Social Sentinel: Predicting National Self-Harm Trends Through Social Networks

D. Gayathri¹, Mr. G. Lokesh²

¹PG student, Vemu Institute Technology, P. Kothakota,
²Assistant Professor, Vemu Institute Technology, P. Kothakota

ABSTRACT
Since this study delves into the profound impacts of self-harm on individuals and economies, emphasizing the inadequacy of traditional statistics in tracking national trends. Introducing the innovative FAST framework, it harnesses social media data to forecast self-harm incidents. By training language models to discern mental health signals from online messages, this method transforms them into insightful time series data. Using machine learning regressors, the framework demonstrated superior forecasting accuracy. In a Thai case study, it surpassed conventional methods by over 40%. Additionally, incorporating the Decision Tree algorithm enhanced accuracy, reducing Mean Absolute Error compared to other algorithms. This research pioneers a transformative approach to predict nationwide self-harm trends and potentially forecast socioeconomic factors using social media analytics.

Keywords: Self-Harm, Social Networks

INTRODUCTION
Self-harm and suicide pose significant challenges, particularly in developing countries where 77% of suicide cases occur. This escalating trend is linked to technological advancements and rapid urbanization. Such incidents not only bring personal tragedies but also strain economies due to reduced labor productivity. Traditional methods of tracking these trends through administrative reports are resource-intensive and often yield delayed and coarse-grained data, limiting proactive policymaking. Existing forecasting techniques, like ARIMA and Holt-Winters, have proven inadequate, emphasizing the need for alternative data sources capturing real-time population reactions. This study introduces FAST, a pioneering framework...
leveraging social media data to forecast aggregate-level self-harm trends. By extracting mental signals from platforms like Twitter and Facebook, FAST combines this data with historical statistics to develop machine-learning-based forecasting models, offering timely and accurate insights for policymakers and healthcare stakeholders.

**LITERATURE SURVEY**

**M. Akyuz and C. Karulet al**

Author explores the influence of industrial production (IP), inflation, and investment on suicide mortality in Turkey from 1988-2018. Using Fourier cointegration tests and dynamic ordinary least square regression, the study reveals that IP and investment inversely correlate with suicide mortality, indicating a protective effect, while inflation shows a positive correlation, suggesting a detrimental impact. The findings underscore the significance of economic policies like investment promotion, IP enhancement, and disinflation not only for economic growth but also for mental health. This research highlights the need for holistic suicide prevention policies in developing countries, considering broader economic factors beyond just unemployment and GDP.

**A. Aldayel and W. Magdyet al**

Since the author delves into the evolving field of stance detection on social media, a nuanced approach to opinion mining that surpasses traditional sentiment analysis. This paper offers a comprehensive survey across diverse disciplines like natural language processing, web science, and social computing, examining their unique perspectives on stance detection. It presents an in-depth analysis of various techniques, including task definitions, target types, feature sets, and machine learning methods. Highlighting state-of-the-art results and effective strategies, the study also explores emerging trends and applications, such as opinion mining, prediction, and fake news detection. Ultimately, it identifies research gaps and suggests future directions to advance stance detection on social media.

**B. E. Belsher, D. J. Smolenskiet al**

As the author underscores the potential of suicide prediction models in enhancing the identification of individuals at elevated suicide risk through predictive algorithms applied to extensive datasets. These models are being tailored for prominent healthcare systems like the US Department of Defense, US Department of Veterans Affairs, and Kaiser Permanente. The study aims to assess the diagnostic accuracy of these models and simulate their impact on population-level suicide rates. After a thorough literature review spanning multiple databases and evaluating 64 unique prediction models across five countries with over 14 million participants, the findings indicate high global classification accuracy but extremely low predictive validity for actual suicide events. Consequently, despite promising classification accuracy, these models currently lack the precision needed for real-world clinical applications.
PROBLEM STATEMENT:

Now-a-days world is facing a new disease called depression which is the source to cause all diseases even suicides. Peoples may feel depression due to over competition at education and professional levels. Peoples often express depression from their faces or in their writing post skills. Addressing this problem requires monitoring and forecasting self-harm trends using real-time data from social media, providing timely insights for policy-makers to prevent and mitigate these tragic occurrences.

PROPOSED METHOD:

ARCHITECTURE

This paper employing FAST (forecast self-harm patterns) novel technique which analyse users post from online social networking sites, then extract emotions and sentiments from tweets to create training dataset. Author has collected suicides and death data from Thailand health department and then combines both emotions and death values to form a dataset. This trained model can be applied on user’s new tweet data to forecast self harm activity like injury and death. This prediction can help government to send all those depressed peoples for counselling to reduce self harm activities.
In above dataset screen first row represents dataset column names and remaining rows represents dataset values and each columns contains emotions and sentiments values like positive, negative, neutral etc. So by using above dataset we will train and test all algorithm performance. In propose paper author evaluated each algorithm performance in terms of MAE, RMSE and MAPE metrics. Each metric represents difference between True Test value and predicted value so the lower the difference the better is the algorithm.

