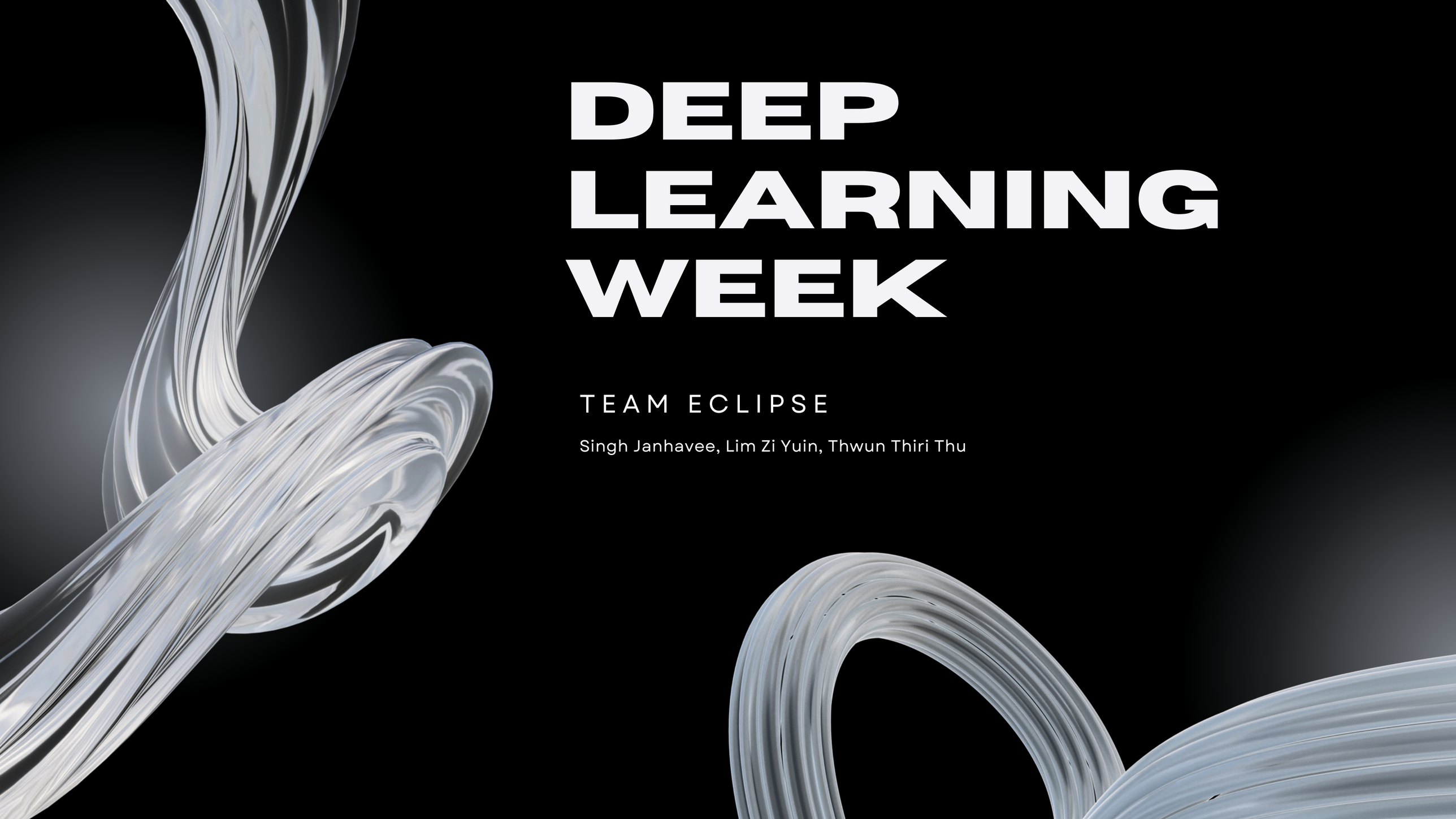
DEEP LEARNING WEEK

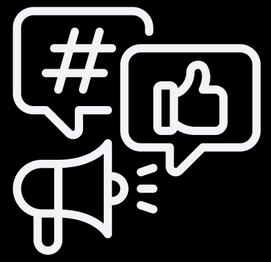
TEAM ECLIPSE

Singh Janhavee, Lim Zi Yuin, Thwun Thiri Thu



CURRENT CHALLENGES

- **Rapid transformation** of media



- Emerging issues like content moderation, deep fake detection, bias, and misinformation (false news are **70% more likely to be reposted**).

- Issue of content theft: Creators' and authors' original work are often used **without proper credit**, undermining their rights.

- **Need for trustworthy digital content** in the midst of manipulated media and fake news.

VERIFACT

An AI Powered web app designed to uphold media integrity and foster trust in today's digital content.

Core Features

- Deepfake Detection
- Fake News Detector
- Bias Checker
- Plagiarism Tracker
- Online Similarity Checker

To empower individuals and organizations through AI-driven tools to transform media integrity standards.

OUR SOLUTION

Verifact

Deepfake Detection | Fake News Detector | Bias Checker | Plagirism Checker | Online Similarity Checker

Fake News & Misinformation Detector

Enter article text for analysis:

Analyze for Fake News

Deepfake Detection

Upload an image to check for deepfake content

Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Screenshot 2025-03-02 001007.png 0.9MB

Analyze for Deepfake

Deepfake Detection Result: Deepfake Probability: 49.67%

Deepfake Detection | Fake News Detector | Bias Checker | Plagirism Checker | Online Similarity Checker

Check for Duplicate Content Online

Enter text to check for plagiarism:

Artificial Intelligence (AI) is revolutionizing the healthcare industry. AI-powered models can predict diseases, assist in diagnostics, and enhance personalized treatment. Recent studies highlight the potential of deep learning in medical image analysis, allowing early detection of conditions such as cancer.

Search for Duplicates

Similarity Scores

	Article	Link	Score
0	Revolutionizing healthcare: the role of artificial intelligence in ...	https://bmcmededuc.biomedcentral.com/articles/10.1186/s12909-023-04698-z	0.24
1	How Artificial Intelligence Is Shaping Medical Imaging ...	https://pmc.ncbi.nlm.nih.gov/articles/PMC10740686/	0.18
2	The Potential of Artificial Intelligence in Unveiling ...	https://pmc.ncbi.nlm.nih.gov/articles/PMC11566355/	0.38
3	Artificial Intelligence for Clinical Prediction: Exploring Key ...	https://www.sciencedirect.com/science/article/pii/S2666990024000156	0.30

DEEPFAKE DETECTION

1.DEEPFAKE DETECTION

- A pre-trained ***Xception model*** to analyze images and detect signs of manipulation.
- Assign a ***probability score*** (0 to 1) indicating the likelihood of an image being a deepfake.

```python
# Define the Xception model (using the timm library)
model = timm.create_model('xception', pretrained=False, num_classes=1)  # Set num_classes to 1 for binary classification
# Load the pre-trained weights
state_dict = torch.load("model.pth", map_location=torch.device("cpu"))
model.load_state_dict(state_dict, strict=False)

# Set model to evaluation mode
model.eval()

print("Model loaded successfully!")

# Image transformation
transform = transforms.Compose([
    transforms.Resize((299, 299)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

```python
def detect_deepfake(image_path):
    try:
        # Open and preprocess the image
        image = Image.open(image_path).convert("RGB")
        image = transform(image).unsqueeze(0)

        # Run inference
        with torch.no_grad():
            output = model(image)

        # Apply sigmoid if the output is raw logits for binary classification
        prediction = torch.sigmoid(output).item()

        return f"Deepfake Probability: {prediction:.2%}"
```

- **State-of-the-Art** Model – Uses Xception, known for detecting manipulated media.
- **Highly Accurate** – Applies deep learning techniques to spot inconsistencies.
- **Real-Time Processing** – Quickly evaluates and flags deepfake content.

FAKE NEWS DETECTOR

2. FAKE NEWS DETECTOR

- Leverages upon ***Azure OpenAI*** to analyze given articles
- AI Model evaluates content and provides a ***credibility score*** between 0 to 1, along with an ***explanation***

```python
# Fake News Detection using Azure OpenAI
def check_fake_news(article_text):
    response = client.chat.completions.create(
        model=deployment_name,
        messages=[
            {"role": "system", "content": "Analyze if this news is fake or real and provide a score from 0 to 1."},
            {"role": "user", "content": article_text}
        ],
        max_tokens=300
    )
    result = response.choices[0].message.content.strip()

    try:
        score, explanation = result.split("\n", 1)
        score = float(score.strip())
    except ValueError:
        score, explanation = None, result

    return score, explanation
```

- **Scalable & Cloud-Based** – AI runs efficiently using Azure's cloud computing.
- **Data-Driven Decisions** – Provides quantifiable scores & insights on misinformation.
- **User-Friendly Integration** – Can be embedded into websites & news platforms.

FAKE NEWS DETECTOR

Technologies used

- ***Azure Open AI*** - Analyze articles
- ***Natural Language Processing*** - Extract patterns, bias, and factual inconsistencies in news article
- ***Machine Learning and AI Ethics*** - Ensures model predictions are fair and accurate

BIAS CHECKER

3. BIAS CHECKER

- Leverages upon ***Azure OpenAI*** to to evaluate news articles for potential bias
- AI Model evaluates content and ***provides bias score (0 to 1) along with an explanation*** to highlight areas of concern

```python
# Bias Detection using Azure OpenAI
def detect_bias(text):
    response = client.chat.completions.create(
        model=deployment_name,
        messages=[
            {"role": "system", "content": "Analyze bias in this news article. Provide a score (0 to 1) and explanation."},
            {"role": "user", "content": text}
        ],
        max_tokens=300
    )
    result = response.choices[0].message.content.strip()

    try:
        score, explanation = result.split("\n", 1)
        score = float(score.strip())
    except ValueError:
        score, explanation = None, result

    return score, explanation
```

- **Unbiased Media Analysis** – Detects potential slants in reporting.
- **Scoring System** – Provides a bias probability score (0-1).
- **Explainable AI** – Offers insights into why the content is biased.

PLAGIARISM TRACKER

4. PLAGIARISM TRACKER

- Track if one's content ***have been used without being credited***
- Retrieve text and assess similarity between original and plagiarized versions

Technologies used

- ***Scikit-learn***
- ***TfidfVectorizer*** - text vectorization
- ***cosine_similarity*** - measures similarities

```python
# Extract Text from Local HTML Files
def extract_text_from_html(file_path):
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            soup = BeautifulSoup(file, "html.parser")
            text = soup.get_text().strip()
            return " ".join(text.split()).lower()
    except Exception as e:
        return f"Error reading file: {str(e)}"


# Content Tracking with Fixed Algorithm
def track_content_usage():
    try:
        original_file = "original.html"
        copied_files = ["copied_no_credit.html", "copied_with_credit.html"]

        original_text = extract_text_from_html(original_file)
        copied_texts = [extract_text_from_html(file) for file in copied_files]

        vectorizer = TfidfVectorizer()
        tfidf_matrix = vectorizer.fit_transform([original_text] + copied_texts)
        similarity = cosine_similarity(tfidf_matrix[0], tfidf_matrix[1:])

        # Compute raw plagiarism scores
        plagiarism_no_credit = round(similarity[0][0] * 100, 2)
        plagiarism_with_credit = round(similarity[0][1] * 100, 2)

        # FIX: Ensure "Without Credit" is never lower than "With Credit"
        if plagiarism_with_credit > plagiarism_no_credit:
            plagiarism_no_credit, plagiarism_with_credit = plagiarism_with_credit, plagiarism_no_credit

        return {
            "copied_no_credit": plagiarism_no_credit,
            "copied_with_credit": plagiarism_with_credit
        }
    except Exception as e:
        return {"error": str(e)}
```

- **NLP PROCESSING TECHNIQUE (TF-IDF)**
- **SAMPLE HTML** USED TO IMITATE ORIGINAL AND COPIED FILES

ONLINE SIMILARITY CHECKER

5. ONLINE SIMILARITY CHECKER

- Input text to check for duplicates.
- Retrieve articles using SerpAPI.
- Extract content and calculate similarity using TF-IDF and Cosine Similarity.

Tech Used:

- SerpAPI
- Newspaper Library
- Scikit-learn (TfidfVectorizer,Cosine Similarity)

```python
# Compute Similarity
def check_similarity(original_text, retrieved_texts):
    if not retrieved_texts:
        return []
    texts = [original_text] + [text["content"] for text in retrieved_texts if text["content"]]
    vectorizer = TfidfVectorizer(stop_words='english')

    try:
        tfidf_matrix = vectorizer.fit_transform(texts)
        return cosine_similarity(tfidf_matrix[0:1], tfidf_matrix[1:]).flatten()
    except ValueError:
        return []
```

```python
# Search Google using SerpAPI
def search_google(query, num_results=5):
    search_url = "https://serpapi.com/search"
    params = {"engine": "google", "q": query, "api_key": SERPAPI_KEY, "num": num_results}

    response = requests.get(search_url, params=params)
    if response.status_code != 200:
        return []
    results = response.json().get("organic_results", [])
    return [{"link": result["link"], "title": result.get("title", "No Title")} for result in results if "link" in result]


# Fetch and Extract Article Content
def fetch_content(url):
    try:
        article = Article(url)
        article.download()
        article.parse()
        return article.text.strip()
    except Exception:
        return ""
```

- The integration of Google Search via SerpAPI allows for r**eal-time content comparison directly from the web**, making it an efficient tool for tracking content originality online.

MARKET POTENTIAL

Increased Digital media consumption

- Number of digital media users worldwide is projected to exceed ***4.5 billion by 2025***

Rise in misinformation and fake news

- **80%** of Singaporeans were confident in detecting fake news, yet **91%** mistook fake headlines for real ones.

Intellectual property concerns in digital media

- Digital/ Cyber IP theft is increasing with **25%** of all IPs being digitally stolen/misued

Unique Integration:
Integration of ***deep learning for deepfake detection, Azure OpenAI for news analysis, and custom algorithms*** for plagiarism and bias detection makes Verifact stand out in the industry.

Innovation

Creativity, Feasibility, and Uniqueness

Feasibility:
With a modular architecture using technologies like Streamlit, OpenAI, and SerpAPI, Verifact is highly scalable and can handle ***large-scale content monitoring***.

Creative Approach:
Combines ***cutting-edge AI technologies*** to tackle issues of deepfakes, fake news, bias, and plagiarism in one platform.

User-Centric Design:
Intuitive and user-friendly UI with an easy navigation system to quickly detect, analyze, and report content issues.

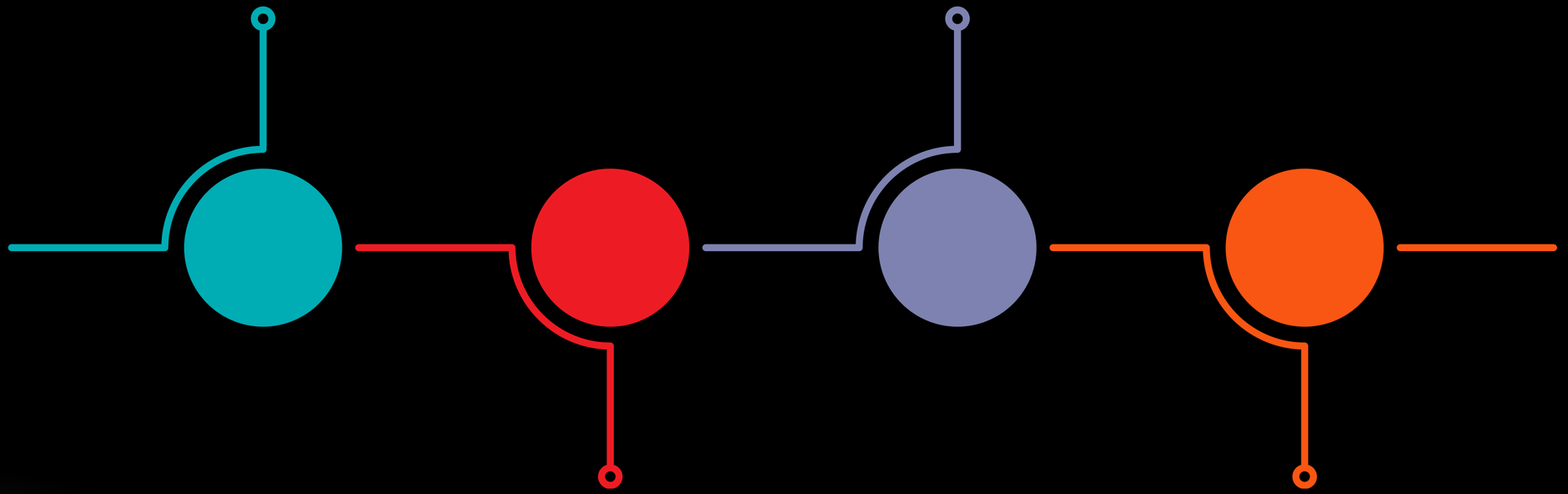
IMPACTS

Efficiency in Content Management

- Reduce time and resources spent as Verifact is a all-in-one tool.

Enhance trust in Digital Media

- Provide tools that help increase reliability of digital information.



Protect one's right to one's own work
valuable for creators, publishers and educational institutions.

Reduce spread of fake news

- Verifact allows one take ownership of the content we consume

THANK YOU

REFERENCES

1.Statista. (2021). Number of digital media users worldwide from 2017 to 2025. https://www.statista.com/topics/1164/social-networks/

2. World Intellectual Property Organization (WIPO). (2020). WIPO Director General Highlights Impact of Digital Technologies on Creative Industries. https://www.wipo.int/edocs/pubdocs/en/wipo_pub_941_2020.pdf

3. National Library Board. (n.d.). News literacy: S.U.R.E bingo [Infographic]. Retrieved from https://www.nlb.gov.sg/main/site/-/media/NLBMedia/Images/SURE/Resources-for-Teens/NLB_Infographic_News_Literacy_SURE_BINGO.pdf

4.Reuters. (2018, March 8). False news 70 percent more likely to spread on Twitter: Study. Reuters. https://www.reuters.com/article/technology/false-news-70-percent-more-likely-to-spread-on-twitter-study-idUSKCN1GK2R7/