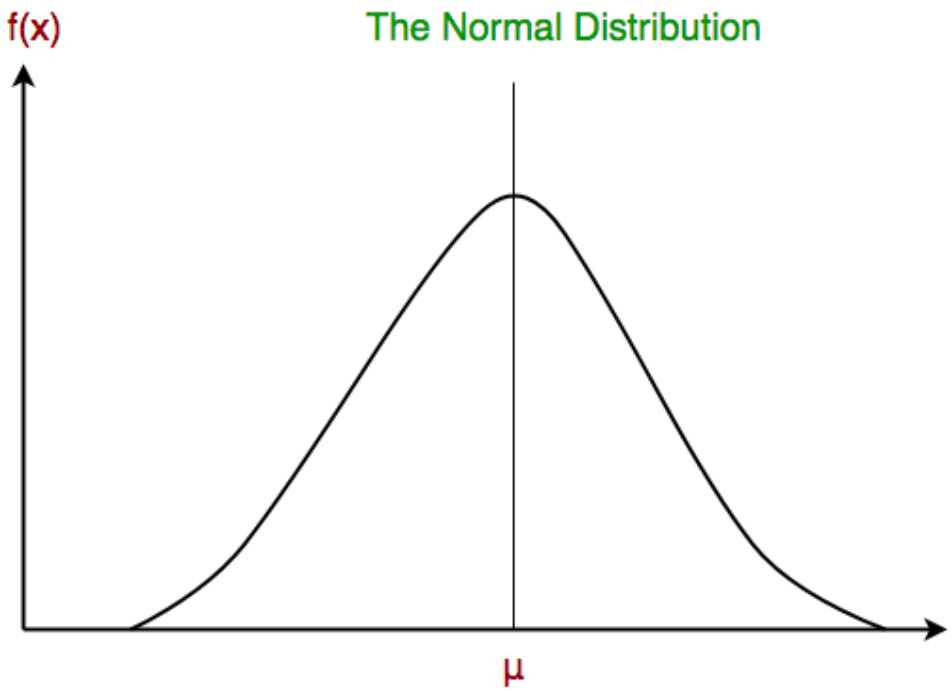

Figure 3.9: Gaussian Naive Bayes [108]

Support Vector Machine

SVM is a supervised machine learning algorithm model used for classification and regression. The basic idea of this algorithm is to find out the hyperplane in an N-dimensional space that separates the data points belonging to different classes in the best possible manner. The dimension of the hyperplane depends upon the number of features of the dataset. If two features are used then a hyperplane is just a line. If the used features are three then the hyperplane would be a 2-D plane. If the number of features increases the dimension also increases hence the complexity of the algorithm also increases.

To understand the idea of how SVM selects the best hyperplane separating the two

classes, consider the example shown in Figure 3.10. Consider red and blue dots representing the different class data points. All the hyperplanes shown in Figure 3.10 are separating the class data points perfectly but which one is the best way to separate the two classes?

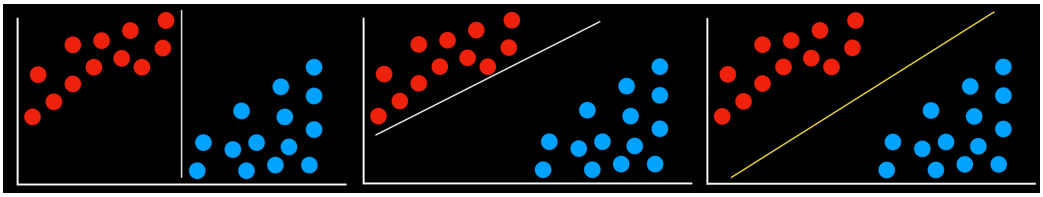

Figure 3.10: SVM hyperplanes [109]

The boundaries shown in the two leftmost decision boundaries are very close to some of the data points. Hence if the new data point in the test dataset is very close to these points they might be classified as other class data points. The right-most boundary is the best hyperplane to separate these two classes as the decision boundary has the maximum distance from each of the group data points. This type of boundary is known as the Maximum Margin Separator. In most cases, it is impossible to separate all the data points of different classes. In that case, the goal is to find the best hyperplane separating two classes of data points without overfitting on data.

$$c(x,y,f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \tag{3.20}$$

The hinge loss function (eq.3.18) is used to maximize the margin. The cost is 0 if the predicted value and actual values have the same sign. If they have a different sign then the loss function is evaluated, A regularization parameter is also added to the cost function to balance the margin maximization and loss. After adding the regularization parameter, the cost function looks as follows:

$$min_w \lambda ||w||^2 + \sum_{i=1}^{n} (1 - y_i < x_i, w >)_+ \tag{3.21}$$

To find the gradient, partial derivatives w.r.t weights are evaluated. Using the gradients, weights are updated.

$$\frac{\delta}{\delta w_k} \lambda ||w||^2 = 2\lambda w_k \tag{3.22}$$

$$\frac{\delta}{\delta w_k}(1 - y_i < x_i, w >) = \begin{cases} 0, & \text{if } y_i < x_i, w >) \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \tag{3.23}$$

When the model correctly predicts the class of the query data point, the gradient from the regularization parameter is updated.

$$w = w - \alpha.(2\lambda w) \tag{3.24}$$

If the model is unable to classify the query points then the loss function along with the regularization parameter needs to be considered to update the gradient.

$$w = w + \alpha.(y_i.x_i - 2\lambda w) \tag{3.25}$$

For SVM, there are many hyperparameters that can be tuned to get better performance. The most important parameters are “kernel”, “gamma” and “C”.

- Kernel: For SVM there are several kernel options that could be utilized e.g. linear kernel is most commonly used to generate a linear hyperplane such as a line. The rbf and poly kernel are non-linear and generate non-linear hyperplanes such as circles or curves as shown in Figure 3.11.
- Gamma: Gamma value can increase the sensitivity of the training data set used for

training the model. Higher the gamma the model will try to close fit on the training data set hence increasing the chance of overfitting. If higher gamma values are used the model needs to be tested for overfitting. An example is shown in fig 3.12.

- C: C is the penalty parameter of the error term. It also controls the trade-off between smooth decision boundaries and fitting the model on the training points precisely as shown in Figure 3.13.

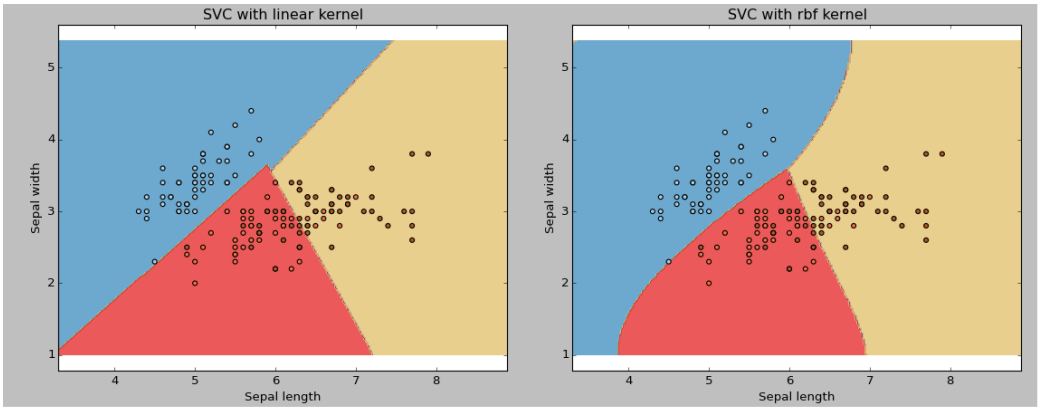

Figure 3.11: SVM using linear and rbf kernel [110]

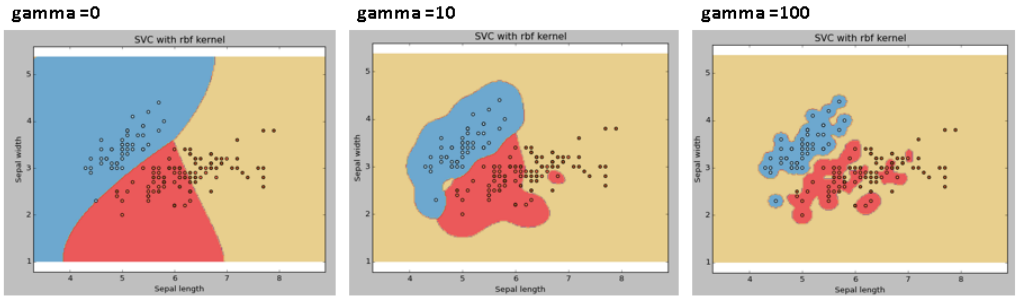

Figure 3.12: SVM using different values of gamma [110]

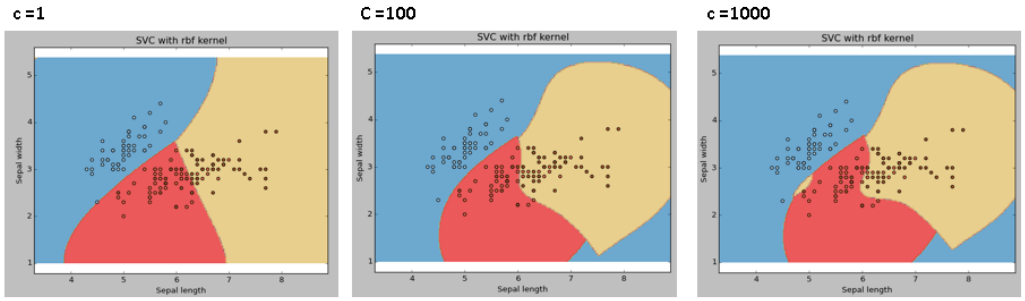

Figure 3.13: SVM using different values of C [110]

Decision Tree

A Decision Tree (DT) is a supervised learning algorithm that can be used for both classification and regression tasks. It creates a hierarchical tree structure that contains roots, branches, internal nodes and leaf nodes as shown in Figure 3.14.

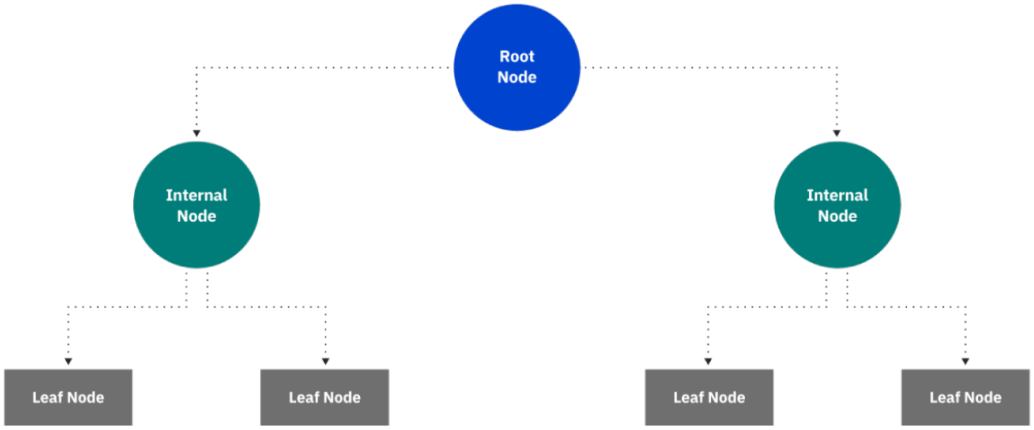

Figure 3.14: Decision Tree [111]

As shown in Figure 3.14, the algorithm starts with the root node which does not have

any incoming branches. The outgoing branches from the root node feed into the condition or internal node, based on which the tree splits into branches also known as edges. The end of a branch that does not split any further is known as a decision or leaf. For example, consider a situation in which we are trying to predict whether the conditions are favorable to do surfing or not as shown in Figure 3.15. The node "is there swell" is a root node. The node "wind" and "wind direction" are internal nodes and the nodes "surf" and "do not surf" are decision or lead nodes.

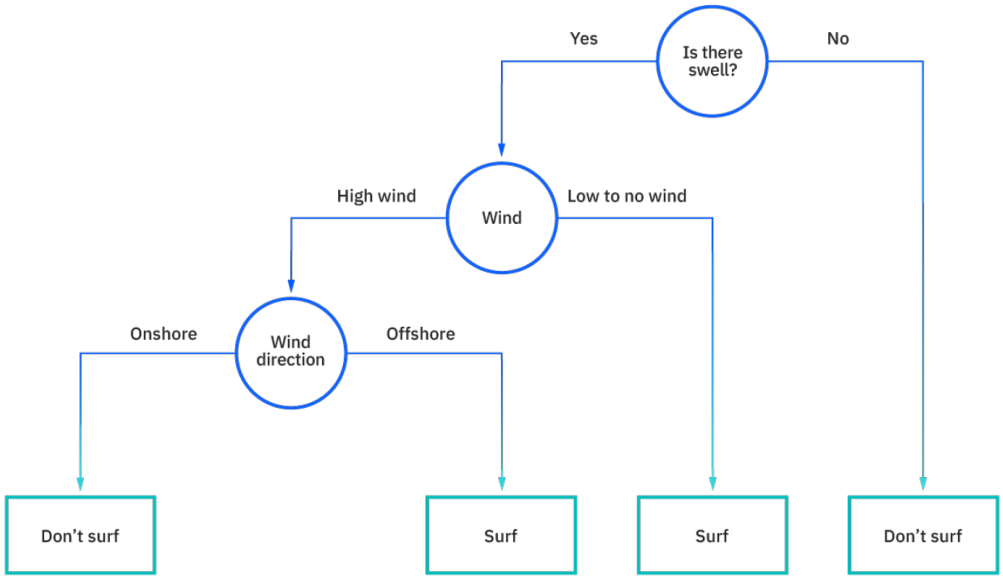

Figure 3.15: Decision Tree example [111]

For a simple dataset, the relationship between features can be viewed clearly. This methodology of learning a tree is known as a learning decision tree from data and the generated tree is known as a classification tree. In regression problems, the trees are generated in a similar fashion but they predict continuous values. But the real datasets are more complex and contain more features and the resultant tree could have hundreds of branches. Such trees are very complex and can lead to overfitting. One way to avoid

overfitting is to set a minimum number of training inputs to use on each leaf. Another way is to set the maximum depth of the model. Maximum depth will represent the longest path from a root node to the leaf.

Another way to increase the performance of DT is by using pruning [112]. Pruning is the way to remove the branches in DT that make use of features having low importance. There are two methods to implement pruning. One is reduced error pruning which works by removing each node with the most popular class in that leaf, if the accuracy remains the same after removing the node the change is retained. Another way to perform pruning is known as weakest link pruning where a learning parameter is used to weigh whether nodes can be removed based on the size of the sub-tree.

Some advantages of using DT are mentioned below:

- It is a simple algorithm to implement and understand,
- It can incorporate the numerical and categorical data having multiple outputs.
- Non-linear relationships between parameters do not affect the tree performance.

The limitations of DT includes:

- It is not suitable for complex datasets having a large number of features.
- A small variation in data can change the whole tree generated.
- This algorithm does not guarantee to the generation of the optimal trees.
- If the data is not balanced the DT does not perform well. In that case, it will generate a biased tree and end up classifying most of the data points in the class which is dominating in the training dataset.

Random Forest

Random Forest is a supervised ML technique used for classification and regression problems. It builds multiple decision trees on different samples of datasets and uses the majority vote to classify the query point. In the case of regression, the average is computed to decide the final output of the query data point. RF is an ensemble technique based on a bagging algorithm to combine the output of multiple classifiers to produce the final output. In bagging, different training subsets from training data with replacement are generated. This method of generating the subsets is also known as row sampling. Bagging is also known as Bootstrap Aggregation. Each model is trained independently on the created subsets of the dataset. The results of all the models are combined together and the final output is generated based on majority voting. This step of combining all the results of all generated models to get the final result is known as aggregation. Figure 3.16 explains the idea of a random forest classifier.

Some of the advantages of using RF classifier are:

- As the output is based on majority voting or averages it solves the problem of overfitting.
- The performance RF is not affected by null/missing values.
- Individual decision trees are created independently of each other thus it shows the property of parallelization.
- In RF classifier each tree does not consider all the attributes, so the feature space is reduced.

Some of the limitations of the RF include

- It is a complex algorithm when compared to decision trees.
- The training time of RF is more as compared to other models due to its complexity.

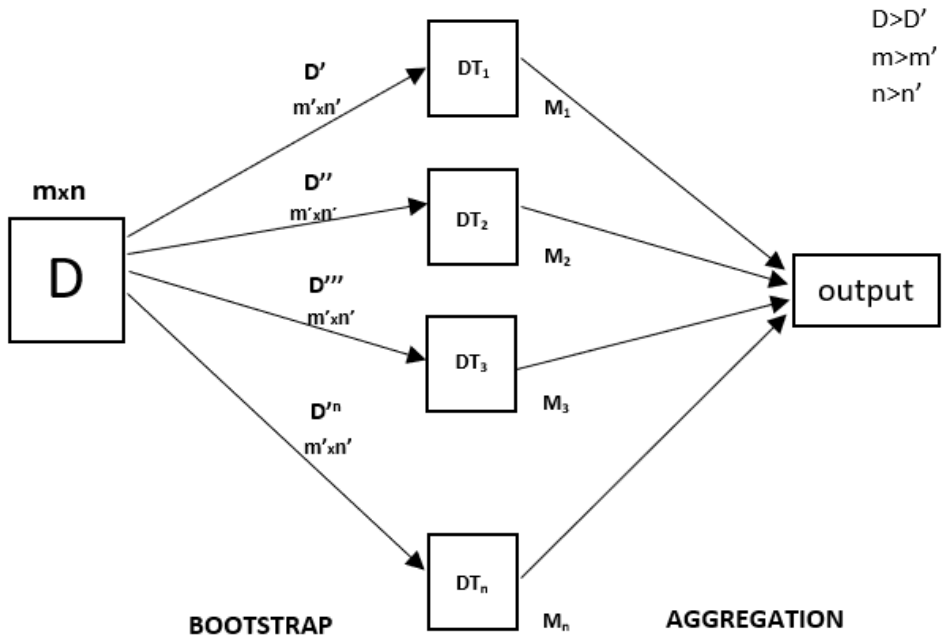

Figure 3.16: Working of RF [113]

XGBoost

XGBoost is an ensemble technique based on Gradient Boosted decision trees. As mentioned earlier in the ensembled technique multiple classifiers are utilized to enhance the performance of the final classifier. XGboost utilizes gradient boosting to implement the ensemble technique. In gradient boosting, each classifier corrects its predecessor's error. Each predictor is trained using the residual error of the predecessor as labels as shown in Figure 3.17. This technique is known as gradient-boosted trees. In this algorithm, decision trees are created in sequential form. Weights are assigned to all the independent variables which are then fed into a decision tree that generates results. The weights of the wrong predictions made by the first decision tree are increased and then fed into the second decision tree. These individual classifiers then ensemble to give a better performing model.

XGBooost is equally suitable for classification and regression problems.

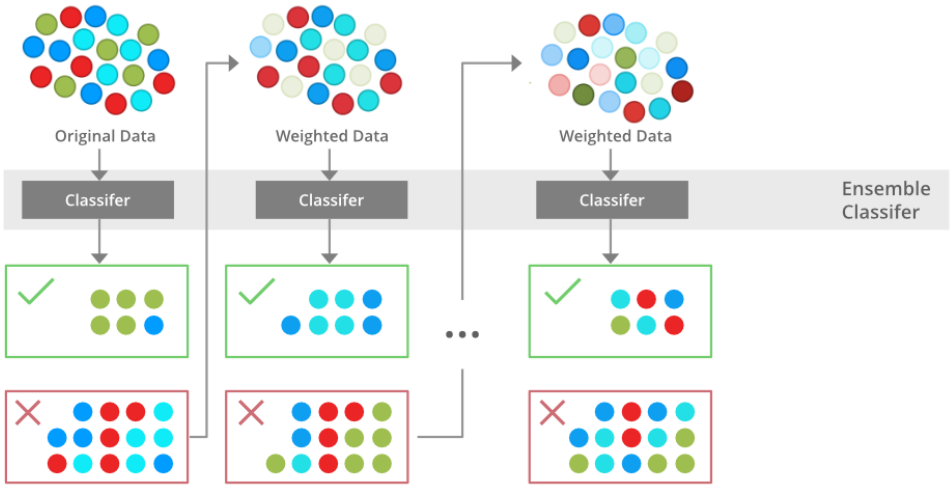

Figure 3.17: Working of XGBoost [114]

Some key parameters of XGBooost are mentioned below:

- The ensembling of multiple decision trees is used which can lead to a very complex model. Regularization is utilized to penalize the highly complex model.
- In XGboost the multiple decision trees cannot be trained in parallel so the data need to be stored in order. The cost of storing the data is high. In order to reduce the cost, it stores the data in blocks. It stored the data in the compressed column format, with each column sorted by the corresponding feature value which helps in improving the performance by offsetting any parallelization overheads in computation.
- Tree Pruning is utilized to limit the nodes of a tree. Maximum depth is defined before training the algorithm.
- Like a decision tree, this algorithm is suitable for data with missing values and can classify this type of data efficiently.

- This algorithm comes with a built-in cross-validation method to prevent overfitting during the training of the classifier.

Artificial Neural Network

Artificial neural networks (ANN) are inspired by the biological neural networks found in the human brain. These Neural Network (NN)s are able to learn from images, real-life objects and text to perform different tasks, for example, classify the objects in different categories, and translate the text into different languages. NN consists of layers performing different tasks to achieve the required results. The number of layers can vary from three to hundreds. NN made up of only three or four layers are known as shallow neural networks, whereas networks containing more than four layers are known as deep neural networks. Deep neural networks break down the input into several levels of abstraction hence deep learning models are able to learn and understand the input better than a shallow neural network.

Neural Network Architecture

The basic building blocks of a neural network include layers, nodes, edges, biases and activation functions.

Each layer of NN is made up of individual nodes called neurons. The number of neurons depends on the requirement of the layer and/or the required task performed by the NN. There are three kinds of layers present in a neural network.

- The input layer consists of the input data entering the neural network. In any NN there is only one input layer and it learns from the provided input to generate output. The input layer nodes are connected to each node present in the proceeding layer. The variables of input data are known as features and the output is dependent on

these input features.

- The hidden layer is the layer where the actual computation to perform the required task is computed. A hidden layer is made up of nodes known as activation nodes. Each node possesses an activation function which is a mathematical function that is applied to the provided input to generate output. This is the layer that is repeated multiple times according to the actions needed to be performed by a NN. The output generated by one hidden layer is fed to another hidden layer depending upon the number of hidden layers used in a NN.
- The output layer is the last layer of the NN and it consists of nodes that provide the final output. A typical NN is shown in Figure 3.18.

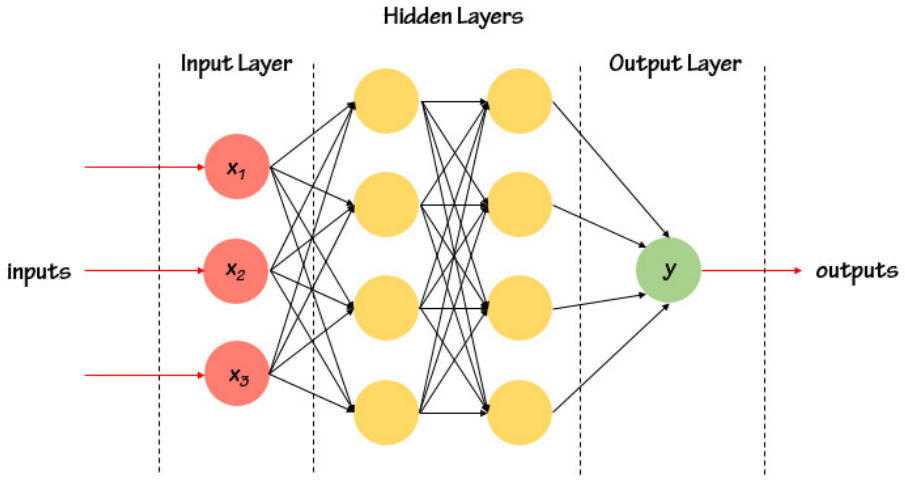

Figure 3.18: Artificial Neural Network [99]

Each neuron contains an activation that shows whether the neuron is active or not. An activation function is responsible decide the state of the neuron depending on the inputs of a neuron and the weighted sum. The output function is responsible to generate output for a particular activation node based on the activation function. These all components lie in the

activation nodes present in the hidden layers. Input and output layer neurons do not have these functions.

A connection between two nodes from two different layers is known as an edge. Each edge that is connected to the activation node has its own weight which can be considered as the weightage assigned to an input. Weights can be positive or negative. The values of all input nodes are multiplied with weights on each edge then these values are added together to get a weighted sum. The weighted sum basically shows how much impact that nose has on the input. If the value is small the output will be less affected by this value and if this value is large the output will be highly affected by this value.

Bias is a value added to the calculate weighted sum hence contributing to the output of the node. It allows the activation function to shift either to the right or to the left. This helps the model to better fit the data producing higher accuracies.

As mentioned earlier the activation functions are embedded in the activation nodes. They introduce non-linearity into neural networks enabling them to learn more complex functional relations that exist within the data. In summary, the activation node calculates the weighted sum of inputs adds a biase to it and then applies an activation function to generate the final output. This generated output will then be fed to the next layer as input as shown in Figure 3.19 . This is the way data flows in a NN. There are several activation functions used in NN according to the need of a particular problem. The most commonly used activation functions are step function, logistic sigmoid and ReLU as shown in Figure 3.20, Figures 3.21 and 3.22 respectively.

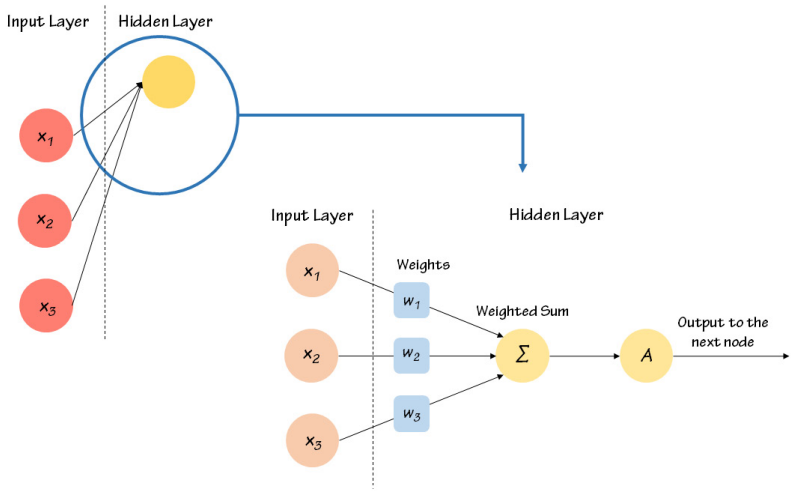

Figure 3.19: Working of Artificial Neural Network [99]

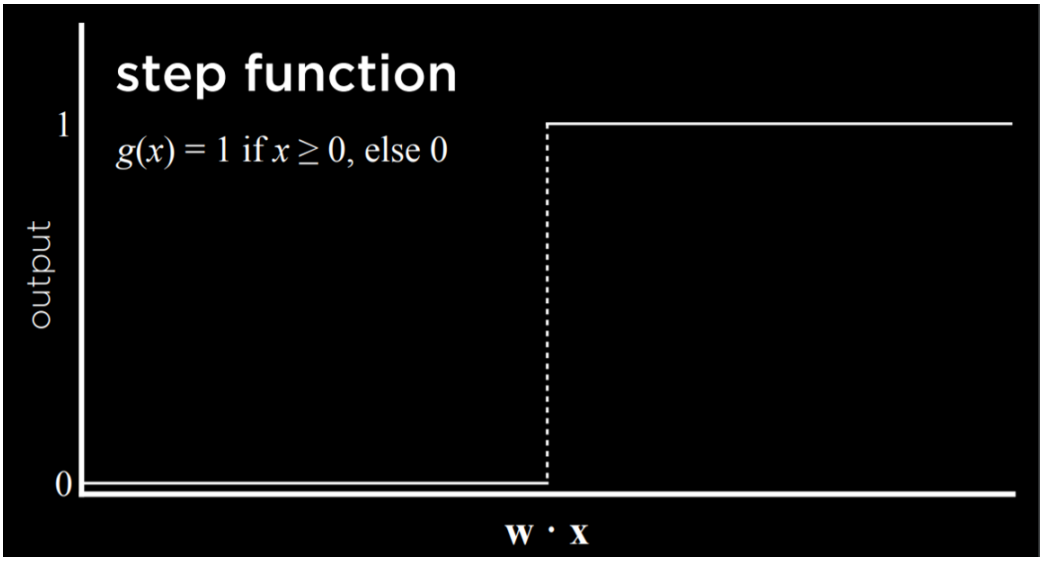

Figure 3.20: Step Function [115]

3.2 Chatbot

Chatbots are interactive systems that allow a human to interact with machines in natural language. Chatbots usually interact via text messages but speech interfaces can also be

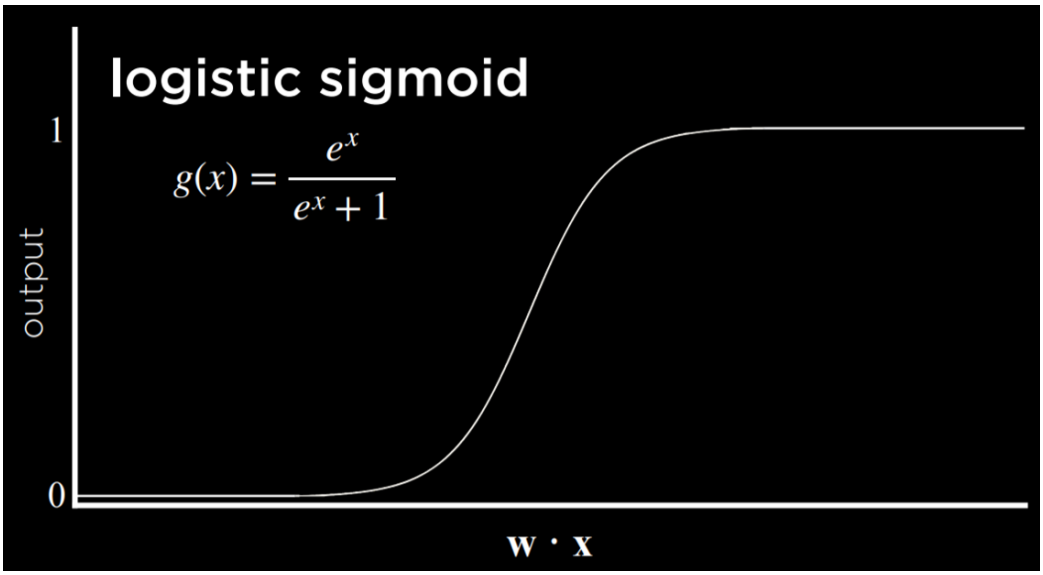

Figure 3.21: Logistic sigmiod [115]

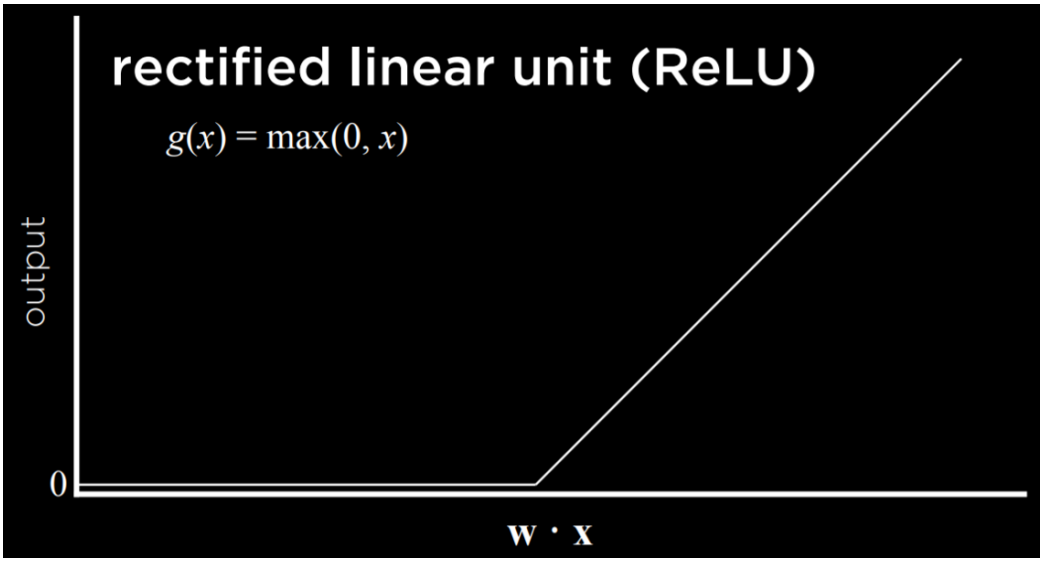

Figure 3.22: ReLU [115]

integrated. Platforms like Siri, Amazon Alexa and Google Assistance are some examples of chatbots. Nowadays chatbots are integrated with websites to support customers in many

aspects. The basic tasks of chatbots include understanding the language and generating meaningful and in context answers to the asked questions. The understanding of the language and generation of language both are part of NLP.

The history of the chatbot started in 1966 when Joseph Weizebaum built ELIZA the first chatbot ever. ELIZA was based on regular expression and rules. Many spoken dialog systems were built in 1980 -2000 but these bots were generating fixed responses fed to the bot and all are rule-based. They were unable to answer any question not provided in the rules defined for them. In recent years, chatbots have become more intelligent and useful due to the recent advances in ML and AI.

A chatbot can access a range of knowledge, which determines its Knowledge Domain. Chatbots that can answer any kind of question-related to any domain are called Generic Chatbots. Domain-specific chatbots can answer only questions concerning a domain. Another kind of chatbot is known as an interpersonal chatbot assisting people with different booking services like making a reservation in a restaurant or booking a service on an airline. Intrapersonal chatbots are close companions that understand the needs of a person. Finally, inter-agent chatbots provide communication with other chatbots for example Alexa and Cortana are two chatbots that were connected to each other to communicate with each other [116].

There are two commonly used approaches to developing a chatbot: pattern matching and machine learning approaches.

3.2.1 Pattern matching approach

Rule base chatbots match the user input to a rule pattern and select the most suitable pre-defined answer from a set of responses using pattern-matching algorithms. The response

format and context can also contribute to the rule selection. Rule-based chatbots usually do not create any answer, it only selects an answer from the predefined rules generated by the developer. ELIZA and its successor ALICE were the first rule-based chatbots. The performance of the rule-based chatbot is highly dependent upon the database containing all the rules. The more extensive the rules are the better response will be generated by the chatbot. The developer needs to define thousands of rules to get an acceptable response from the chatbots. This process is time-consuming and lots of manual effort is required. This type of chatbot is sensitive to grammatical and syntactic error's in the user's response. Rule-based chatbots usually perform better for single-turn communication, if multiple questions are asked in a single instance the bots usually reply to the last question. The rule-based chatbots response are automated and repeated due to the limited number of responses in the database. On the other hand, the response time is fast as a syntactic or semantic analysis of the user input.

The two most commonly used languages to implement rule-based chatbots are the following:

- Artificial Intelligence Markup Language (AIML) is based in XML and an open-source language to generate rule-based chatbots. ALICE was the first chatbot with the pattern-matching approach in the Artificial Intelligence Markup Language (AIML) language. AIML is the most commonly used chatbot language due to its usability and ease of learning and execution. AIML data objects consist of topics and the relevant categories are part of these topics. A category is a rule defined for the chatbot containing a pattern to represent the user's input and a template of the chatbot's response. The pattern usually includes single spaces, words, and wildcard symbols. An object is known as Graphmaster and it is represented as a tree with its

nodes representing the categories and leaves defining the templates of the chatbot's responses.AIML uses first depth search in the Graphmaster to match the best rule with the user input. Figure 3.23 represents some of the rules defined for a chatbot using AIML. Nowadays AIML is used together with Latent semantic analysis (LSA) so AIML is used to search the pattern of the user's input in defined rules. If the user's utterance does not match any rule, the answer is generated by LSA.

- Chatscript is a professional system for developing rule-based chatbots with an open-source scripting language. Some improvements are made as compared to AIML markup language. In addition to matching the user input with the defined rules, it also improves the user input in terms of grammar, syntax and semantics. Long-term and short-term memory is added to the system by using variables that store user-specific information to improve the user experience.

3.2.2 ML based chatbots

Machine learning-based chatbots use NLP to extract the content from the user input and have the ability to learn from the conversations. They need extensive training before implementation and do not require predefined responses for each possible user utterance. Usually, this kind of chatbot considers the whole conversation context to generate responses and not only considers the current user input. As ML base chatbots require a lot of training so large datasets are needed to get a good model of a chatbot. The ML based chatbots are either retrieval-based models or generative models. Usually, ANN are used to implement these chatbots. Retrieval-based models use NN to assign scores to the most suitable response from the set of generated responses. Generative models generate the reply using deep learning techniques. NLP based chatbots use multiple modules to perform different

```
<category>
   <pattern> HELLO </pattern>
    <template>
      <random>
          <li> Hi! What's your name? </li>
         <li> Hello, How are you? </li>
          <li> Hello! </li>
      </random>
   </template>
 </category>

<category>
   <pattern> MYNAME IS * </pattern>
   <template>Nice to meet you <set name="nameUser"><star/></set></template>
</category>

<category>
   <pattern> NIGHT </pattern>
    <template>Good night <get name="nameUser"/></template>
</category>

<category>
   <pattern> _ NIGHT </pattern>
   <template><srai> NIGHT </srai></template>
</category>

<category>
   <pattern> NIGHT * </pattern>
   <template><srai> NIGHT </srai></template>
</category>

<category>
   <pattern> _ NIGHT * </pattern>
   <template><srai> NIGHT </srai></template>
</category>
```

Figure 3.23: Rules for pattern based chatbot using AIML [116]

subtasks. They consist of Natural language understanding (NLU)which is responsible to understand a text and Natural Language Generator (NLG) which is responsible to generate the text commonly conducted by ANN. NLU is used to retrieve context from the

unstructured user utterance in natural language. NLU supports intent classification and entity extraction by considering the context information.

3.2.3 General architecture of chatbots

In this section general architecture of a chatbot is discussed. Many chatbot models may or may not include all the components mentioned in the architecture but it will help a user to have a complete understanding of the chatbot architecture. The general architecture of the chatbot is shown in Figure 3.24.

User Interface Component

User Interface receives messages from users through an application using text or speech input like Facebook, Slack, Viber or Skype.

User Message Analysis Component

The user interface controller drives the user's request to the Message analysis component to extract useful information from the message. The main information collected from the message includes the user's intention and extraction of entities which are useful for pattern-matching chatbots to choose the right category. For ML based chatbots, this is a very important component as it determines the user's intent which is helpful for retrieval-based chatbots to choose the most relevant answer and to generate the most relevant answer for generative models. In the case of ML based chatbots, this component is known as NLU which is responsible to process the user input to extract the purpose of the message which is the intent. The chatbot needs to understand the intent to perform the required action. Another important action taken in this component is entity identification which includes

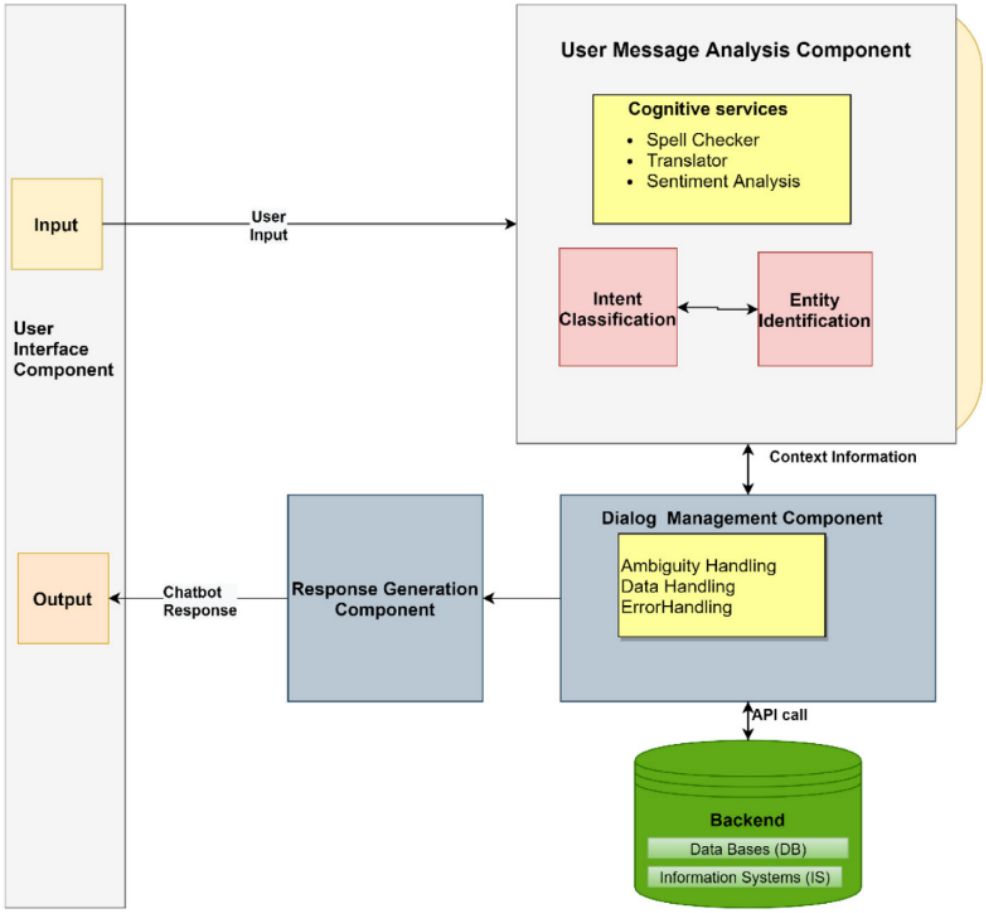

Figure 3.24: General Architecture of a chatbot [116]

the extraction of key elements in the unstructured text like names of people and places.

Some cognitive tasks are also performed in order to make the user message more understandable like spelling correction, translation of text in case of multilingual chatbot users and sentiment analysis to the user input to know about the user's positive or negative opinion within the text.

Dialog Management Component

The dialog management component is responsible to maintain the conversation between the user and the chatbot. It keeps the current intent and if the chatbot is unable to understand the context it asks questions to understand the intent. It is also responsible for the follow-up questions after recognizing the intent. Usually, it includes the following components:

- Ambiguity handling is responsible to give answers when the chatbot is unable to find the intent of the user. The chatbot may indicate that it did not have an answer or ask more questions to understand the intent, start a new topic or give a general answer to keep the conversation going.
- Data handling is also catered in the dialog management component. User information is stored in a file by this component of the chatbot.
- Error handling is crucial to tackling any unexpected errors to ensure the proper working of the chatbot.

After understanding the intent the chatbot may generate a response to the user or it can retrieve the information from the backend. In the first case, the control will be transferred to the response-generating component. In the latter case, the control passes to the backend.

Backend

The chatbot retrieves the information required to address the intent of the user's input from the backend through external APIs calls or database requests. After getting the suitable information from the backend it is forwarded to the dialog management module and then passed to the response generation module. In pattern-based chatbots, the knowledge base (KB) is defined which includes the hand-written responses that are retrieved from KB when match with the user input. The defined KB must cover a wide variety of responses to enable

the chatbot to respond to a variety of user inputs. A relational database is used to facilitate communication between KB and the chatbot. To maintain the consistency and relevance to the dialog generated by the chatbot access to previous information of user input is required. A relational database can be used to recall past conversations. Creation of the KB is a time-consuming task as it requires the developer to define the pattern and responses but it is crucial to building an extensive KB so a chatbot can generate responses to the user input. The responses generated by the rule-based chatbots are usually emotionless and do not look like human-generated sentences. To accommodate this some tricks are implemented such as deliberate typing mistakes, irrational responses and responses showing the existence of a personality.

Response generation component

The response-generating component is responsible to generate the response to the user's input. In a rule-based chatbot, the response is selected from the pre-defined responses included in the KB without generating a new text response. The retrieval-based model is more flexible; it analyzes the multiple responses generated by the available resources using APIs and selects the most suitable response. The Generative model uses NLG to respond in a human-like natural language based on the last and previous inputs. This model generates the best responses as compared to the other two models however training these models is a big challenge as it requires an extensive dataset to learn to generate meaningful responses. Finally, the generated response is presented to the user and the chatbot waits for the user's feedback.

3.2.4 ML algorithms used in chatbots

Artifical Neural Network

ANN are inspired by the biological neural networks found in the human brain. These NNs are able to learn from images, real-life objects and text to perform different tasks, for example, classify the objects in different categories, and translate the text into different languages. The detailed working of ANN is included in section 3.1.5.

If we want to detect a human in an image, the single image can be used as input in ANN. Based on different features the model will try to classify whether a human is seen in the image or not. However, if we need to understand the action of a person in an image then a single input image may not be enough. To detect the action of a person in an image series of images is required and a model needs to consider a sequence of images to understand the action of a person in an image. For this purpose, simple ANN or feed-forward NN is not useful because it does not remember the sequence of the input. For this purpose, Recurrent Neural Network (RNN) are utilized.

Recurrent Neural Network

The main difference between feed-forward NN and RNN is that RNN have memory elements called states which keep the information from the previous inputs. RNN are trained on the sequence of inputs rather than a single input. The state elements contain information about the previous inputs to process the current input sequence as shown in Figure 3.25. For each input in the input sequence, RNN gets a state, generates its output and sends its state to the following input sequence. The process is repeated for all the elements in the sequence.

The inputs in the feedforward neural networks are independent of each other so input

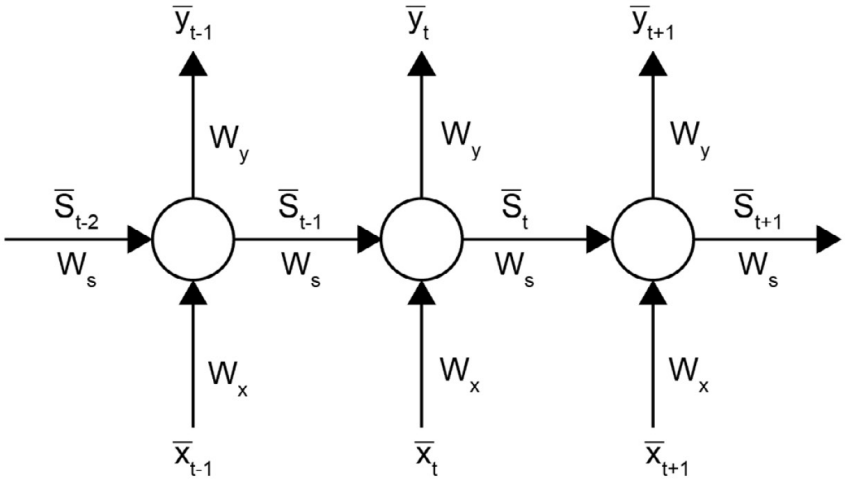

Figure 3.25: Architecture of RNN [99]

can be fed into the network in any sequence. The generated output is a function of input and weights as shown in equation 3.26.

$$\overline{y_t} = F(\overline{x}_t, W) \tag{3.26}$$

In contrast, at any time t the output of the RNN not only depends on the current input and weights but also on the previous inputs. The output of RNN at time t is defined as follows:

$$\overline{y_t} = F(\overline{x}_t, \overline{x}_{t-1}, \overline{x}_{t-2}, ..., \overline{x}_{t-t_0}, W) \tag{3.27}$$

Transformer

The transformer is an architecture to solve tasks sequence-to-sequence and handle long-time dependencies [117]. In contrast to RNN, the transformer does not require to represent the input in a sequential form. The Transformer relies entirely on the self-attention mechanism. The basic architecture of transformers is shown in Figure 3.26. The transformer model is

based on the encoder-decoder architecture. The encoder is responsible for stepping through the input time steps and encoding the entire sequence into a fixed-length vector called a context vector. The decoder is responsible for stepping through the output time steps while reading from the context vector.

In a sequence-to-sequence model, the series of words one at a time is fed into an encoder. The encoder process each item in the input sequence it captures the context of the sentence and produces a context vector. The encode and decoders are simply RNNs. After processing the entire sequence, the encoder sends the context to the decoder, which produces the output sequence item by item. The user can set up the context vector size and it can be 256, 512 or 1024. As explained earlier the RNN takes two inputs (input vector at time t and a hidden state) and produces an output. This output is then fed into the second encoder with the following sequence of input. This process will repeat and then in the end the output of the last encoder is fed to the first decoder.

In the transformer model, instead of passing only one output to the decoder, all the hidden states used in the encoders all passed to the first decoder. All the hidden states are then assigned to a score based on their relevance to the input sequence of the present time step of input. Then each hidden state is multiplied by its softmax score which will amplify the value of the hidden states with high scores. The process can be summarized in the following steps:

- All the hidden states of the encoders are passed to the first decoder. Let's call the initial hidden states $h1$, $h2$ and $h3$.
- The decoder processes its inputs, producing an output and a new hidden state

$$(h4)$$

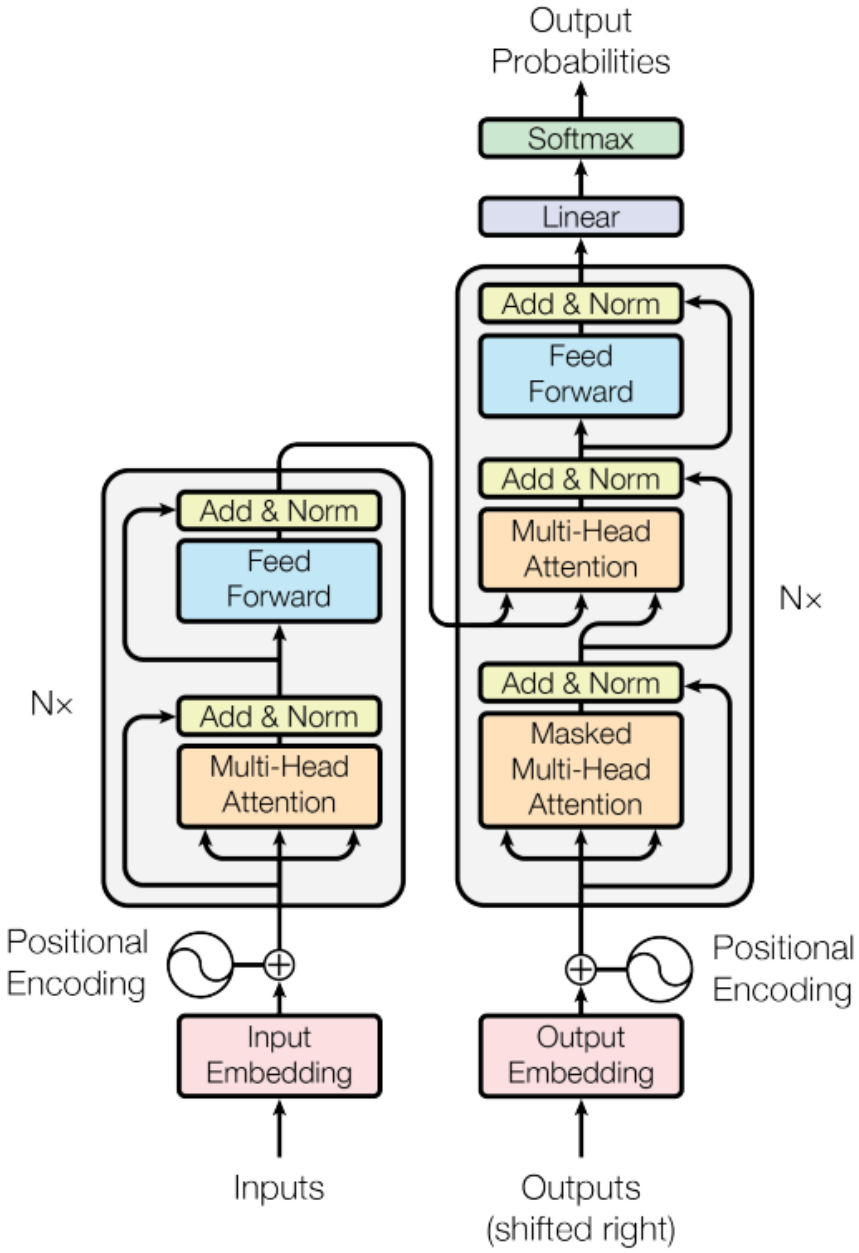

Figure 3.26: Architecture of RNN [117]

. The output is discarded.

- The input hidden states $(h1, h2, h3)$ to the first decoder and the newly produced hidden state $(h4)$ vector are used to calculate the context vector $(C4)$.
- The $h4$ and $C4$ are concatenated to produce a single vector.
- This vector is then passed to the feed-forward neural network as shown in Figure 3.26.
- The output of the feed-forward neural network is the output of this time step.
- This process is repeated for each time step.

This mechanism is called self-attention. This mechanism is used in the training of the transformer model. There are three kinds of attention in a model:

Encoder-decoder Attention Attention between the input sequence and output sequence. Self attention in the input sequence Attends to all the words in the input sequence. Self attention in the output sequence: Attends to all the words in the output before the time step. Self attention is limited to the words that occur before a given word. For this all the output words occurring after the given word are masked.

DialoGPT

DIALOGPT is a tunable gigawordscale neural network model for the generation of conversational responses [118]. DialoGPT is based on a Generative Pre-trained transformer (GPT-2) model and uses a multi-layer transformer as a model architecture. DialoGPT is trained on a large-scale dialogue pair retrieved from Reddit discussion. GPT-2 model is helpful in creating continuous text that contains relevant information on the topic presented to the model [119]. During experimentation, it was observed that the sentences generated by DialoGPT are diverse and contain relevant information specific to the user prompt. The

evaluation of the model is performed on the public benchmark dataset DSTC-7 and a new 6k multireference test dataset extracted from Reddit postings. State-of-the-art results are obtained in both automatic and human evaluation showing the performance of DialoGPT is near-human response quality.

The dataset used for training of DialoGPT is retrieved from comments chains scraped from Reddit spanning from 2005 – 2017. The training of the DialoGPT model is on the basis of the GPT-2. The GPT-2 model is based on the generic transformer language model and uses a stack of masked multi-head self attention layers consisting of encoders and decoders to train on web-text data. The text generated by DialoGPT is realist-looking and generated from scratch. The model is using a 12-to-48-layer transformer with layer normalization used in GPT-2. Three different sizes of DialoGPT are available trained on total parameters of 117M, 345M and 762M available as small, medium and large models.

3.2.5 Integration of chatbot with Applications

The working of a chatbot is highly dependent on its integration into the application. There are many ways to integrate a chatbot into the application. The most commonly used ways of integration are as follows

Integration using API's

The Application Programming Interface (API) is a set of definitions and communication protocols for building and integrating application software. API is a software architecture style, which provides a framework to build software that can communicate across devices, other software and different platforms. Application Programming Interface (API)s allow unrelated software products to integrate and interoperate with other software, device and

data seamlessly. APIs also allow the developer to add features and functionality to the developed software.

API protocols and architectures

APIs exchange commands and data that require protocols, architectures, rules, structures and constraints that regulate an API's operation. There are three kinds of commonly used API protocols: REST, RPC and SOAP [120].

- The representational state transfer (REST) architecture is the most popular approach to building APIs. REST architecture is based on client-server architecture. In the client-server approach, the front and back end of APIs are separated and hence it provides flexibility to use and implement these APIs. REST API are stateless, which means the API stores no data or status between requests. REST API uses HTTP requests such as GET, PUT, PATCH, DELETE, TRACE, CONNECT, POST, OPTIONS, etc. to provide communication and manipulation of data that is stored on the Web server. It ensures communication between the Web Server and the Client. Using RESTful API, it is possible to make a single chatbot instance up and running on multiple different websites. A chatbot service can be made by utilizing the RESTful architecture. Using REST APIs chatbot can be integrated into multiple websites where the respective service is required. The overall structure of the REST web service can be visualized as provided in Figure 3.27.
- The remote procedural call (RPC) protocol is a simple means to send multiple parameters and receive results. The REST APIs are mainly used for the exchange of data and documents but RPC APIs are used to execute the coded actions and processes. For coding, RPC utilizes two different languages, JSON and XML.
- The simple object access protocol (SOAP) is a messaging standard defined by the

World Wide Web Consortium. It is generally used to create web APIs using XML. SOAP supports a wide range of communication protocols used over the internet such as HTTP, SMTP and TCP. SOAP is also extensible and style-independent, so developers can write SOAP APIs in different ways and easily add features and functionality. SOAP is highly structured and it has defined standards to be considered before utilizing it. Usually, SOAP messages contain four components; envelope, header, body and fault.

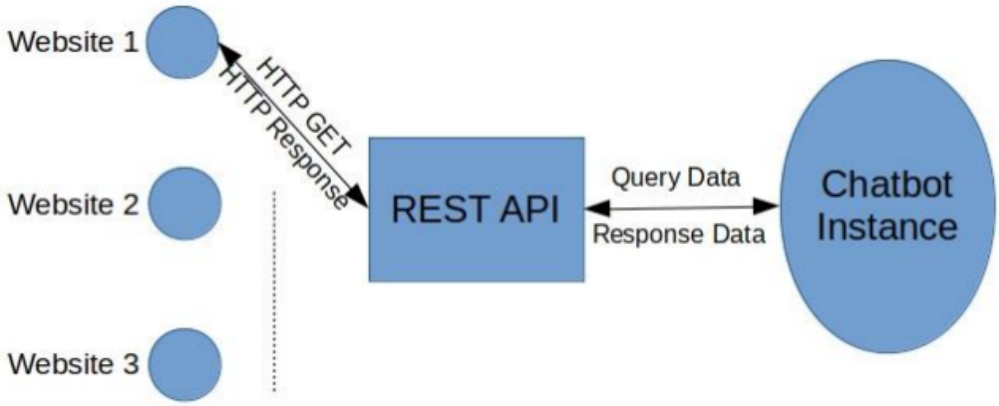

Figure 3.27: REST APIs working [121]

The choice of API to be used for integration of a chatbot with the application can have a long-term impact on its level of security, the complexity of the information that is needed to exchange and the speed or performance required by the chatbot. REST and SOAP APIs are designed to connect applications and mainly utilize HTTP protocols and commands. Both can use XML in requests and responses. However, SOAP depends on XML by design, while REST can also use JSON, HTML and plain text. SOAP uses strict rules, while REST allows flexibility in its rules and is instead governed by architectures. SOAP is built from remote procedure calls, while REST is based on resources [120].

Both REST and SOAP exchange information but in different ways. SOAP is used when an enterprise requires tight security and clearly defined rules to support more complex data exchanges and the ability to call procedures. Developers frequently use SOAP for internal or partner APIs. REST is used for fast exchanges of relatively simple data. REST can also support greater scalability, supporting large and active user bases. These characteristics make REST popular for public APIs, such as in mobile applications. RPC is only useful for designing simple APIs and hence not suitable for public APIs.

Manual Integration of APIs

If the user intends to use a chatbot only on a single website manual integration provides the best and quick way to do the integration. In this type of integration, the chatbot, website as well as server resides on the same server. Hence, this is the simplest kind of integration that a chatbot can have.

Third party integration

If the developer does not want to design a customized API but wants to have the full functionality then third-party API 's providing a good alternative.

4. ProtectBot

4.1 Introduction

This Chapter provides a detailed architecture of the proposed solution, shown in Figure 1.1, *Protectbot*: A chatbot to detect child predators on gaming platforms. *Protectbot* is a text-based chatbot that is designed to have conversations with suspects on any gaming platform. The chat between the *Protectbot* and the suspects were not stored to protect the privacy of users. The chat is pre-filtered on run time and passed to the pre-processing units. The pre-processed data is fed to the pre-trained fastText model which generates the numerical vectors of the input data. These numerical vectors were saved using the SQLite database. Finally, the chats were classified as predatory or non-predatory. If the predatory behavior is detected the *Protectbot* would generate a report for the LEA. The material covered in this Chapter has been submitted for publication in *IEEE Access* under the title of *Protectbot: A Chatbot to Protect Children on Gaming Platform*.

Over the past two decades, research to explore the use of AI tools and ML algorithms for children's online safety has increased. However, the main focus of this research remains restricted to the detection of predatory behavior in chat logs. This solution is not enough to protect children from predators, as it is only a step toward building a mechanism to

protect children online. The authors in [25, 26, 28, 122] picked chat logs from selected games, such as MovieStarPlanet [25], Online Battle Arena [26], World of Tanks [122], and Dota Ragnarok [28], to detect predatory behavior in online gaming. It is observed that the focus of research on detecting predatory behavior is on chat logs pertaining to social media's chatting platforms [122–126]. Gaming platforms have rarely been explored. The main reason for this research gap is the unavailability of public datasets of online game players' chat logs. Owing to privacy issues, the data are not stored or saved, which makes it difficult to build a dataset.

The literature that detects child predators on chatting platforms can be subdivided into two main categories: conversational models that detect child predators in a chat environment, and classification models that detect child predators in chat logs. Table 4.1 provides a summary of the chatbots proposed in the literature along with the models and results. Distributed platforms chatbots have been developed to detect child predators on chatting platforms [29], [36], [42]. However, owing to the limited availability of relevant datasets, bots have not been tested on balanced and large datasets.

Negobot was the first prominent contribution in developing an AI agent to detect a child's predatory behavior on chatting platforms [29]. Negobot was trained on a dataset consisting of 377 chats taken from the PJ website [127]. When the conversation is started, the user sends a text to Negobot, which processes the data using a conversational unit by removing all emotions and slang, translating when needed, creating a meaningful sentence, and feeding it into a high-performance information retrieval (IR) tool, which evaluates the similarity index of the conversation with conversations taken from the PJ. Negobot uses a structure of seven chatter-bots specifically designed to perform in different scenarios depending on the conversation topic. AIML is used to provide Negobot with the ability to

converse with users. Negobot collects as much information as possible by applying the game theory. Finally, the chat was classified into levels assigned by the system from -1 to +3 such that -1 is assigned to the chat demonstrating the least predatorial content while +3 indicates a high likelihood of predatorial content from a suspect. The system was tested by using two chats, one of which was aggressive and the other was passive. System testing is not extensive enough to claim the performance of Negobot.

A conceptual platform called BotHook was proposed in [30]. BotHook is a chatbot containing three major modules. The first module is the capture, classification and analysis of cybercriminals and cyberpedophiles module (CCAM), which attracts and analyzes attacks on the system. The second module is bot module (BOTM), which is responsible for an interactive chat with suspects without showing its identity as a bot. The third module is the pedophile trend characterization module (PTCM), which assigns the value of a pedophile trend to users based on their chats. This work is a theoretical proposal, whereby practical implementation and testing were not performed. In [36], a chatbot was designed to collect data from the website Omegle [128]. An SVM classifier was used as an emotional classifier while MNB was used as an opinion classifier. The users who were not interested in child predatory behavior were labelled as "indifferent". The people who showed an interest in predatory behavior without committing an offense were labelled as "interested". Finally, the users who actually behaved as child predators were labelled as "perverts". Later, this work was extended to [42] and the results were improved by deploying the chatbot for 50 days and collecting information from 7,199 users.

The literature shows that the detection of predatory behavior in chatlogs remains the main focus over the past two decades. The detection of child predators was modeled as a text classification task. Table 4.2 summarizes the classification models proposed in the

Table 4.1: Proposed chatbots to detect child predators.

Ref.	Conversational model	Classification model	Dataset Source	Results
[29]	AIML	Game Theory	PJ	Classified two chats
[30]	BOTM	PTCM	N/A	N/A
[36]	AIML, LSTM-NN	Emotion classifier (SVM), Opinion classifier (MNB)	Omegle	Classified 35 chats as indifferent, suspected and perverts
[42]	AIML, LSTM-NN	Emotion classifier (SVM), Opinion classifier (MNB)	Omegle	Classified 7,199 users as indifferent, interested and perverts

literature to detect child predatory behavior. In [32], a classifier was built to label chats as grooming or non-grooming. The dataset was compiled using 105 grooming chats collected from the PJ website and 45 non-grooming chats collected from the website [129]. Seventy grooming chats and 30 non-grooming chats were used for training purposes. The test data consisted of 35 grooming and 15 non-grooming chats. Seventeen characteristics of grooming chats were identified in the training and test datasets. It was deducted that chats can be classified using the number of grooming characteristics found in a conversation. The proposed classifier labeled a chat as grooming if 11 out of 17 characteristics were found in the conversation. Otherwise, the chat was labeled as non-grooming. The proposed classifier achieved an accuracy of 96.8% using the built database. For the same dataset, the accuracies of the SVM and KNN were 98.6% and 97.8%, respectively.

The authors of [33] used the PAN13 dataset [130] to train a classifier using a five-step process to detect a child's predatory behavior in chat logs. As the PAN13 dataset was generated for age and gender detection in chat logs, it was included in the first step of the preprocessing of data, which eliminated all metadata except the words included in the conversation and conversation ID. In the next step, the feature extraction of the BOW and TF-IDF was applied using the fuzzy rough feature selection (FRFS) method, which was used to identify the most important features that describe the dataset. The data were then classified using Gaussian Naïve Bayes (GNB),RF, Logistic Regression (LR), and AdaBoost. Using different normalization techniques, such as l_1, l_2, and power

normalization (PN), the highest accuracy was achieved for the LR classifier using BOW, PN-l_2 normalization, and LR classification. This study was extended to [34], by developing a dataset using the two sources in [127], [131]. Additional classifiers, that is, linear and RBF coordinate descent fuzzy twin support vector machines (CDFTSVM), were tested.

The authors of [35, 38, 39] used the text classification approach to detect predatory behavior in chat logs using a wide variety of well-known classifiers such as SVM,Convolution Neural Network (CNN), deep artificial networks (DAN), RF, and NB. In [37], three different approaches to detecting child predators were based on the message, author, and conversation using a wide variety of classifiers, such as LR, ridge, NB, SVM and NN. In [41], the authors used the text classification method with three types of features: textual, behavioral, and demographic features. The SVM and Bernoulli NB classifiers were used to classify the chats into two categories. The dataset used in this study was collected from PJ [127] and from [132]. In [43], a two-stage classifier was used whereby the messages were classified in the first stage and the whole conversation was classified in the second stage to detect predatory behavior in chat logs.

In [47], the authors explained how the task in the digital forensic investigation process could be mapped to ML methods. After proposing a mapping between digital forensic and ML methods, the classification of chat logs was performed for predatory and non-predatory chats using LR, XGBoost, multilayer perceptron (MLP) and long short-term memory (LSTM). In [48], the authors presented the detection of child grooming behavior in chat logs using an SVM. An age detection mechanism using deep neural networks (DNN) was introduced to reduce the false positives, as only chats with children were classified, whereas others were discarded. The dataset was developed using two sources [127], [133]. In [50], two types of features were extracted: vocabulary-based and emotional-based, which were

Table 4.2: Proposed classification models to detect child predators.

Ref.	Classification	Preprocessing	Feature Extraction Method	Feature Selection Method	Classifier
[25]	Binary	Yes	BOW, TF-IDF	NA	DT, MLP, LR, KNN, SVM
[26]	Binary	Yes	TF-IDF	NA	SVM
[32]	Binary	Yes	TF-IDF	17 defined characteristics	SVM, KNN
[33]	Binary & Multi-label	Yes	BOW, TF-IDF	FRFS	GNB, RF, AdaBoost, LR
[34]	Binary & Multi-label	Yes	BOW, TF-IDF	FRFS	GNB, RF, LR AdaBoost, LCDFTSVM, RCDFTSVM
[35]	Binary& Multi-label	NA	BOW, TF-IDF	FRFS	Backpropagation neural network
[28]	Binary	Yes	None	NA	CNN
[37]	Binary	Yes	BOW, TF-IDF	NA	LR, Ridge, NB, SVM, NN
[38]	Binary	Yes	Term frequency inverse document frequency inverse class space density frequency (TF.IDF.ICSDF)	FRFS	SVM
[39]	Multi-label	Yes	NA	NA	DAN, CNN
[40]	Binary	Yes	BOW, TF-IDF	NA	SVM linear, SVM non-linear, RF, NB
[41]	Binary	Yes	Manual textual, behavioral and demographical feature extraction	NA	SVM, Bernoulli NB
[43]	Binary	NA	NA	NA	Recurrent neural network
[44]	Binary	Yes	BOW, TF, TF-IDF	NA	MNB, Bernoulli NB, SVM, NN, KNN, LR, RF, DT
[45]	Binary	Yes	Continuous BOW, skip gram	NA	NLP, CNN
[46]	Binary	Yes	NA	NA	HGBDT
[47]	Binary	Yes	NA	NA	LR, XGBoost, MLP, LSTM
[48]	Binary	Yes	LIWC	NA	RF, NB, SVM, KNN, AdaBoost-DNN
[49]	Binary	Yes	Word embedding aggression	NA	Linear Discrimination Analysis, SVM, RF, LASSO, Generalized boosting machine
[50]	Binary	Yes	BOW, MoodBook	NA	DT, SVM, RF
[51]	Binary	Yes	CNN	NA	MLP CNN
[52]	Binary	Yes	Pre-trained bidirectional encoder representations from transformers (BERT)	NA	BERT_frozen, BERT_tuned

fed into a variety of classifiers, such as DT, SVM, and RF to classify the conversation as predatory or non-predatory. The researchers in [50] developed 2 datasets with respect to vocabulary-based and emotion-based features using Pan12 dataset [53].

In 2012, a competition was conducted to identify sexual predators in online chats. A dataset PAN12 was developed to perform two tasks: the first task was to identify the predator among all users in different conversations and the second task was to identify the lines or parts of the conversation that were distinctive to predatory behavior [53]. The authors of [53] provided a summary of all submissions to the competition. This study is a prominent contribution to the field of identifying predatory behavior in chats, as it provides a publicly available dataset containing predatory and non-predatory chats. The authors in [40, 44, 46, 52, 53] used Pan12 dataset to evaluate the performance of the proposed models. The best performer in this competition got the $F_{0.5}$ score of 0.93 [53]. The author did not use any pre-processing though pre-filtering was applied. The conversations had only one participant and long sequences of unrecognized characters were removed. Also, the conversation having less than 6 interventions per user were removed. For feature extraction, BOW with the TF-IDF weighting scheme was utilized and NN was used to classify the chats as predatory/ non-predatory chats.

The author in [40] prefiltered the conversation between more than two or less than

two chatters. The features were extracted using BoW models using the TF-IDF weighting scheme. A wide variety of ML algorithms SVM, RF and NB were trained using three different *n*-grams including 1-gram, 2-gram and 3-gram. The best results were obtained by using the linear SVM model with 1-gram and 2-gram features and removing the stop words. For the predatory conversations, using linear SVM model when 1-gram features were utilized, the best results of precision = 0.87, recall = 0.82 and F1-score = 0.86 were obtained. In [44], a two-stage classification was implemented, such that the chat logs were classified in the first stage using a wide variety of well-known classifiers and the results from the first stage were used for classification in the second stage using a soft voting-based ensemble. The author utilized three methods (i) BoW with TFIDF weighting scheme as feature extractor and SVM to classify the chats, (ii) BoW with binary weighting scheme as feature extractor and multinomial NB classifier to classify the chats and (iii) BOW with binary weighting scheme and logistic regression classifier to classify the chats. These three methods were used in stage 1 then soft voting was used to evaluate the final results. In [46], an approach was presented to address class imbalance using hybrid sampling and class re-distribution to build an augmented dataset, and then classify it by using histogram gradient boosted decision trees (HGBDT). The author in [52] applied a transfer learning approach using a pretrained Bidirectional Encoder Representations from Transformers (BERT) model to generate the word embeddings. Then the numerical vectors were fed into a feedforward neural network for classification. A recall of 0.99 and $F_0.5$ score of 0.98 was obtained.

As described above, efforts have been made by researchers to protect children from predators but in the literature, no solution was found to provide a comprehensive solution to protect children on the gaming platform. It is worth noting that in the literature the

developed chatbots were not tested on publicly available large datasets but rather tested on the small dataset developed by the authors in a controlled environment. The classification methods used to detect predatory behavior have not been applied to gaming platforms to avert real-time threats while children are playing and interacting. In literature, these classification methods are just limited to detecting the predatory behavior in chatlogs which does not provide a promising solution. In this work, we propose an integrated framework to protect children on the gaming platform. A chatbot *Protectbot* is developed to communicate with the suspects on the gaming platform. The chatbot is integrated with the best-performing classifier to classify the chat into predatory or nonpredatory chats. The classifier is trained and tested on Pan12 dataset [53] and shows the best performance as compared to available classifier models proposed in the literature.

The rest of the Chapter is designed as follows: section 4.2 provides a detailed working of the *Protectbot*. Section 4.3 includes the results generated by the proposed model along with the comparison of the proposed model with the available models in the literature. Section 4.4 concludes the Chapter.

4.2 Protectbot

The proposed framework of *Protectbot* is shown in Figure 1.1. *Protectbot* is using DialoGPT [118] to communicate with the suspects. DialoGPT could be integrated into any gaming platform by using REST/SOAP APIs. DialoGPT extends the open source GPT-2 model [119] to facilitate the chatbot to have a conversation with suspects without showing its identity. When the conversation is complete the chatlogs are classified as predatory chats or non-predatory chats. The pipeline for the classification model used by *Protectbot*

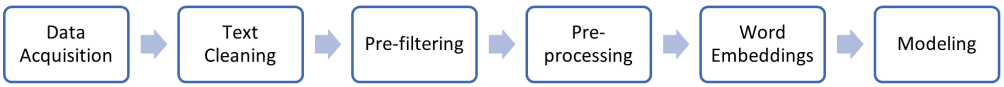

Figure 4.1: NLP pipeline

is shown in Figure 4.1. The different stages of the NLP pipeline are explained below:

4.2.1 Data Acquisition

Pan 12 sexual predator identification dataset

For the training and testing of *Protectbot*, Pan 12 sexual predator identification dataset [53] was used. The sexual predator identification competition was launched in 2012 by the Conference and Labs of the Evaluation Forum (CLEF) [53]. This dataset contains three types of chats (1) Normal conversations obtained from the two sources [134] and [131] (2) Sexual conversations among adults with mutual consent obtained from data collection [128], (3) child predators chats collected from PJ [127]. The PJ is a non-profitable American organization that initiated an operation to detect child pedophiles in different chatting platforms [127]. The trained police officers and volunteer adults behaved as a child to uncover the child pedophiles. When the user is confirmed to be a child pedophile, the LEA are alerted and informed, which helps the LEA to convict 623 child pedophiles. Chat logs of trained adults and child pedophiles are available on the website. Photos of the convicted are also shared on the website. PJ is the only resource available for public chat logs of verified child pedophiles. Many researchers have used it in their work.

The dataset contains two corpora; training corpus and test corpus. There were a total of

Table 4.3: Pan 12 dataset

Number of	Training Dataset	Test Dataset
Conversations	66927	155128
Predatory Conversations	2016	3737
Predators	142	254

66,927 conversations in the training dataset, consisting of 2,016 predatory chats and 64,911 non-predatory chats. In the test dataset, the conversations are 15,5128 out of which 3737 are predatory chats and the remaining are non-predatory chats as shown in Table 4.3. This dataset is highly imbalanced containing a large number of non-predatory chats to mimic the real-life situation in which the ratio of child predators chat is comparatively very low than that of normal chats. The dataset is provided in XML format containing conversations as shown in Figure 4.2. Each conversation is identified by a unique conversation ID consisting of a set of messages. Each message contains a line number, author ID of the message, time of message and text of the message. The dataset also contains a text file mentioning all the predator's IDs and predatory conversation IDs.

The Pan 12 dataset contains a small percentage of predatory chats (True positives) and a large number of normal chats (True negatives). The dataset also contains sexual conversations between adults which contributes to the false positives. Some predatory chats are incorrectly classified as non-predatory which contributes to the false negatives generated by the proposed model.

PJ dataset

We collected 71 full predatory conversations from the website PJ. Pan 12 data set was developed in 2012. To make sure the collected dataset is not overlapping with the Pan 12 dataset; we collected all the conversations that happened after 2012.

```
</conversation>
<conversation id="85f0abac6ef5a2a23814a2ced73b5fb7">
  <message line="1">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>hello there</text>
  </message>
  <message line="2">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>how are ya?</text>
  </message>
  <message line="3">
    <author>2ded7a428b8b4536d49393c352fe1d1c</author>
    <time>14:11</time>
    <text>hey</text>
  </message>
  <message line="4">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>so, where are you from, Stranger</text>
  </message>
</conversation>
<conversation id="80e3c3978ea07f46819f1f945cb04949">
  <message line="1">
    <author>a4529d1761aaeada66c6bdd6c93c78ea</author>
    <time>15:46</time>
```

Figure 4.2: Pan12 Dataset

Table 4.4: Pan 12 Dataset after pre-filtering

Number of	Training Dataset	Test Dataset
Conversations	8783	20608
Predatory Conversations	973	1730

4.2.2 Text Cleaning & Pre-filtering

The Pan 12 datasets are in XML format. To feed the data into the word embeddings model, the data is converted into CSV files containing four columns conversation ID, author ID, the text of messages and the label of the conversation. This conversation is done for the training dataset as well as the test dataset. The 0 is assigned as a label to the non-predatory conversation and 1 is assigned to the predatory conversation. A conversation is marked as predatory if the identity of one of its chatters is listed as a sexual predator. Note that the Pan 12 is a large dataset containing chats involving one author, two authors and more than two authors. Child predators always try to hide themselves; hence sexual predators will engage in 1-on-1 conversations with their victims and not engage in multi-person chats [37, 40, 44, 46, 51, 135]. Therefore all the conversations between more than 2 authors and less than two authors were filtered.

In Pan12 few conversations have less number of messages thus they do not provide enough information to any classifier to be classified as predatory or non-predatory [37, 40, 44, 46, 51, 135]. Therefore the conversations having less than six messages are filtered in the pre-filtering step. After the pre-filtering, the dimensions of the data were reduced. Now the training set contains 8783 non-predatory conversations and 973 predatory conversations. The test dataset contains 20608 non-predatory conversations and 1730 predatory conversations as shown in Table 4.4.

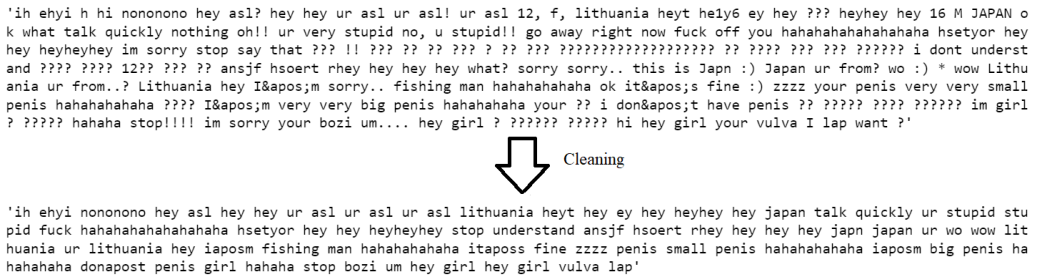

Figure 4.3: Example of predatory chat before and after cleaning

4.2.3 Pre-processing

The pan 12 dataset contains chat messages that usually do not follow any regular grammar rules thus the messages contain lots of misspelled words, acronyms or phrases like h r u? and words with repetitions of characters like “heyyyyy” and “soryyyyy”. In NLP pre-processing is a very important step; the information that is not important for the classification of text is usually filtered in this step. All the URLs, punctuations, special characters like $, & #, *, +, ānd numbers used in messages were removed. Some words such as is, are, the and a, appear frequently in a text but do not provide any valuable information in text mining or semantic analysis such words are known as stop words. In most NLP models, the stop words are removed from the dataset before generating the word embeddings. The stop words were removed from the Pan 12 training and test datasets. A predatory chat after and before all preprocessing steps is shown in Figure 4.3.

In preprocessing step, some researchers have removed all the misspelled words and tried to convert the chats into normal text but using this pre-processing a lot of relevant information that gives clues about predatory behavior can be removed. It is important to note that to keep the information as much as possible, no stemming or lemmatization in the preprocessing of the data was performed.

4.2.4 Word Embeddings & Classification

To perform classification, numerical vectors are needed to train and test a classification model. In the literature on predatory behavior identification, the most commonly used method to obtain numerical vectors was BoW. In BoW, a set of words present in a corpus is generated. Each word in the dictionary is presented by an index to a high-dimensional vector. A text is represented in a vector where all the entries are zero except those words in the text. The non-zero entries can be further represented in many ways, such as a binary value 1 or 0 it can be represented by term count, or it can be represented by term frequency or it can be represented by TF-IDF of the word. Clearly, BoW does not capture the relationship between words or the semantics of words. BOW is also not suitable for a large corpus as the dimension of the vectors representing the words is very high and sparse.

Word embedding is an alternative approach to represent the natural language in numerical vectors in such a way that similar-meaning words will be encoded into similar feature vectors and hence able to capture the semantics of words. The word2vec [136], Glove [137]and fastText [138] is the most commonly used methods to generate word embeddings. The word2vec and Glove are trained on words using skipGram or CBOW while fastText is trained on character n-grams of words. The word2vec and Glove are unable to generate the numerical vectors of the words that are not used in the training of the models. However, the fastText generates n-grams characters of words so it can generate numerical vectors of out-of-vocabulary words. Therefore this method provides better performance as compared to word2vec and Glove [103] The author in [103] shows the performance comparison of word2vec, Glove and fastText on the public dataset of news stories from UCI KDD Archie [104]. The results show that the fastText shows better results as compared to the other word embedding methods. Another research [105] shows the

performance comparison of the above-mentioned three embedding techniques on publicly available two datasets. The results showed that fastText performs better than word2vec and Glove.

For the detection of predatory behavior in chat logs developed by Pan 12, all the messages in a conversation are merged together to make a single sentence. This sentence is then converted into a numerical vector of 300 dimensions using a pre-trained fastText model *cc.en.300.bin* [138]. fastText is trained on data obtained from Wikipedia meta-pages, Statmt.org news, UMBC news, Gigaword and Common Crawl [138] containing 649.9 billion words. The generated sentence vectors are utilized to train the multiple classifiers KNN, SVM, RF and xgboost. The classifiers are then tested on Pan12 test dataset and a dataset containing 71 predatory chats obtained from PJ.

4.3 Results

The Pan12 train dataset was used to train all the models presented in this section. In addition to the Pan 12 test dataset, a dataset containing 71 full conversations of child predators obtained from the PJ was also utilized to evaluate the models. Class 0 indicated the non-predatory chats and class 1 is representing the predatory conversations. The results for the Pan12 test dataset are shown in Table 4.5. We leveraged the state-of-the-art methods that have shown the best performance for Pan 12 dataset. The bold values show the best-performing classification models. SVM shows the best averages for accuracy, recall, F_1-score and $F_{0.5}$-score. KNN shows the best recall of class 1, so if *Protectbot* is required to be more sensitive to detect the predators on the gaming platform the KNN classifier could be utilized.

Table 4.5: Results

Classifiers	**Accuracy**			**Recall**			**F_1-score**			**$F_{0.5}$-score**		
	class 0	class 1	Weighted Average	class 0	class 1	Weighted Average	class 0	class 1	Weighted Average	class 0	class 1	Weighted Average
xgboost	0.99	0.96	0.98	1	0.84	0.98	0.99	0.90	0.98	0.98	0.93	0.98
KNN (n=15, p=2)	1	0.83	0.98	0.98	**0.95**	0.98	0.99	0.88	0.98	0.99	0.85	0.98
SVM (C: 10, gamma: 1, kernel:rbf)	0.99	0.96	**0.99**	1	0.93	**0.99**	1	0.95	**0.99**	0.99	0.95	**0.99**
Random Forest	0.97	0.98	0.98	1	0.69	0.97	0.99	0.81	0.97	0.98	0.90	0.97

Table 4.6 shows a comparison of our results with other best-performing papers in the literature along with the methods utilized by them. It can be observed by Table 4.6 that our model is showing the best results in terms of accuracy, recall, F_1-score and $F_{0.5}$-score. To check the model for overfitting, 10-fold cross-validation was used on the training dataset. The accuracy and recall values obtained for the training dataset are very close to the values obtained for the testing dataset. Hence the model is not showing any signs of overfitting.

To further elaborate on the results the confusion matrix for each classifier is shown in Figure 4.4. For SVM, out of 20608 non-predatory conversations 20544 were classified correctly and 64 were classified as class 1 chats. For the predatory chats out of 1730, 1614 were classified correctly and 116 were incorrectly classified as class 0 chats. For SVM, the false positives were acceptable but the false-negative instances were very low which indicates that the system is favorable to be used for public chatting platforms. As it will not generate too many false positives which could waste the efforts of LEA by tracking the people who are actually not predators. For KNN, the false negatives were high but the false positives were low which is favorable if the system needs to be more sensitive about predators and in this case, *Protectbot* will be able to uncover most of the predators.

We also collected a dataset containing 71 full predatory chats obtained from the

Table 4.6: Performance comparison of various approaches against proposed approach

Technique	Ref.	Precision	Accuracy	Recall	F_1-score	$F_{0.5}$-score
BoW with TFID weighting + NN	[139]	0.98	-	0.78	0.87	0.93
BoW with TFIDF weighting + linear SVM	[40]	-	0.98	-	-	-
Soft voting using BoW with TFIDF weighting+ SVM, BoW with binary weighting+ MNB and BoW with binary weighting+ LR	[44]	1.0	0.99	0.95	0.98	0.99
Word2vec + CNN	[45]	0.29	0.88	0.70	0.42	-
Word2Vec + classimbalance + Histogram Gradient Boosted Decision Trees	[46]	-	0.99	-	0.99	0.94
BOW + SVM + Random Forest classifier	[49]	1	-	0.82	-	0.957
CNN+ multilayer perceptron	[51]	0.46	-	0.72	-	-
BERT + Feed forward NN	[52]	0.98	-	0.99	0.98	0.98
TFIDF + SVM	[135]	0.92	0.91	0.89	0.91	0.91
One-hot CNN	[140]	0.92	-	0.72	0.81	-
Pretrained fastText + SVM	**Ours**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**

website PJ to further evaluate the performance of our classifier. The developed dataset of 71 predatory chats is also undergone the same text cleaning, pre-filtering and preprocessing stages and the word embeddings are obtained using the fastText. Using SVM classifier the model was able to identify the 48 chats correctly and 23 chats were incorrectly classified as non-predatory. Using KNN classifier the proposed model was able to correctly identify the 66 chats as predatory and incorrectly classifies 5 chats as non-predatory conversations which proves the good performance of the proposed model.

4.4 Conclusion

This Chapter provides the detailed framework of the proposed *Protectbot* and its working along with related literature. The *Protectbot* is able to classify the chats using the best-performing classifier showing the accuracy, recall, F_1 score and $F_{0.5}$ sore of 0.99. Protectbot

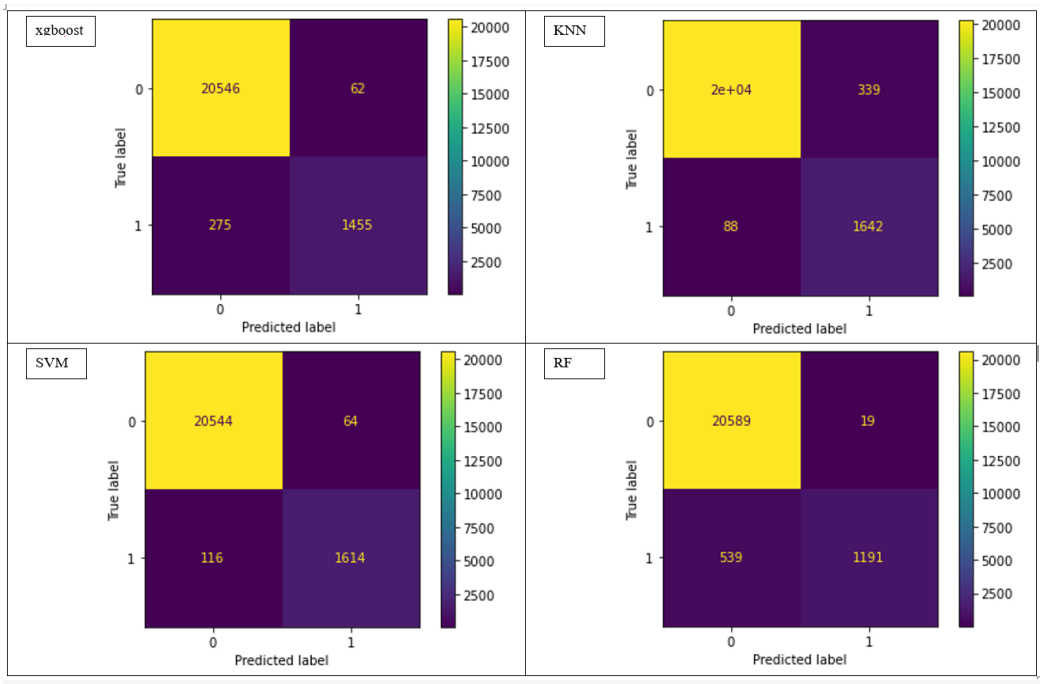

Figure 4.4: Confusion Matrix for xgboost, KNN, SVM, RF classifiers

proved to outperform state-of-the-art benchmark methods that have been applied to the publicly available dataset Pan 12. To further demonstrate the performance of the *Protectbot* the chatbot is also tested on the dataset containing 71 predatory chats obtained from PJ and the promising results are shown by the proposed model.

5. Conclusion

In this study, we conducted a systematic survey of online child protection mechanisms covering the multifaceted threats and efforts made to enhance child protection. The motivation behind this work is to highlight the existing gaps in research and presently available solutions. In our survey, we observed the following limitations:

1) A collaborative global response is lacking to protect the increased number of children connected to the Internet through web browsing, education, social media, gaming, and entertainment websites and applications. This is presumably in part due to the systemic, industrial, regulatory, and protection gaps accentuated by the absence of a global authority that ensures children's protection, a global strategy to protect children online, global clear standards, regulations and guidelines on the COP and coordination among multiple stakeholders;

2) in the last two decades, researchers were attempted to develop AI tools to enhance children's protection online, but there is no robust system or framework deployed in research or in the industry; and 3) AI tools and ML algorithms are used to detect child predatory behavior in a limited context of chat logs that only pertain to chatting platforms. Thus, the use of AI for child protection is limited, which does not provide a robust solution to the threats that children face in the gaming environment. As child protection in

this particular environment has not been sufficiently investigated, intelligent and robust mechanisms are needed to address child protection in gaming by detecting child predators and blocking them before they can approach children and harm them emotionally or physically.

After surveying the existing solutions to protect children from predators we developed an integrated platform to protect children from online predators. We developed a chatbot *Protectbot* to chat with suspects on any gaming platform. We also developed an improved text classifier based on NLP. After pre-processing the data, the word embeddings were generated using the fastText and SVM & KNN classifiers were utilized to classify the chats. To protect the privacy of children we did not save the chat logs instead the feature vectors generated by the word embedding model were saved to use these vectors in classification models. Using the developed classifier the chats were classified into two categories i.e. predatory or non-predatory. We trained and tested the text classifier on the only publicly available dataset Pan 12. The developed text classifiers showed an accuracy, recall, F_1-score and $F_{0.5}$-score of 0.99 which is better than the performance shown by proposed models in the literature. If predatory behavior is detected in chats a report is generated for LEA. We also created a new dataset containing 71 predatory chats. Using KNN classifier 66 out of 71 predatory chats are classified correctly.

5.1 Discussion

The explosion of information and communication technology (ICT) has created unprecedented opportunities for both children and young people. Both benefits and undeniable risks exist. Children are vulnerable to many kinds of threats in online gaming environments,

such as inappropriate content and predatory behavior. Inappropriate content includes adult, violent or gory content. The influence of violent games on children's mental health varies. Factors such as physical abuse, a divided family, the toxic environment at home, and predatory behavior from peers or family members are considered strong catalysts that influence the mental health of children. Predatory behavior, which includes child bullying, harassment, pedophilia, and grooming, can adversely affect children's mental health and traumatize them. The most common issues reported by victims of child predators are depression, fear, panic attacks, lack of trust in their relations, anxiety, self-harm, and suicides [141]. Child predators not only adversely influence children but also devastate their families.

The industry has made a fair contribution to enhancing the cop by adding parental controls and age rating guidance to the gaming environment. However, the effectiveness of parental controls remains controversial. Paid tools, such as NetAlert [142], NetNanny [143], and Bark [144], provide options to parents but are limited to content filtering, and thereby, they only address the content threat. It is observed that many children do not abide by the age limitation requirements placed on game registration platforms [12], [77]. The mechanisms used to determine the age of a user are not sufficiently robust to detect the actual age of the consumer [145]. Children can easily access age-inappropriate content by entering a fake age into the sign-up process. An AI-enabled facial recognition system to identify the age of the user could be a solution to this problem but it may violate children's right to privacy. A better solution is that the parents would always be aware of the content used by the children and would adopt appropriate parental control options to safeguard their children from online threats. The lack of awareness of parents is another important issue yet to be addressed. As indicated in Section V, surveys show that more than half of

the parents in the US and UK are not aware of the usage of parental controls [13], [14].

On the other hand, the available control tools have not proven to be a robust solution to overcome multitude of threats to children. Algorithmically filtered content can significantly influence child development, opinions, values, and habits. The filter creates an isolation bubble, which restricts children from exploring a wide variety of opinions and ideas [77]. To combat contact and conduct threats, parental controls can be used to restrict the children's accounts, so that they cannot participate in chats with unknown people. However, in many cases, peers are responsible for cyberbullying, defamation, or the exclusion of victims. So, by allowing children to chat with friends only does not provide protection to children against these threats. Moreover, if the chat features are fully blocked the actual essence of most of the games vanishes. More robust solutions are needed to protect against risks without compromising the quality of playing and socializing on online platforms.

Parents/caretakers' and educators' awareness of threats to children is another important aspect. To address this, governments, civil society organizations, and international multilateral are playing their roles by developing interactive user-friendly tools to enhance the knowledge of children and parents about online threats and ways to combat them. The Sango developed by ITU [73] and the social-emotional learning tool developed by UNICEF [17] are great initiatives at the international level. The interactive platform developed by the EU also helps enhance the knowledge of parents and children [18]. At the national level, governments are taking action to better protect children. For instance, the UAE government developed the kidX tool to provide children with awareness about their rights and to increase their knowledge about functions and services provided by the government to protect them online [16].

A concerted and collaborative effort to reduce the risks of the digital world, particularly

the risks targeting children and adolescents, is needed among multiple stakeholders, such as governments, civil society organizations, international multilaterals, industry, and users. Policymakers and regulators must continue striving for higher safety and protective measures to keep children safe online. We live in an ever-connected, digitized world, with children increasingly using the Internet and digital technologies for a multitude of purposes, including their learning, gaming, and social connection. Protecting children online and in digital technologies such as gaming is a global issue; however, a coordinated global response is lacking to protect the increased number of children connected to the Internet through web browsing, education, social media, gaming, and entertainment websites and applications. In recent years, the COVID-19 pandemic has perpetuated a great surge in the number of children and young people using the Internet and digital technologies. Certainly, the Internet provides opportunities for children's learning and growth; however, it also exposes them to many types of risks. This hyper-connectivity exacerbates the exposure of children to a multitude of risks, which is becoming a global phenomenon. Policymakers need to map out urgent strategies and plans aimed at tackling challenges for the protection and safety of children online. Therefore, it is imperative to establish global capacity-building programs, launch collaborative and multi-stakeholder initiatives, strategize transnational legislation and laws, and develop standards and regulations with frameworks and programs to protect children's welfare and well-being online.

The following are recommendations that are, in part, in response to systemic, industrial, regulatory, and protection gaps:

- Establish a global authority or governance body to ensure children's rights are protected from online harm.
- Develop a global strategy aligning existing international normative frameworks

for children's rights and provisions with the requisite multi-stakeholder policy frameworks coordinating industry with intergovernmental bodies such as UNICEF.

- Publish global standards, regulations, and guidelines on child online protection and safety to foster safer Internet/digital technologies/gaming for children at the industry, global, regional, national, familial, and individual levels needed to enable children's online safety and protection.
- Improve coordination among multiple stakeholders including international bodies, governments, law enforcement agencies, industry, policymakers, academics, and civil society organizations.
- Increase awareness, legislative and regulatory measures, and mechanisms at strategic, tactical, and operational levels.
- Provide opportunities to improve innovations and build capacity and capability in child online protection and safety literacy, training, re-skilling, and upskilling, as well as children's ability to protect themselves.

Over the past two decades, research to explore the use of AI tools and ML algorithms for children's online safety has increased. However, the main focus of this research remains restricted to the detection of predatory behavior in chat logs. This solution is not enough to protect the child from predators, as it is only a step toward building a mechanism to protect children online. It is also observed that the focus of research on detecting predatory behavior is on chat logs pertaining to social media's chatting platforms [122–126]. However, gaming platforms have rarely been explored. The main reason for this research gap is the unavailability of public datasets of online game players' chat logs. Owing to privacy issues, the data are not stored or saved, which makes it difficult to build a dataset. We recommend that the industry and researchers collect data on gaming

platforms after obtaining the consent of the parents and children to share their chat logs to build the required datasets. Distributed platforms chatbots have been developed to detect child predators on chatting platforms [29], [36], [42]. However, owing to the limited availability of relevant datasets, bots have not been tested on balanced and large datasets.

In this thesis, a chatbot *Protectbot* is designed to consider the recommendations by stakeholders to combat various threats including sexual predation and cyberbullying without compromising the rights of children to play, leisure and culture (Article 31). *Protectbot* is able to detect child predators on a live chatting platform and report the predators before they could cause any harm.

5.2 Future Work

Child protection on a gaming platform is a multifaced problem. The proposed solution needs improvement and enhancements in many ways. The following are the improvements that we are planning to implement in the next phases of this project:

- The Pan 12 dataset is an unbalanced dataset with the training dataset containing only 2016 predatory conversations. In order to build a better classifier that could be integrated with the live network, the classifier should be trained on a dataset containing more instances of predatory chat so it could better detect the predators on a gaming platform. In the next phase of this project, we want to build a new dataset containing more predatory chats obtained on gaming platforms. The new dataset would be used to train and test the Protectbot.
- In the upcoming phases of this project, we would like to integrate *Protectbot* on the gaming platform to protect the children on gaming platforms.

- We also aim to develop better content filtering methods that could be integrated with the gaming platforms to provide age-appropriate content to the children using the platforms. These filtering methods should be approved by all the stakeholders and must be added as the requirement from the LEA as a minimum standard of child protection.
- We also intend to develop an age detection mechanism that could be integrated with the gaming platforms to identify the people who are behaving as a child to build bonds with other children. Many gaming platforms require to meet minimum age requirements. The age detection tool can also be utilized to detect if a child is using his/her actual age to join any platform.

Bibliography

[1] S. Kemp, "Digital 2021," Kepios Pte.Ltd, Singapore, Jan. 2021., (Accessed: Jan. 5, 2022). [Online]. Available: https://datareportal.com/reports/digital-2021-global-overview-report

[2] P. Stalker, S. Livingstone, D. Kardefelt-Winther, and M. Saeed, "Growing up in a connected world," UNICEF Office of Res., Innocenti, Florence, Itlay, Nov. 2019.

[3] T. Weru, J. Sevilla, J. Olukuru, L. Mutegi, and T. Mberi, "Cyber-smart children, cyber-safe teenagers: Enhancing internet safety for children," in *2017 IST-Africa Week Conference (IST-Africa)*, Windhoek, Namibia, May 2017, pp. 1–8, doi: 10.23919/ISTAFRICA.2017.8102292.

[4] American Psychological Association, "Resolution on violent video games," APA, Washington, DC, USA, Oct . 2019, (Accessed: Dec. 8, 2021). [Online]. Available: http://www.apa.org/about/policy/violent-video-games.aspx

[5] T. Nuangjumnong, "The effects of gameplay on leadership behaviors: An empirical study on leadership behaviors and roles in multiplayer online battle arena games," in *2014 Int. Conf. Cyberworlds*, Santander, Spain, Oct. 2014, pp. 300–307, doi: 10.1109/CW.2014.48.

[6] N. Pobiedina, J. Neidhardt, M. d. C. Calatrava Moreno, L. Grad-Gyenge, and H. Werthner, "On successful team formation: Statistical analysis of a multiplayer online game," in *2013 IEEE 15th Conf. Business Informatics*, Vienna, Austria, Jul. 2013, pp. 55–62, doi: 10.1109/CBI.2013.17.

[7] M. H. Hussein, S. H. Ow, L. S. Cheong, M.-K. Thong, and N. A. Ebrahim, "Effects of digital game-based learning on elementary science learning: A systematic review," *IEEE Access*, vol. 7, pp. 62 465–62 478, May 2019, doi:10.1109/ACCESS.2019.2916324.

[8] J. S. Kinnebrew, S. S. Killingsworth, D. B. Clark, G. Biswas, P. Sengupta, J. Minstrell, M. Martinez-Garza, and K. Krinks, "Contextual markup and mining in digi-

tal games for science learning: Connecting player behaviors to learning goals," *IEEE Trans. on Learn. Technol.*, vol. 10, no. 1, pp. 93–103, Jan. 2016, doi: 10.1109/TLT.2016.2521372.

[9] O. Dele-Ajayi, J. Sanderson, R. Strachan, and A. Pickard, "Learning mathematics through serious games: An engagement framework," in *2016 IEEE Frontiers in Education Conference (FIE)*, Oct. 2016, pp. 1–5, doi: 10.1109/FIE.2016.7757401.

[10] C.-H. Ko, J.-Y. Yen, C.-S. Chen, Y.-C. Yeh, and C.-F. Yen, "Predictive values of psychiatric symptoms for internet addiction in adolescents: a 2-year prospective study," *Arch. Pediatr. Adolesc. Med.*, vol. 163, no. 10, pp. 937–943, Oct. 2009, doi: 10.1001/archpediatrics.2009.159.

[11] J.-L. Wang, J.-R. Sheng, and H.-Z. Wang, "The association between mobile game addiction and depression, social anxiety, and loneliness," *Frontiers in public health*, vol. 7, p. 247, Sep. 2019, doi: 10.3389/fpubh.2019.00247.

[12] UNICEF, "Child rights and online gaming: Opportunities & challenges for children and the industry," UNICEF Office of Res., Innocenti, Florence, Itlay, Aug. 2019.

[13] Statista Research Department, "Percentage of parents placing limits on children's media consumption in the united states in 2019 and 2020, by medium," statista.com., (Accessed: Dec. 5, 2021). [Online]. Available: https://www.statista.com/statistics/232345/parental-control-over-childrens-media-consumption-in-the-us/#:~:text=Parental%20control%20over%20children%27s%20media%20consumption%20U.S.%202019%2D2020&text=A%202020%20study%20revealed%20that,slightly%20from%20the%20previous%20year

[14] I. Taylor, "Ukie: Only 19% of parents set and enforce screen time limits for their children," gamesindustry.biz., (Accessed: Dec. 5, 2021). [Online]. Available: https://www.gamesindustry.biz/articles/2018-09-14-digital-school-house-only-19-percent-of-parents-set-and-enforce-screen-time-limits-for-their-ch

[15] International Telecommunication Union, "Keeping children safe online," itu.com., (Accessed: May 9, 2022). [Online]. Available: https://www.itu-cop-guidelines.com/

[16] The United Arab Emirates' Government portal, "Gamification," u.ae., (Accessed: May 16, 2022). [Online]. Available: https://u.ae/en/about-the-uae/digital-uae/gamification

[17] "Tilli: gamified social-emotional learning for child online safety," unicef.org., (Accessed: May 16, 2022). [Online]. Available: https://gdc.unicef.org/resource/tilli-gamified-social-emotional-learning-child-online-safety

[18] European Commission, "Enhancing professionals' capacity to deal with child victims," childprotect.eu., (Accessed: May 16, 2022). [Online]. Available: http://childprotect.eu/

[19] E. Lebedeva and J. A. Brown, "Companion AI for starbound game using utility theory," in *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, Innopolis, Russia, Dec. 2020, pp. 1–5, doi:10.1109/NIR50484.2020.9290164.

[20] I. Oh, S. Rho, S. Moon, S. Son, H. Lee, and J. Chung, "Creating pro-level ai for a real-time fighting game using deep reinforcement learning," *IEEE Trans. Games*, vol. 14, no. 2, pp. 212–220, Jun. 2022, doi: 10.1109/TG.2021.3049539.

[21] H. Baier, A. Sattaur, E. J. Powley, S. Devlin, J. Rollason, and P. I. Cowling, "Emulating human play in a leading mobile card game," *IEEE Trans. Games*, vol. 11, no. 4, pp. 386–395, Dec. 2019, doi: 10.1109/TG.2018.2835764.

[22] S. Ariyurek, A. Betin-Can, and E. Surer, "Automated video game testing using synthetic and humanlike agents," *IEEE Trans. Games*, vol. 13, no. 1, pp. 50–67, Mar. 2021, dOI: 10.1109/TG.2019.2947597.

[23] M. Ishihara, S. Ito, R. Ishii, T. Harada, and R. Thawonmas, "Monte-carlo tree search for implementation of dynamic difficulty adjustment fighting game ais having believable behaviors," in *2018 IEEE Conf. Computational Intelligence and Games (CIG)*, Maastricht, Netherlands, Aug. 2018, pp. 1–8, doi : 10.1109/CIG.2018.8490376.

[24] W. Konen, "General board game playing for education and research in generic ai game learning," in *2019 IEEE Conf. on Games (CoG)*, London, UK, Aug. 2019, pp. 1–8, doi: 10.1109/CIG.2019.8848070.

[25] Y.-G. Cheong, A. K. Jensen, E. R. Guðnadóttir, B.-C. Bae, and J. Togelius, "Detecting predatory behavior in game chats," *IEEE Trans. Computational Intelligence and AI in Games*, vol. 7, no. 3, pp. 220–232, Sep. 2015, doi:10.1109/TCIAIG.2015.2424932.

[26] M. Märtens, S. Shen, A. Iosup, and F. Kuipers, "Toxicity detection in multiplayer online games," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, Zagreb, Croatia,, Dec. 2015, pp. 1–6, doi: 10.1109/NetGames.2015.7382991.

[27] S. Murnion, W. J. Buchanan, A. Smales, and G. Russell, "Machine learning and semantic analysis of in-game chat for cyberbullying," *Computers & Security*, vol. 76, pp. 197–213, Jul. 2018.

[28] J. A. Cornel, C. Christian Pablo, J. A. Marzan, V. Julius Mercado, B. Fabito, R. Rodriguez, M. Octaviano, N. Oco, and A. D. La Cruz, "Cyberbullying detection for online games chat logs using deep learning," in *2019 IEEE 11th Int. Conf. Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Laoag, Philippines, Nov. 2019, pp. 1–5, doi: 10.1109/HNICEM48295.2019.9072811.

[29] C. Laorden, P. Galán-García, I. Santos, B. Sanz, J. M. G. Hidalgo, and P. G. Bringas, "Negobot: A conversational agent based on game theory for the detection of paedophile behaviour," in *Int. Joint Conf. CISIS'12-ICEUTE 12-SOCO 12 Special Sessions*, vol. 189, Berlin, Heidelberg, Jan. 2013, pp. 261–270.

[30] P. Zambrano, M. Sanchez, J. Torres, and W. Fuertes, "Bothook: An option against cyberpedophilia," in *2017 1st Cyber Security in Networking Conference (CSNet)*, Rio de Janeiro, Brazil, Oct. 2017, pp. 1–3, doi: 10.1109/CSNET.2017.8241994.

[31] A. H. Alduailej and M. B. Khan, "The challenge of cyberbullying and its automatic detection in arabic text," in *2017 Int. Conf. Computer and Applications (ICCA)*, Doha, Qatar, Sep. 2017, pp. 389–394, doi: 10.1109/COMAPP.2017.8079791.

[32] F. E. Gunawan, L. Ashianti, and N. Sekishita, "A simple classifier for detecting online child grooming conversation," *Telecommunication Computing Electronics and Control*, vol. 16, no. 3, pp. 1239–1248, Jun. 2018.

[33] Z. Zuo, J. Li, P. Anderson, L. Yang, and N. Naik, "Grooming detection using fuzzy-rough feature selection and text classification," in *2018 IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8, doi: 10.1109/FUZZ-IEEE.2018.8491591.

[34] P. Anderson, Z. Zuo, L. Yang, and Y. Qu, "An intelligent online grooming detection system using ai technologies," in *2019 IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA, Jun. 2019, pp. 1–6, doi:10.1109/FUZZ-IEEE.2019.8858973.

[35] Z. Zuo, J. Li, B. Wei, L. Yang, F. Chao, and N. Naik, "Adaptive activation function generation for artificial neural networks through fuzzy inference with application in grooming text categorisation," in *2019 IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA,, Jun. 2019, pp. 1–6, doi: 10.1109/FUZZ-IEEE.2019.8858838.

[36] J. Murcia Triviño, S. Moreno Rodríguez, D. O. Díaz López, and F. Gómez Mármol, "C3-sex: A chatbot to chase cyber perverts," in *2019 IEEE Intl. Conf. Dependable, Autonomic and Secure Computing, Intl. Conf. Pervasive Intelligence and Computing,*

Intl. Conf. Cloud and Big Data Computing, Intl. Conf. Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, Aug. 2019, pp. 50–57, doi: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00024.

[37] P. Bours and H. Kulsrud, "Detection of cyber grooming in online conversation," in *2019 IEEE Int. Workshop on Information Forensics and Security (WIFS)*, Delft, Netherlands, Dec. 2019, pp. 1–6, doi: 10.1109/WIFS47025.2019.9035090.

[38] N. R. Sulaiman and M. M. Siraj, "Classification of online grooming on chat logs using two term weighting schemes," *Int. J. Innov. Comp.*, vol. 9, no. 2, pp. 43–50, Nov 2019.

[39] T. R. Ringenberg, K. Misra, and J. T. Rayz, "Not so cute but fuzzy: Estimating risk of sexual predation in online conversations," in *2019 IEEE Int. Conf. Systems, Man and Cybernetics (SMC)*, Bari, Italy, Oct. 2019, pp. 2946–2951, doi: 10.1109/SMC.2019.8914528.

[40] P. R. Borj and P. Bours, "Predatory conversation detection," in *2019 Int. Conf. Cyber Security for Emerging Technologies (CSET)*, Doha, Qatar, Oct. 2019, pp. 1–6, doi: 10.1109/CSET.2019.8904885.

[41] S. Andleeb, R. Ahmed, Z. Ahmed, and M. Kanwal, "Identification and classification of cybercrimes using text mining technique," in *2019 Int. Conf. Frontiers of Information Technology (FIT)*, Doha, Qatar, Dec. 2019, pp. 227–2275, doi: 10.1109/FIT47737.2019.00050.

[42] J. I. Rodríguez, S. R. Durán, D. Díaz-López, J. Pastor-Galindo, and F. G. Mármol, "C3-sex: A conversational agent to detect online sex offenders," *Electronics*, vol. 9, no. 11, p. 1779, Oct. 2020, doi: 10.3390/electronics9111779.

[43] J. Kim, Y. J. Kim, M. Behzadi, and I. G. Harris, "Analysis of online conversations to detect cyberpredators using recurrent neural networks," in *Proc. 1st Int. Workshop on Social Threats in Online Conversations: Understanding and Management*, Marseille, France, May 2020, pp. 15–20.

[44] M. A. Fauzi and P. Bours, "Ensemble method for sexual predators identification in online chats," in *2020 8th int. Workshop on Biometrics and Forensics (IWBF)*, Porto, Portugal, Apr. 2020, pp. 1–6, doi: 10.1109/I BF49977.2020.9107945.

[45] G. Isaza, F. Muñoz, L. Castillo, and F. Buitrago, "Classifying cybergrooming for child online protection using hybrid machine learning model," *Neurocomputing*, vol. 484, pp. 250–259, May 2022, doi:10.1016/j.neucom.2021.08.148.

[46] P. R. Borj, K. Raja, and P. Bours, "Detecting sexual predatory chats by perturbed data and balanced ensembles," in *2021 Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, Sep. 2021, pp. 1–5, doi: 10.1109/BIOSIG52210.2021.9548303.3.

[47] C. H. Ngejane, J. H. Eloff, T. J. Sefara, and V. N. Marivate, "Digital forensics supported by machine learning for the detection of online sexual predatory chats," *Forensic Science Int.: Digital Investigation*, vol. 36, p. 301109, Mar. 2021, doi: 10.1016/j.fsidi.2021.301109.

[48] K. S. Kirupalini, A. Baskar, A. Ramesh, G. Rengarajan, S. Gowri, S. Swetha, and D. Sangeetha, "Prevention of emotional entrapment of children on social media," in *2021 Int. Conf. Emerging Techniques in Computational Intelligence (ICETCI)*, Hyderabad, India, Aug. 2021, pp. 95–100, doi:10.1109/ICETCI51973.2021.9574068.

[49] Y. Singla, "Detecting sexually predatory behavior on open-access online forums," in *Proc. Res. and Applications in AI, Advances in Intelligent Systems and Computing*, Kolkata, India, Jun. 2021, vol. 1355, pp. 27–40, doi: 10.1007/978-981-16-1543-6_3.

[50] M. A. Wani, N. Agarwal, and P. Bours, "Sexual-predator detection system based on social behavior biometric (SSB) features," *Procedia Computer Science*, vol. 189, pp. 116–127, May 2021.

[51] S. Preuß, L. P. Bley, T. Bayha, V. Dehne, A. Jordan, S. Reimann, F. Roberto, J. R. Zahm, H. Siewerts, D. Labudde *et al.*, "Automatically identifying online grooming chats using cnn-based feature extraction," in *Pro. 17th Conf. Natural Language Processing (KONVENS 2021)*, Dusseldorf, Germany, Sep. 2021, pp. 137–146.

[52] N. Agarwal, T. Ünlü, M. A. Wani, and P. Bours, "Predatory conversation detection using transfer learning approach," in *7th Int. Conf. Machine Learning, Optimization, and Data Science*, vol. 13163, Grasmere, United Kingdom, Oct. 2022, pp. 488–499, doi: 10.1007/978-3-030-95467-3_35.

[53] G. Inches and F. Crestani, "Overview of the international sexual predator identification competition at pan-2012," in *CLEF (Online working notes/labs/workshop)*, vol. 30, Sep. 2012.

[54] A. Faraz, J. Mounsef, A. Raza, and S. Willis, "Child safety and protection in the online gaming ecosystem," *IEEE Access*, vol. 10, pp. 115 895–115 913, Oct. 2022, doi: 10.1109/ACCESS.2022.3218415.

[55] E. Staksrud and S. Livingstone, "Children and online risk: Powerless victims or resourceful participants?" *Information, Communication & Society*, vol. 12, no. 3, pp. 364–387, Mar. 2009, doi: 10.1080/13691180802635455.

[56] M. Stoilova, S. Livingstone, and R. Stoilova, "Investigating risks and opportunities for children in a digital world: A rapid review of the evidence on children's internet use and outcomes," UNICEF Office of Res., Innocenti, Florence, Itlay, Feb. 2021.

[57] P. A. Chan and T. Rabinowitz, "A cross-sectional analysis of video games and attention deficit hyperactivity disorder symptoms in adolescents," *Annals of General Psychiatry*, vol. 5, no. 1, pp. 1–10, Oct. 2006, doi: 10.1186/1744-859X-5-16.

[58] M. B. Mathur and T. J. VanderWeele, "Finding common ground in meta-analysis wars on violent video games," *Perspectives on psychological science*, vol. 14, no. 4, pp. 705–708, 2019, doi: 10.1177/1745691619850104.

[59] Z. Hussain and M. D. Griffiths, "Excessive use of massively multi-player online role-playing games: A pilot study," *Int. J. Ment. Health Addiction*, vol. 7, no. 4, pp. 563–571, Feb. 2009, doi: 10.1007/s11469-009-9202-8.

[60] C. J. Ferguson, "Do angry birds make for angry children? A meta-analysis of video game influences on children's and adolescents' aggression, mental health, prosocial behavior, and academic performance," *Perspect Psychol Sci.*, vol. 10, no. 5, pp. 646–666, Sep. 2015, doi: 10.1177/1745691615592234.

[61] S. Domingues-Montanari, "Clinical and psychological effects of excessive screen time on children," *J. of paediatr. Child Health*, vol. 53, no. 4, pp. 333–338, Feb. 2017, doi : 10.1111/jpc.13462.

[62] D. Richards, P. H. Caldwell, and H. Go, "Impact of social media on the health of children and young people," *J. Paediatr. Child Health*, vol. 51, no. 12, pp. 1152–1157, Nov. 2015, doi: 10.1111/jpc.13023.

[63] V. Bell, "Online information, extreme communities and internet therapy: Is the internet good for our mental health?" *J. Mental Health*, vol. 16, no. 4, pp. 445–457, Jul. 2007, doi: 10.1080/09638230701482378.

[64] V. Suchert, R. Hanewinkel, and B. Isensee, "Sedentary behavior and indicators of mental health in school-aged children and adolescents: A systematic review," *Preventive Medicine*, vol. 76, pp. 48–57, Jul. 2015, doi: 10.1016/j.ypmed.2015.03.026.

[65] R. Armitage, "Bullying in children: impact on child health," *BMJ paediatrics*, vol. 5, no. 1, Mar. 2021, doi: 10.1136/ bmjpo-2020-000939.

[66] M. V. Geel, P. Vedder, and J. Tanilon, "Relationship between peer victimization, cyberbullying, and suicide in children and adolescents: A meta-analysis," *JAMA pediatrics*, vol. 168, no. 5, pp. 435–442, May 2014, doi: 10.1001/jamapediatrics.2013.4143.

[67] The New York Times., "Video games and online chats are hunting grounds for sexual predators," nytimes.com., (Accessed: May 9, 2022). [Online]. Available: https://www.nytimes.com/interactive/2019/12/07/us/video-games-child-sex-abuse.html

[68] J. Campbell and J. Kravarik, "A 17-year-old boy died by suicide hours after being scammed. The FBI says it's part of a troubling increase in sextortion cases," cnn.com., (Accessed: May 9, 2022). [Online]. Available: https://edition.cnn.com/2022/05/20/us/ryan-last-suicide-sextortion-california/index.html

[69] M. K. Khan, O. Bamasag, A. A. Algarni, and M. Alqarni, "Policy brief: Fostering a safer cyberspace for children," Think 20 Engagement Group, Dec. 2020.

[70] X. Zhang, "Charging children with child pornography - using the legal system to handle the problem of sexting," *Computer Law & Security Review*, vol. 26, no. 3, pp. 251–259, May 2010, doi: 10.1016/j.clsr.2010.03.005.

[71] C. Doyle, E. Douglas, and G. O'Reilly, "The outcomes of sexting for children and adolescents: A systematic review of the literature," *J. Adolescence*, vol. 92, pp. 86–113, Oct. 2021.

[72] Code of Federal Regulations, "Part 312- children's online privacy protection rule," ecfr.gov., (Accessed: Jun. 9, 2022). [Online]. Available: https://www.ecfr.gov/current/title-16/part-312

[73] International Telecommunication Union (ITU)., "Guidelines for children," itu.com., (Accessed: May 9, 2022). [Online]. Available: https://www.itu-cop-guidelines.com/children

[74] International Telecommunication Union (ITU)., "Guidelines for parents and educators on child online protection 2020," ITU, Place des Nations, Geneva, Switzerland, 2020.

[75] International Telecommunication Union (ITU)., "Guidelines for industry on child online protection 2020," ITU, Place des Nations, Geneva, Switzerland, 2020.

[76] European Commission, "Self-regulation for a better internet for kids," ec.europa.eu., (Accessed: Jun. 9, 2022). [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/self-regulation-better-internet-kids

[77] International Telecommunication Union (ITU)., "Guidelines for policy-makers on child online protection 2020," ITU, Place des Nations, Geneva, Switzerland, 2020.

[78] W. M. Taibah, H. K. Khalifa, and A. M. Alshebaiki, "Strengthening the convention on the rights of the child (CRC) governing children's digital world," Think 20 engagement group, Dec. 2020.

[79] International Age Rating Coalition, "How iarc works," globalrating.com., (Accessed: Apr. 4, 2022). [Online]. Available: https://www.globalratings.com/how-iarc-works.aspx

[80] Australian Classification, "Helping you choose what to watch and play," classification.gov.au., (Accessed: Apr. 5, 2022). [Online]. Available: https://www.classification.gov.au/

[81] Ministry of Justice and Public Security, "What is the rating system?," gov.br., (Accessed: Apr. 5, 2022). [Online]. Available: https://www.gov.br/mj/pt-br/assuntos/seus-direitos/classificacao-1

[82] Gaming Rating & Administration Committee, "Age rating symbol," grac.or.kr., (Accessed: Apr. 6, 2022). [Online]. Available: https://www.grac.or.kr/english/

[83] Entertainment Software Rating Board, "Tools for parents," esrb.org., (Accessed: Apr. 6, 2022). [Online]. Available: https://www.esrb.org/

[84] Pan European Game Information, "Pegi helps parents to make informed decision when buying video games," pegi.info., (Accessed: Apr. 6, 2022). [Online]. Available: https://pegi.info/

[85] Unterhaltungssoftware Selbstkontrolle, "Classification of games and apps," usk.de., (Accessed: Apr. 8, 2022). [Online]. Available: https://usk.de/

[86] Entertainment Software Association, "2021 Essential facts about the video game industry," ESA, US, Jul. 2021.

[87] "For parents," roblox.com., (Accessed: Jan. 3, 2022). [Online]. Available: https://corp.roblox.com/parents/

[88] "Family view," steampowered.com, (Accessed: Jan. 3, 2022). [Online]. Available: https://help.steampowered.com/en/

[89] "How to set parental controls on PS4 consoles," playstation.com., (Accessed: Jan. 10, 2022). [Online]. Available: https://www.playstation.com/en-ae/support/account/ps4-parental-controls-and-spending-limits/

[90] "Gaming that is safer for all," xbox.com., (Accessed: Jan. 10, 2022). [Online]. Available: https://www.xbox.com/en-US/community/for-everyone/responsible-gaming

[91] "Nintendo switch parental controls mobile app," nintendo.com., (Accessed: Jan. 10, 2022). [Online]. Available: https://www.nintendo.com/switch/parental-controls/

[92] UNICEF, "Case study: Millicom's impact assessment," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: https://sites.unicef.org/csr/files/MILLICOM_casestudy.pdf

[93] UNICEF, "Case study: Microsoft's photo dna," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: https://sites.unicef.org/csr/files/MICROSOFT_casestudy.pdf

[94] UNICEF, "Case study: Supporting national governments to develop child online protection-related national action plans," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: Available:https://sites.unicef.org/csr/files/Case_study_Microsoft.pdf

[95] UNICEF, "Case study: Safaricom: Integrating children's rights into core business," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: Available:https://sites.unicef.org/csr/files/Case_study_Safaricom.pdf

[96] UNICEF, "Case study: Amigos Conectados project by the Walt Disney company latin America and chicos.net," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: Available:https://sites.unicef.org/csr/files/Case_study_Disney.pdf

[97] UNICEF, "Case study: Strengthening technology companies practices' to fight child sexual exploitation on their platforms," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: Available:https://sites.unicef.org/csr/files/Case_study_Thorn_Sound_Practices_Guide.pdf

[98] UNICEF, "Case study: LEGO supplier training through the LEGO academy in India," unicef.org., (Accessed: Jun. 5, 2022). [Online]. Available: Available:https://sites.unicef.org/csr/files/LEGO_supplier_training.pdf

[99] K. R. Bokka, S. Hora, T. Jain, and M. Wambugu, *Deep Learning for Natural Language Processing: Solve your natural language processing problems with smart deep neural networks*. Birmingham, UK: Packt Publishing Ltd, 2019.

[100] A. Kao and S. R. Poteet, *Natural language processing and text mining*. Springer Science & Business Media, Mar. 2007.

[101] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical natural language processing: A comprehensive guide to building real-world NLP systems*, 1st ed. Sevastopol, CA, US: O'Reilly Media, Jun. 2020.

[102] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, Dec 2017.

[103] E. M. Dharma, F. L. Gaol, H. Leslie, H. Warnars, and B. Soewito, "The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification," *J Theor Appl Inf Technol*, vol. 100, no. 2, p. 31, Jan. 2022.

[104] T. Mitchell, "20 newsgroups," The UCI KDD Archive Information and Computer Science University of California, Irvine, (Accessed: Nov. 5, 2022). [Online]. Available: Available:https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html

[105] H. N. Nguyen, S. Teerakanok, A. Inomata, and T. Uehara, "The comparison of word embedding techniques in rnns for vulnerability detection." in *ICISSP*, 2021, pp. 109–120.

[106] A. Jung, *Machine Learning: The Basics*. Gateway East, Singapore: Springer Nature, 2022.

[107] "What is the k-nearest neighbors algorithm?" ibm.com, (Accessed: Nov. 4 , 2022). [Online]. Available: Available:https://www.ibm.com/ae-en/topics/knn

[108] "Naive bayes classifiers," geeksforgeeks.org, (Accessed: Nov. 15, 2022). [Online]. Available: Available:https://www.geeksforgeeks.org/naive-bayes-classifiers/

[109] "Machine learning," cs50.harvard.edu, (Accessed: Nov. 20 , 2022). [Online]. Available: Available:https://cs50.harvard.edu/ai/2020/notes/4/

[110] S. Ray, "Understanding support vector machine(svm) algorithm from examples (along with code)," analyticsvidhya.com, (Accessed: Nov. 20 , 2022). [Online]. Available: Available:https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[111] "What is a decision tree?" ibm.com, (Accessed: Nov. 15, 2022). [Online]. Available: Available:https://www.ibm.com/ae-en/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.

[112] P. Gupta, "Decision trees in machine learning," towardsdatascience.com, (Accessed: Nov. 15, 2022). [Online]. Available: Available:https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

[113] A. Dutta, "Random forest regression in python," geeksforgeeks.org, (Accessed: Nov. 15, 2022). [Online]. Available: Available:https://www.geeksforgeeks.org/random-forest-regression-in-python/

[114] "Xgboost," geeksforgeeks.org, (Accessed: Nov. 15, 2022). [Online]. Available: Available:https://www.geeksforgeeks.org/xgboost/

[115] "Neural networks," cs50.harvard.edu, (Accessed: Oct. 20 , 2022). [Online]. Available: Available:https://cs50.harvard.edu/ai/2020/notes/5/

[116] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, Dec. 2020.

[117] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, Dec. 2017.

[118] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," *arXiv preprint arXiv:1911.00536*, Nov. 2019.

[119] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, Feb. 2019.

[120] "What are the types of apis and their differences?" techtarget.com, (Accessed: Oct. 2, 2022). [Online]. Available: Available:https://www.techtarget.com/searchapparchitecture/tip/What-are-the-types-of-APIs-and-their-differences#:~:text=There%20are%20four%20principal%20types,any%20outside%20developer%20or%20business.

[121] A. Trivedi, V. Gor, and Z. Thakkar, "Chatbot generation and integration: A review," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 5, no. 2, pp. 1308–1311, 2019.

[122] M. Fire, R. Goldschmidt, and Y. Elovici, "Online social networks: threats and solutions," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 4, pp. 2019–2036, May 2014, doi: 10.1109/COMST.2014.2321628.

[123] M. A. Al-Garadi, M. R. Hussain, N. Khan, G. Murtaza, H. F. Nweke, I. Ali, G. Mujtaba, H. Chiroma, H. A. Khattak, and A. Gani, "Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges," *IEEE Access*, vol. 7, pp. 70 701–70 718, May 2019, doi: 10.1109/ACCESS.2019.2918354.

[124] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: A hybrid deep learning approach for cyberbullying detection in twitter social media platform," *IEEE Access*, vol. 10, pp. 25 857–25 871, Feb. 2022, doi: 10.1109/ACCESS.2022.3153675.

[125] A. Jevremovic, M. Veinovic, M. Cabarkapa, M. Krstic, I. Chorbev, I. Dimitrovski, N. Garcia, N. Pombo, and M. Stojmenovic, "Keeping children safe online with limited resources: Analyzing what is seen and heard," *IEEE Access*, vol. 9, pp. 132 723–132 732, Sep. 2021, doi: 10.1109/ACCESS.2021.3114389.

[126] J.-I. Martínez-de Morentin, A. Lareki, and J. Altuna, "Risks associated with posting content on the social media," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 16, no. 1, pp. 77–83, Feb. 2021, dOI: 10.1109/RITA.2021.3052655.

[127] Perverted Justice Foundation, "Perverted-justice.com archives," perverted-justice.com, (Accessed: Dec. 5, 2021). [Online]. Available: http://www.perverted-justice.com/

[128] inportb.com., (Accessed: Jan. 5, 2022). [Online]. Available: http://inportb.com

[129] literotica.com., (Accessed: May. 5, 2022). [Online]. Available: http://literotica.com

[130] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, and G. Inches, "Overview of the author profiling task at pan 2013," in *CLEF 2013 evaluation labs and workshop – Working Notes Papers*, Valencia, Spain, 2013, pp. 352–365.

[131] "Kick ass open web technologies irc logs," krijnhoetmer.nl., (Accessed: Jan. 5, 2022). [Online]. Available: http://krijnhoetmer.nl/irc-logs/

[132] myspace.com., (Accessed: Jan. 5, 2022). [Online]. Available: https://myspace.com/

[133] "Abusive and sexual conversations between adults," fugly.com., (Accessed: May. 9, 2022). [Online]. Available: https://www.fugly.com/victims/

[134] "Internet relay chat," netsplit.de., (Accessed: Jan. 5, 2022). [Online]. Available: http://netsplit.de

[135] P. R. Borj, K. Raja, and P. Bours, "On preprocessing the data for improving sexual predator detection : Anonymous for review," in *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA*, Nov. 2020, pp. 1–6.

[136] G. Inc., "Computing numeric representation of words in high dimensional space," Patent US 9,037,'464 B1, May, 2015.

[137] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Oct. 2014, pp. 1532–1543.

[138] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[139] E. Villatoro-Tello, A. Juárez-González, H. J. Escalante, M. Montes-y Gómez, and L. V. Pineda, "A two-step approach for effective detection of misbehaving users in chats." in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 1178, Sep. 2012.

[140] M. Ebrahimi, C. Y. Suen, and O. Ormandjieva, "Detecting predatory conversations in social media by deep convolutional neural networks," *Digital Investigation*, vol. 18, pp. 33–49, Sep. 2016.

[141] C. M. Arata, J. Langhinrichsen-Rohling, D. Bowers, and L. O'Farrill-Swails, "Single versus multi-type maltreatment: An examination of the long-term effects of child abuse," *J. of Aggression, Maltreatment & Trauma*, vol. 11, no. 4, pp. 29–52, Aug. 2005, doi: 10.1300/J146v11n04_02.

[142] "Net alert," netalert.me., (Accessed: Jan. 5, 2022). [Online]. Available: https://netalert.me/

[143] "Net nanny," netnanny.com., (Accessed: Jan. 5, 2022). [Online]. Available: https://www.netnanny.com/

[144] "bark.us." bark.us., (Accessed: Jan. 5, 2022). [Online]. Available: https://www.bark.us/

[145] M. Pietikainen, "Recommendations for the online gaming industry on assessing impact on children gaming & the rights of the child," UNICEF, United Nation, Apr. 2020.

Acronyms

AI Artificial Intelligence. 3–6, 22, 32, 74, 91, 92
AIML Artificial Intelligence Markup Language. 75, 76
ANN Artificial neural networks. 69, 76, 77, 82
API Application Programming Interface. 87–90

BERT Bidirectional Encoder Representations from Transformers. 97
BoW Bag of words. 38–41, 97, 104

CBOW Continuous bag of words. 44, 47
CFR Code of Federal Regulations. 13
CNN Convolution Neural Network. 95
COP child online protection. 7, 16, 19–22
CSAM Child Sexual Abuse Material. 20

DAN deep artificial networks. 95
DialoGPT Dialogue Generative pre-trained Transformer. 4, 86, 87, 98
DL Deep Learning. 32
DT Decision Tree. 63, 65

eCFR electronic code of federal regulation. 13, 14
ESA Entertainment Software Association. 24
ESRB Entertainment Software Rating Board. 23, 24
EU European Union. 22

FRFS fuzzy rough feature selection. 94

GNB Gaussian Naïve Bayes. 94
GPT2 Generative Pretrained Transformer. 4
GRAC Game Rating and Administration Committee. 23

IARC International Age Rating Coalition. 23, 24
ICT information and communication technology. 15, 20, 22

IDF Inverse Document Frequency. 40, 41
ITU International Telecommunication Union. 7, 16, 17, 19–22

KNN K- nearest neighbour. v, 52, 54, 105, 107, 110

LEA Law Enforcement Agencies. 4, 7, 12, 91, 99, 106, 110, 116
LR Logistic Regression. 94, 95
LSA Latent semantic analysis. 76
LSTM long short-term memory. 95

ML Machine learning. 6, 7, 32, 34, 35, 39, 41, 42, 66, 74, 76, 78, 91, 95
MLP multilayer perceptron. 95
MNB Multinomial Naive Bayes. 58, 93, 107

NB Naive Bayes. 55, 58, 95, 97
NCA National Cybersecurity Authority. 22
NLG Natural Language Generator. 77, 81
NLP Natural Language Processing. 5, 7, 31–36, 38, 39, 41, 74, 76, 103, 110
NLU Natural language understanding. 77, 78
NN Neural Network. 69–71, 76, 82, 95, 96

PEGI Pan European Game Information. 23
PJ Perverted Justice. v, 5, 92, 95, 99, 100, 105, 107, 108
PN power normalization. 94

RF Random Forest. x, 66, 67, 94, 95, 97, 105
RNN Recurrent Neural Network. 82–84

SVM Support Vector Machine. x, 59–63, 93, 95, 97, 105–107, 110

TF Term Frequency. 40, 41
TF-IDF term frequency-inverse document frequency. 40, 41, 97

VSM Vector Space Model. 38

A. Title of Appendix A

The following diagram shows a sample of pan12 dataset content.

```
</conversation>
<conversation id="85f0abac6ef5a2a23814a2ced73b5fb7">
  <message line="1">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>hello there</text>
  </message>
  <message line="2">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>how are ya?</text>
  </message>
  <message line="3">
    <author>2ded7a428b8b4536d49393c352fe1d1c</author>
    <time>14:11</time>
    <text>hey</text>
  </message>
  <message line="4">
    <author>bb8b358a10488f1ce25cdeb1df4a842a</author>
    <time>14:11</time>
    <text>so, where are you from, Stranger</text>
  </message>
</conversation>
<conversation id="80e3c3978ea07f46819f1f945cb04949">
  <message line="1">
    <author>a4529d1761aaeada66c6bdd6c93c78ea</author>
    <time>15:46</time>
```

Figure A.1: Pan 12 Dataset

B. Title of Appendix B

The following diagram shows a sample of dataset collected from the perverted justice website.

```
crazy_town_4_2_0 (07/19/12 5:00:54 PM): hi how are you?
crazy_town_4_2_0 (07/19/12 5:01:02 PM): so you just moved around here?
caffinatedboredom (07/19/12 5:01:51 PM): hi
caffinatedboredom (07/19/12 5:01:53 PM): yea
crazy_town_4_2_0 (07/19/12 5:01:57 PM): figured this was easier then email although i did email you too
crazy_town_4_2_0 (07/19/12 5:01:58 PM): lol
crazy_town_4_2_0 (07/19/12 5:02:12 PM): so where did you move here from?
caffinatedboredom (07/19/12 5:02:16 PM): what? i didnt even see it
crazy_town_4_2_0 (07/19/12 5:02:33 PM): oh...from matt should have one i just sent it a few mins ago
caffinatedboredom (07/19/12 5:02:45 PM): there it is
caffinatedboredom (07/19/12 5:02:48 PM): just popped up
crazy_town_4_2_0 (07/19/12 5:02:55 PM): lol oh random
caffinatedboredom (07/19/12 5:02:58 PM): im brandy 14 from near santa maria
crazy_town_4_2_0 (07/19/12 5:03:12 PM): oh nice im matt nice to meet you im in paso robles live alone here
caffinatedboredom (07/19/12 5:03:22 PM): nice
crazy_town_4_2_0 (07/19/12 5:03:25 PM): it said 18 on there whoops im 27 lol
crazy_town_4_2_0 (07/19/12 5:03:29 PM): im old :)
caffinatedboredom (07/19/12 5:03:57 PM): where did it say 18???
caffinatedboredom (07/19/12 5:04:07 PM): lol 27 isnt old
crazy_town_4_2_0 (07/19/12 5:04:09 PM): on your craigslist thing i thought
crazy_town_4_2_0 (07/19/12 5:04:14 PM): oh ok lol good
crazy_town_4_2_0 (07/19/12 5:04:27 PM): oh it didn't say an age my bad
crazy_town_4_2_0 (07/19/12 5:04:51 PM): lol glad im not too old
caffinatedboredom (07/19/12 5:05:15 PM): lol ok good cuz i was like wtf? i never said that lol
caffinatedboredom (07/19/12 5:05:21 PM): nah ur not
crazy_town_4_2_0 (07/19/12 5:05:37 PM): lol im sorry read it fast saw your cute picture figured i couldnt leave
you bored your too cute
crazy_town_4_2_0 (07/19/12 5:05:41 PM): so where did you come from?
caffinatedboredom (07/19/12 5:05:51 PM): ventura
caffinatedboredom (07/19/12 5:05:56 PM): damn nice pic :)
crazy_town_4_2_0 (07/19/12 5:06:01 PM): oh nice that is where i bought my car
crazy_town_4_2_0 (07/19/12 5:06:07 PM): aw really?
crazy_town_4_2_0 (07/19/12 5:06:23 PM): im tall hope you like tall guys
caffinatedboredom (07/19/12 5:06:31 PM): um yea
```

Figure B.1: Dataset Sample collected from Perverted Justice website