**METHODOLOGY:**

1. **Data Preprocessing:**

The initial phase of any data science project involves preparing the dataset for analysis and modeling. Here’s how we handle this crucial step:

**Importing Essential Libraries:** We begin by importing essential Python libraries such as pandas for data manipulation, numpy for numerical computations, and scikit-learn for machine learning tasks.

**Reading the Dataset:** Our data resides in a CSV file, and we read it into a pandas DataFrame for easier manipulation and analysis.

**Handling Missing Values:** Data often comes with missing values, which can skew our analysis. We implement strategies like imputation or removal to address these missing values, ensuring data integrity.

**Feature-Target Split:** To train our predictive models, we separate the dataset into feature variables (X) and target variables, which in this case are ‘injury’ and ‘death counts’.

2. **Model Selection and Evaluation:**

Now that our data is preprocessed, let's move on to selecting and evaluating predictive models:

**Algorithm Selection:** We opt for a diverse set of algorithms to capture different aspects of our data's patterns. Our selection includes ARIMA for time series forecasting, Bayesian Ridge for regression tasks, SVR, XGBoost, Random Forest, CatBoost, and Decision Trees.

**Data Splitting:** Before training our models, we split our dataset into training and testing subsets using train_test_split from scikit-learn, ensuring that our models are evaluated on unseen data.

**Training and Prediction:** We train each selected model on the training data and then make predictions on the test set.

**Performance Evaluation:** To gauge the efficacy of our models, we employ key performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

**Visualization:** For a more intuitive understanding of our models’ predictions, we visualize the predicted values against the actual values using plots or graphs.

3. **Algorithm Comparison:**

With predictions and evaluations in hand, it's time to compare the performance of our selected algorithms:
Performance Metrics Comparison: We juxtapose the performance of all algorithms based on our chosen evaluation metrics, providing insights into their relative strengths and weaknesses.

Tabular Summary: To simplify the comparison, we summarize the results in a tabular format, detailing the performance of each algorithm for both 'injury' and 'death' predictions.

4. Prediction on Test Data:

Now, let's apply our best-performing model to make predictions on new, unseen data:

Reading Test Data: We read the test data, stored in a CSV file, into a pandas DataFrame, ensuring consistency with our training data preprocessing.

Handling Missing Values: As before, we address any missing values in the test dataset to ensure robust predictions.

Final Predictions: Leveraging the insights from our evaluations, we deploy the best-performing model, in this case, the Decision Tree, to make predictions on the test data for both 'injury' and 'death' counts.

Mean Squared Error (MSE):

MSE is calculated by taking the average of the squared differences between the predicted values and the actual values.

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2
\]

Root Mean Squared Error (RMSE):

RMSE is the square root of the MSE and provides a measure of the average magnitude of the errors in the predictions.

\[
RMSE = \sqrt{MSE}
\]

RESULTS:

In above graph x-axis represents number of test data and y-axis represents Injury values and red line represents True Injury and green line represents Predicted injury and there is lots of gaps between red and green line so ARIMA performance is not good. If predicted values are accurate then both lines will overlap.
In above screen training Bayesian Ridge on Injury data and then displaying its performance MAE and other values and in graph both lines are overlapping with little gap.

Above is the Bayesian ridge training on death data

Above is the SVM training on Injury data

Above is the SVM training on death data
Above is the XGBOOST training on injury data

Above is the random forest training on Injury data

Above is the XGBOOST training on Death data

Above is the random forest training on Death data
In above screen training extension decision tree on Death data and can see its performance metrics

<table>
<thead>
<tr>
<th>Prediction Type</th>
<th>Algorithm Name</th>
<th>MAE</th>
<th>RMSE</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Injury</td>
<td>ARIMA</td>
<td>147.634804</td>
<td>159.191989</td>
</tr>
<tr>
<td>1</td>
<td>Injury</td>
<td>Bayesian Ridge</td>
<td>84.689499</td>
<td>124.825600</td>
</tr>
<tr>
<td>2</td>
<td>Injury</td>
<td>Linear SVR</td>
<td>160.226892</td>
<td>184.536620</td>
</tr>
<tr>
<td>3</td>
<td>Injury</td>
<td>XGBoost</td>
<td>94.073066</td>
<td>141.645493</td>
</tr>
<tr>
<td>4</td>
<td>Injury</td>
<td>Random Forest</td>
<td>100.055556</td>
<td>154.443715</td>
</tr>
<tr>
<td>5</td>
<td>Injury</td>
<td>Cat Boost</td>
<td>136.929022</td>
<td>147.390562</td>
</tr>
<tr>
<td>6</td>
<td>Injury</td>
<td>Extension Decision Tree</td>
<td>35.111111</td>
<td>96.452637</td>
</tr>
</tbody>
</table>

In above screen can see each algorithm MAE, RMSE and MAPE error values of all algorithms on INJURY data and this metrics are represented as error so if error values are less then algorithm will be consider as best and in all algorithms XGBOOST and extension decision tree got less error rates.

<table>
<thead>
<tr>
<th>Prediction Type</th>
<th>Algorithm Name</th>
<th>MAE</th>
<th>RMSE</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Death</td>
<td>ARIMA</td>
<td>248.117776</td>
<td>278.234738</td>
</tr>
<tr>
<td>1</td>
<td>Death</td>
<td>Bayesian Ridge</td>
<td>215.090862</td>
<td>286.371124</td>
</tr>
<tr>
<td>2</td>
<td>Death</td>
<td>Linear SVR</td>
<td>262.532685</td>
<td>308.526217</td>
</tr>
<tr>
<td>3</td>
<td>Death</td>
<td>XGBoost</td>
<td>115.542982</td>
<td>172.160009</td>
</tr>
<tr>
<td>4</td>
<td>Death</td>
<td>Random Forest</td>
<td>196.444444</td>
<td>285.175287</td>
</tr>
<tr>
<td>5</td>
<td>Death</td>
<td>Cat Boost</td>
<td>246.000126</td>
<td>272.708290</td>
</tr>
<tr>
<td>6</td>
<td>Death</td>
<td>Extension Decision Tree</td>
<td>78.111111</td>
<td>177.544674</td>
</tr>
</tbody>
</table>

In above screen can see each algorithm MAE, RMSE and MAPE error values of all algorithms on DEATH data and in all algorithms XGBOOST and extension decision tree got less error rates.
PREDICTION:

Test Data = ['2019-04-30' 0.0013975184680863 0.2199765598397869 0.095684500764850801 0.0741907475203086 0.0267022319923398 0.001002270930726 0.0078279191108553 0.1431122630735526 0.010536230687758 0.0001064150001027 0.700534563642043 0.7590150786462956 0.24010482313537044] =====> Forecasted Injury : 157.0 Forecasted Death : 550.0

Test Data = ['2019-05-31' 0.165547287514801 0.2527987790007002 0.005508844708685 0.6361682392136812 0.04682643848698 0.006676596868302 0.0095823269792046 0.17159268168468617 0.008210297914469 0.009368087132218 0.659348085270299 0.7591766121803085 0.2680233878201102] =====> Forecasted Injury : 134.0 Forecasted Death : 538.0

Test Data = ['2019-06-30' 0.1045612388319541 0.262993651352631 0.005662845216829 0.6267882554561098 0.0820999667717636 0.0013813664910832 0.01214623901047024 0.162595091997457 0.0390008534057982 0.0979625209293197 0.6247853011052636 0.7276021623421914 0.2721937712660885] =====> Forecasted Injury : 154.0 Forecasted Death : 514.0

Test Data = ['2019-07-31' 0.096470872239406 0.2666322952552113 0.0038952640476732 0.6523992977444789 0.0639810739434619 0.0015048690149299 0.0085458626768565 0.17230411373325455 0.0295224718305855 0.0943929074476490 0.04932637852121576 0.7587910133802087 0.24120081866619718] =====> Forecasted Injury : 213.0 Forecasted Death : 407.0

Test Data = ['2019-08-31' 0.108695335376754 0.273703548921453 0.0039302731164312 0.6200730447770681 0.047179964549927 0.0015048426900999999 0.0121597986123169 0.187393214486899 0.0319765079298 0.103854933913812 0.6165936614056945 0.7269796087358125 0.2412008184648795] =====> Forecasted Injury : 154.0 Forecasted Death : 478.0

Test Data = ['2019-09-30' 0.11542387098446663 0.2714538165512693 0.0033235341666214 0.6136319605833612 0.0400105246645784 0.001495120625748 0.017600874949899 0.20761613798424198 0.03031932722034199 0.100318260900629 0.905029208272767 0.7461113129702879 0.358467870071122] =====> Forecasted Injury : 148.0 Forecasted Death : 837.0

In above screen reading test dataset values and then performing forecasting of INJURY and DEATH and in output before arrow symbol we can see test data values and after arrow symbol can see forecasted values of INJURY and DEATH.
CONCLUSION

Depression is a growing concern worldwide, often leading to self-harm and even suicides. This project aims to forecast self-harm trends using social networks, leveraging the FAST (Forecast Self-Harm Patterns) technique. By analyzing users' online posts, emotions, and sentiments, we trained various machine learning algorithms to predict injury and death occurrences. Notably, XGBoost and the extension Decision Tree algorithm outperformed others in forecasting accuracy. The generated dataset and models hold promise for early intervention strategies, guiding governments in providing timely counseling and support to mitigate self-harm activities.

REFERENCES:


