Identifying Potential Criminals and Terrorists From Social Media Analysis

Doug Strueby
Univeristy of Calgary
30122048

Nolan Ruzicki
Univeristy of Calgary
30132405

I. Abstract/Summary

From the continually growing popularity of Twitter (now x) we wanted to use this platform with millions of individuals sharing their opinions, emotions, and ideas to determine aggression behind tweets and create a prediction model that assesses the likelihood that the Twitter user will become a criminal in the future. This topic piqued our interest as we are both interested in criminology and stories related to criminals' disturbing post history on social media which were discovered after the crime is a topic that we found interesting. Additionally, we want to see if there are any similarities between the people who are actual criminals, people who are joking, and people who have not committed crimes but may in the future. We plan to utilize Twitter data paired with crime data from the year 2023 in Toronto to find any potential patterns or signs indicative of criminal behavior. We expect to create an effective model that can discern a post containing actual aggression or intent compared to a post where somebody is telling a joke or poking fun at a friend. We hope that this model will assist in the field of threat detection and allow for the prevention of potentially harmful individuals going unnoticed until it is too late.

II. Background Study/Literature Review

A. *The Relationship Between Social Media Data and Crime Rates in the United States [1]*

In this paper, data was collected from “city-data.com” [9]. If the crime data for 2012 (the year of analysis) was not available for a specific country, the crime rate was inputted using the historical crime rate index. Tweets were collected from May 26 to December 9, 2012. They were then classified as "drug-related" if they contained any keywords from a list associated with drug and substance use. One observation was created for each county, each considered complete if it included geocoded tweet data, city-level crime data, and the Gini index data. Pearson’s correlation coefficient was used to determine the pairwise linear relationship among the aggregate county-level crime data, drug-related tweets, and the Gini index. A negative binomial regression model was used to show the association between crime data and drug-related tweets. This association was additionally modeled and adjusted at the county level using the Gini index. The model’s performance was then compared using Akaike Information Criterion (AIC) scores, with a smaller AIC suggesting better fitness of the model. A random intercept and slope (RIAS) model was used for counties clustered within a state in terms of geolocation, policy, and culture. As mentioned, the population characteristics at the county level were correlated with crime rates. Therefore, additional adjustments were made for the percentage of young people (age 15-44) who contributed more than half of the tweets, the percentage of whites in the county population, and the percentage of African Americans in the population. There was a high correlation between drug-related tweets and the crime rate index, with a correlation coefficient of 0.82 at the state level. At the county level, the frequency of drug-related tweets in 2012 was highly correlated with both the 2012 and 2013 crime indices. The association between drug-related tweets and crime rate was statistically significant in all five models for both 2012 and 2013; the best performance was achieved using the model adjusted by the Gini index, county-level percentage of young people, whites, and African Americans. Based on the results, they expected a 9.5% increase in the crime index rate per 1 million population when drug-related tweets increased by 10%. The association was positively lower when the crime index rate was above average in the state.

B. *Criminal behavior identification using social media forensics [2]*

This paper proposes a framework for crime prevention by detecting criminal behavior using social media data. The dataset they used was a cyber-troll dataset [10] containing data 20,001 tweets of which 7,822 were deemed to be aggressive and the remaining 12,179 were not. The proposed methodology for this model included data preprocessing, feature extraction using unigrams, bigrams, and trigrams, and a Deep Neural Network algorithm to classify the model. They completed feature extraction by using the bag of words (BoW) algorithm and further increased the performance of the model through the use of TI-IDF and GloVe. This is done by using BoW to calculate the occurrences of words, and then important lexical features were extracted through the use of TI-IDF, then GloVe was used to extract the semantic relatedness of words by learning meaningful vector similarities. The performance of the model was evaluated in terms of accuracy, precision, recall, loss, and F1 score. Based on the results in these categories the proposed model outperformed the state-of-the-art approach, with results of an F1 score of 87% using a unigram + bigram and an F1 score of 88% using the bigram + trigram case. Although the implementation of the models

outperforms the base model, data sparsity is a major problem that occurs with the tweets within the dataset due to the short length of the text. This model performance concludes that the proposed framework can be used to identify individuals with criminal mindsets and could potentially be used in assisting law enforcement agencies in crime prevention.

C. Multimodal Deep Learning Crime Prediction Using Tweets [3]

The paper proposes using a multimodal deep learning approach for crime prediction, integrating Twitter data with historical crime data in September 2019. The proposed method uses a modified ConvBiLSTM architecture, which combines CNN and biLSTM. This architecture incorporates a data fusion layer to combine information from both crime data and tweets, to leverage semantic knowledge learned from text data for crime prediction. There are six steps in this methodology being:
Word Vectorization: Crime data and tweet data are independently put into a network and are then segmented into work vectors one by one.
Convolutional Layer: The newly created word vectors undergo a convolutional operation to extract local features to capture the important words from the tweets.
Max-Pooling Layer: Extracts the most prominent features from the convolutional layer and reduces dimensionality.
Data Fusion Layer: Feature-level data fusion is applied to combine information from the crime and tweet data to create a singular representation of the data.
Bi-LSTM Layer: Bidirectional LSTM layers are employed to capture dependencies and context information from past and future directions.
Dense Layer and Result: The dense layers are used for feature mapping, followed by a final output layer which uses the Sigmoid function to produce the output.
This experiment involves a baseline comparison between the proposed method with existing methods. These existing methods are SVM, Logistic Regression, Sentiment SVM, Sentiment-based Logistic Regression, NAHC, DNN, CrimeTelescope, ANN+BERT, and BERT base Model. The results show that the proposed model outperforms the mentioned baseline methods with an accuracy rate of 97.75%. Found that the results indicate that integrating tweet data with historical crime data using multimodal deep learning techniques can enhance crime prediction accuracy. However, the study is limited to English tweets related to specific crime types, and future research could be done to explore other languages and crime categories to further validate the model.

D. Determiniation of Potential Criminals in Social Network [4]

This paper aims to propose a system for finding people who place Tweets with real and serious threats in them. The system presented detects Twitter users that may be probable criminals and performs a content screening of them. The following information can be recorded about people who place criminal Tweets: usernames, Tweets, definitions they made themselves, the content of the Tweets they share, and where and when they made the share (date and place). These data are exposed to semantic analysis, and separated from irrelevant data, which is processed by machine learning. Based on what is relevant a matrix is used that contains the weights of words and sentences by meaning, where the given data is classified using cross-validation, into cyber-crime or organized crime. The classification of these two categories serves as a basis for scanning their potential criminals. They can be scanned based on shares or discovered keywords. Each share can be discovered by given keywords, based on a certain setting (date and location). In the end, the crimes committed can be categorized and put into order (according to the taxonomy given above). The group of occupied individuals met the given criterion which can make an inference whether it is organized or cybercrime they committed based on their activity on Twitter detected by given keywords. Three main machine-learning models were looked at for the analysis in this paper. First, the paper looked at the KNN (K Nearest Neighbours) algorithm was tried because of the success of a study that analyzed popular events on Twitter. Next, the paper looked at the MLP (Multi-Layer Perceptron) algorithm. This model was used to achieve high success in labeling g music by semantically web-based information management. Finally, the Support Vector Machine (SVM) algorithm was used because it had a 91% success rate when determining a Twitter user's political orientation. This paper used cross-validation (CV) of 7 and 10 to analyze and found that the MLP algorithm was the most accurate with a CV of 10. For the KNN algorithm with k=5, the mean absolute error, F-measure, ROC area, and accuracy were computed for both the CV of 7 and 10. The results of the CV of 7 were a mean absolute error of 0.43, an F-measure of 0.63, an ROC area of 0.66, and an accuracy of 63.28%. For the KNN with a CV of 10, the mean absolute error was 0.43, the F-measure was 0.64, the ROC area was 0.67, and the accuracy was 63.54%. A similar process was done for MLP where results were found for CVs of 7 and 10. The results for the CV of 7 were a mean absolute error of 0.34, an F-measure was 0.71, an ROC area was 0.78, and an accuracy was 70.83%. For the MLP model with a CV of 10, the results were, a mean absolute error of 0.34, the F-measure was 0.72, the ROC area was 0.78, and the accuracy was 70.83% showing a slight difference between the models. Finally, for SVM the same process was run. For the CV of 7, the results were, a mean absolute error of 0.30, an F-measure was 0.70, an ROC area was 0.70, and the accuracy was 70.05%, and the CV of 10 having, a mean absolute error of 0.29, the F-measure was 0.71, the ROC area was 0.71, and the accuracy was 70.57%. Making the best-performing model, the MLP model with a CV of 10.

E. Similarity Analysis of Criminal on Social Networks: An Example on Twitter [5]

This work is used to investigate criminals on Twitter and discover similarities between them and groups, leaders, and illegal activities. To achieve this there were three main objectives: Objective 1: Definition of a meta-model that enables user analysis based on their available data.
Objective 2: The definition of an evaluation mechanism that allows for the assessment of groups of suspicious users and identifies similarities among them. Objective 3: Discovering illegal activities along with hidden mediators and

connections between groups of similar users related to OC and TNs. The method proposed in this paper is intended to support law enforcement agencies to speed up the discovery of similarities among suspicious users, on social networks related to organized crime (OC) or Terrorist networks (TN). The method is a three-phase-based process, and the work product input and output are shown in Figure I.

Phases	Input work-products	Output work-products
Meta-Model definition	Suspicious User Profiles	Data-model
Users Similarity identification	Data-model	Users similarity analysis
Behavior Discovering	Users similarity analysis	Groups, Activities, Mediators

Figure 1: The proposed method: phases and related work-products [5]

The initial phase is the meta-model definition, which is faced with the need to define the meta-data that can be retrieved and the availability of this data. To achieve this the data is divided into 4 different groups, Group 1 contains location information (e.g., City, Country, Time zone, etc.). Personal information (e.g., Name, Description, Photo, Age, etc.) and most attributes that are optional fall into Group 2. Group 3 contains information related to the users' interests (e.g., URLs, Posts, etc.). Finally, Group 4 consists of information about the users' relations and links to other users (e.g., Friends, Mentions, etc.). The phase uses an algorithm that takes in an input list of suspicious users' profiles and creates a data model used in the next phase. The next phase is the user similarity identification phase. It takes the data model from the first phase and uses that data to identify similarities between criminal users. This phase is broken up into two steps, Normalization, and Evaluation. The Normalization step is used to make the set of users comparable in a ”fair” way. The evaluation steps the actual similarity between pairs of users. Two similarity measures were used (i.e. Cosine and Jaccard). This step also uses a text analysis of the user's tweets to discover top words which are filtered using a Term Frequency-Inverse Documentation frequency (TF-IDF) [11] two-step approach and topics of interest were found using a Latent Dirichlet Allocation (LDA) [12] approach. The last phase is used to cluster users into groups and identify the leaders among the groups. Clustering and rule mining were combined to extract specific patterns. These processes were combined and using a Random Forest Classifier it produced an accuracy of 92% and a recall of 100%, in the recognition of people related to 5 kinds of illegal activities related to Drugs, Human trafficking, and Arms/Weapon.

F. Crime Detection and Analysis From Social Media Messages Using Machine Learning and Natural Language Proccesing Technique [6]

This paper focuses on crime detection on social media. The crimes focused on in the text are attack and drug-related crimes, hate speech, and offensive messages. This paper aims to use natural language processing and study classification algorithms used for text classification. They aim to accurately predict crime data with algorithms such as the random forest and SVM algorithms. Additionally, they wanted the performance of similar models (e.g. KNN, and Naïve Bayes) to detect crime in social media. This paper took multiple steps in the research that was conducted which were data collection text-preprocessing, feature extraction, classification (detection) with SVM, classification 2 (Analysis) with random forest, and testing using unseen messages as test data. The data set used was the Twitter spam data set, and after examining it, it was found that these Tweets were related to crime. For text pre-processing, there was a significant amount of data cleaning to reduce noise and increase the model's performance. Stop words were removed all with punctuation, words were stemmed, URLs and user mentions were replaced with plain text and finally words were converted to sentences. Feature extraction is used to turn the textual data into numbers so that the machine learning algorithms are run on this data. The TF-IDF algorithm is used to do this. This paper used ranges to determine the performance of each model they evaluated 0%-60% was considered bad, 70%-79% good, 80%-89% excellent and 90%-100% overfitted. Using SVM an “excellent” model was created with an accuracy of 85.69%, a precision of 85.85%, and a recall of 85.46%. The remaining two models were considered “bad” with Naïve Bayes having an accuracy of 60.02%, a precision of 56.59%, and a recall of 85.92%. KNN had an accuracy of 50.21%, a precision of 73.33%, and a recall of 85.92%. This shows that the best model for crime detection was SVM. In terms of algorithms used for crime analysis, random forest had accuracy, precision, and recall of 72.16%. While Cost-sensitive SVM was minorly less practical with an accuracy, precision, and recall of 69.33%. This paper used Natural Language Processing and machine learning algorithms to detect crime. It found that the SVM is the most accurate with an accuracy of 85.69% for the detection of crime and Random Forest is the best for the analysis of crime with an accuracy of 72.16%. Both models have a good to excellent level of accuracy without overfitting.

III. PROPOSED METHODOLOGY

We are going to be using two datasets, one containing 1.6 million tweets, alongside a data set containing crime major crime indicators in Toronto. We are using the crime indicators to determine the count of each crime classification within the year 2023 to determine the most occurring crimes and keywords associated with them. Based on a count of the MCI categories shown in Figure 2 alongside the count of crime classifications, we were able to determine that within Toronto Assault was the most common crime indicator, followed by Auto Theft and Break and Enter. Based on this information we filtered our tweet data based on these keywords paired with other crime-related words related to the identified terms. These are words such as “kill”, “firearm”, “steal”, and “burglary”. Our preprocessing/analysis will be run on the Twitter data that we obtained based on this filtering method for our tweets

based on crime data.

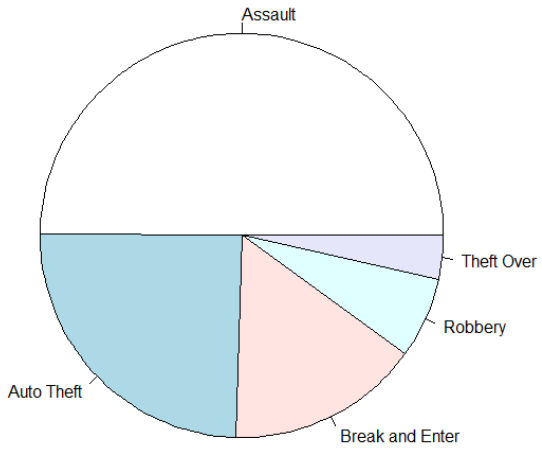

Figure 2: Count of MCI Categories

Before we can run any analysis on this data we must do some preprocessing. The data gathered from Twitter can be very messy with the use of stop words, punctuation, emojis, capitalization, and more. It can be hard to determine what is important from this data. Additionally, this data must be understood by the machine learning models that we will be using, and since we are working with text data and these models only understand how to work with numbers we must extract terms and features so that it can be a usable data set. The first step of the preprocessing will be removing the use of stop words (e.g. as, the, this, that, is, etc.) these words occur often and are unnecessary for the use of criminal classification. Next will be to remove punctuation and capitalization, this also adds no value to the data and can make the same words be confused as different; this will make it more difficult for our model to accurately make classifications. Next, we must handle URLs and mentions of other users. This data will also need to be removed from the data set as our model cannot access this information, making it not useful for us to use. Finally, we were used by Lombo et al.'s [5] we first used stemming to get each word down to its root then we will use Lemmatization which has a better understanding of the word and will help us to produce a better data model for machine learning algorithms. Finally, to make the data readable to our algorithms we need to convert the textual data into numerical. We will utilize the TF-IDF algorithm to achieve this. We decided on this algorithm as opposed to other NPL algorithms like Bag of Words (BoW) because we believe that it will provide us with a more robust data set for our analysis. This is because BoW only provides insights on the number of occurrences of words where TF-IDF provides weighting of importance for these words in a given document. Additionally, it was shown by Dias Canedo et al. [17] that TF-IDF performs better than BoW for some of the algorithms we would like to use, see Figure 3 below.

Binary Classification

	BoW			TF-IDF			CHI^2		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
SVM	0.90	0.90	0.90	0.91	0.91	0.91	0.90	0.90	0.90
MNB	0.91	0.91	0.91	0.91	0.91	0.90	0.89	0.89	0.89
kNN	0.82	0.82	0.82	0.87	0.87	0.87	0.84	0.84	0.84
LR	0.88	0.88	0.88	0.91	0.91	0.91	0.89	0.89	0.89

Figure 3: Binary Classification model performances

This processed data will then be fed into our chosen machine-learning models to classify criminal users. From the information found in the literature review, there was a diverse use of machine learning algorithms which gave us lots to choose from for our paper. We wanted to use multiple algorithms and compare the results we got to see which ones are best for the task that we want them to perform. One of the best-performing algorithms was the Random Forest algorithm [13], we noticed that this one showed up in more than one of the papers we reviewed, and we determined that it was one of the best-performing algorithms. Additionally, the Random Forest algorithm is good for analyzing text data because of its ability to handle high-dimensional and sparse data sets which is common in text data [16]. There are some possible drawbacks to this algorithm which would be the lack of transparency as it is a black box algorithm in that it is difficult to tell how each tree contributes to the final answer. Another algorithm that we can use which would be faster and have more transparency would be Logistic Regression. Logistic Regression is also simple and fast which may be better. Another model that we saw was used quite frequently was the support vector machine (SVM) [14]. This model seemed to have high accuracy on a variety of different datasets related to our problem based on our background literature review, and it works best when the dataset is relatively small and complex, and it will provide similar results to our logistic regression model and allow us to compare which of the two to go forward with. For these reasons and the fact that SVM has a relatively simple setup and a short runtime, it would be an appropriate solution for our model. The final algorithm that we are going to run for our model is long short-term memory (LTSM) [15]. These models are a type of recurrent neural network architecture and have recently been growing in importance in the field of deep learning, especially in terms of sequential data processing in natural language processing. LSTM is typically favored in situations regarding speech recognition and sentiment analysis, due to the memory cell that the model contains which can store and retrieve information over long sequences [18]. The typical problem that researchers run into with this model is that there tends to be an overfitting problem as other deep learning models contain. Additionally, there are long training times that may require powerful hardware to run, and similar to the random forest models LSTM is another black box model that will limit the interpretability of the model [18]. Based on the effectiveness of this model in previous iterations of a similar problem [3] and since the use cases for the algorithm tend to be related to text analysis, we believe that this will be an effective algorithm for creating the model. Before running our data models, we need to preprocess the data as described above, this will be achieved through the R library “tm” [19] then to make the data interpretable by our models we will use the TF-IDF technique to determine the importance of each word in each tweet [20]. We will then be splitting our data into training and test sets. We chose to split them with 80% of our data being in the training set and the remaining 20% being in the test set. Additionally, for each of our models (besides SVM) we chose to reduce our threshold to 0.25 from the normal value of 0.5 to reduce the number of false negatives our model produces as we see this as a more costly error in comparison to a false positive. The first model that will be created will be the random forest model.

The creation of this model will start with our processed data from the preprocessing step. We will take this data and partition it into training and testing data, with a split of 80% and 20% respectively. Then we will take this data and using y as the value as the predicted column we will run a random forest model using the "randomForest" library from R [21], with the number of trees set to 100. The next model to create is logistic regression, we will be using the same partitioned data as we did for the random forest, and we will be using the "glm" function which is included in R [22] for this function we will again have y as the predicted value with all other variables as predictors and the family will use if the binomial family because y is binary. To create the SVM model, we will again use R to create it. We already have the data loaded into training and test sets from the initial setup which we will be using again. To run the model, we will use the "SVM" function from the "e1071" library and we will be predicting the y variable based on the rest of the attributes. We're going to be using the "polynomial" kernel with a cost of 25 for the SVM model [23]. From there, we can use this model to create predictions for our test data as we have with the previous models. In terms of the LTSM model we are going to be using, firstly we will need to load our data differently as we need all variables to initially be in a numerical format and we will additionally need to split the predictors and response variables into independent data frames. From there we defined our model similarly to that in a guide for LTSM in r [24] through the use of 3 layers and the relu activation function being used in the first two layers and the sigmoid being used for the final layer. We then compiled the model using the optimizer "adam", the loss function of "binary_crossentropy", and the metric being accuracy. From there we were able to fit the model based on our training data and we chose to use 25 epochs, a batch size of 128, and a validation split of 0.2, these all being according to the guide [24]. From there we simply made another prediction model with a and changed the threshold to be 0.25 to increase the precision.

IV. Detailed Description of the Dataset

We will be using a dataset from the Toronto Police reporting on major crime indicators [7]. This dataset provides comprehensive information on major crime incidents reported within the city of Toronto. It covers a wide range of different incidents and aims to offer insights into crime trends, patterns, and hotspots within the city, allowing law enforcement to inform the public of safety initiatives. This dataset has information dating back to 2014. However, we will only be using incidents from 2023, reducing the dataset from 372,899 records as of March 27, 2023, to 49,395 entries. The dataset consists of 15 numerical columns containing information such as coordinates, dates (split into multiple columns for year, month, day, and hour), and a variety of different codes, along with 16 categorical columns providing information such as location type, description of the offense, MCI category, and the neighborhood. Our second dataset is a Twitter sentiment analysis dataset sourced from Kaggle [8]. This dataset contains 1.6 million records and includes the following six fields: the polarity of the tweet which we will use as our predictor, the ID of the tweet, the date of the tweet (all from the year 2009), the query (which will not be included in the analysis), the user who tweeted, and the text of the tweet. As mentioned, this dataset contains 1,048,575 tweets. However, using a list of keywords/terms based on the description of the offense and MCI category columns within the crime indicators dataset, we selected offenses/MCI categories that had the most occurrences in 2023 to choose these terms/words. Through this process, we narrowed the number of entries down to 14,016, allowing us to train our model based on relevant tweets.

V. Implementation Requirements

We are going to be using the coding language R to complete all the processes of our analysis including data preprocessing, analysis, and reporting the results. We chose to use R as we are both familiar with the language and it has the capabilities to run all of the algorithms that we are planning to use for the analysis. The libraries that we currently are planning on using within R that aren't included in the base model for our analysis will be:

- "syuzhet" for the sentiment analysis algorithm.
- "MASS" for the LDA algorithm.
- "e1071" for the SVM algorithm.
- "tm" for TF-IDF and TermDocumentMatrixs.
- "randomForest" for building a random forest.
- "keras" and "tensorflow" for building LSTM.

We expect that as we create our model additional libraries will need to be used, but for the time being, these are the current packages that are going to be used.

VI. Results and Detailed Analysis

For our evaluation based on the Twitter sentiment analysis dataset being filtered based on the leading crime categories in Toronto. Using four different algorithms to predict aggression and therefore potential intent or a possibility to follow through with a threat within a tweet. Our results can be seen in Figure 4 and were chosen according to an article that provided insight into the best evaluation methods for classification models [25]. The main indicators we are going to be using in this evaluation are accuracy, precision, recall, and F-measure based on true and false positives derived from our confusion matrixes for each model. We will additionally be providing the RMSE of each model.

	Random Forest	Logistic Regression	SVM	LSTM
Accuracy	0.81	0.75	0.80	0.75
Recall	0.34	0.32	0.27	0.32
Precision	0.17	0.49	0.15	0.47
F-Measure	0.23	0.39	0.19	0.38
RMSE	0.43	0.50	0.45	0.36

Figure 4: Results from our different models

The test accuracy for our four models is listed in Figure 4 and gives an overall sense of the number of correctly predicted values that appeared highest in SVM and Random Forest, largely due to these models airing towards predicting more false predictions rather than true. Our recall and precision were generally low for the models with our highest precision being in logistic regression with the value of 0.49, meaning that every true prediction has a 49% chance of being correct. Our F-measures are also fairly low

for all models with our highest value being 0.39 once again for logistic regression. In terms of RMSE our best-performing model was LSTM with a value of 0.36. In Figures 5 and 6 we can see our models' performances with the original threshold and with our new adjusted threshold.

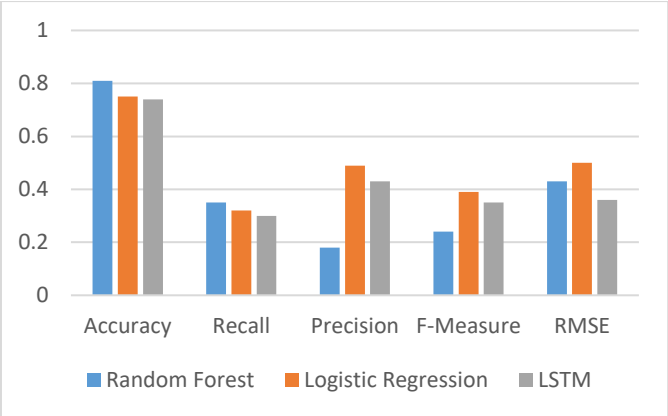

Figure 5: Results with a threshold of 0.25

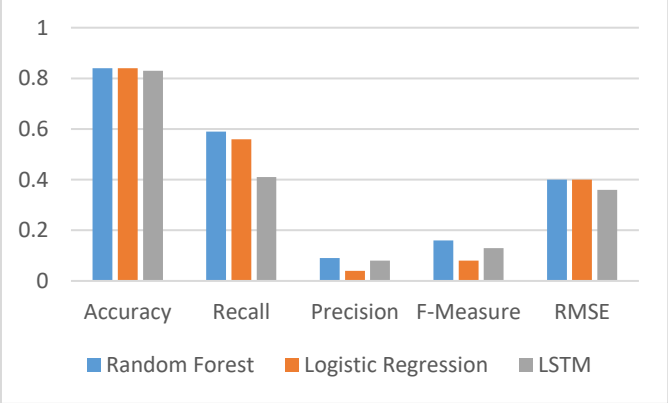

Figure 6: Results with a threshold of 0.50

The threshold of 0.25 increased both our precision and F-measure by a significant amount. However, it additionally decreased our accuracy and recall and increased the RMSE. We believe that these negatives that came with the adjusted threshold are outweighed by the increase in our precision and F-Measure as our main focus is to increase the precision of our model. Based on these results we concluded that the LTSM model was our most effective as it closely follows logistic regression in all aspects but has a significantly better RMSE.

VII. Conclusion

With the ever-growing popularity of social media websites and applications like Twitter, Instagram, and many others there is also a growing potential that someone could be using these websites to voice their criminal ideas and give indications that they could be potentially criminal in the future. If this information can be utilized properly then these social media sites could prove to be more than just a place to connect with friends but another way to keep our cities safe. In our analysis, we explored 4 machine learning algorithms and their performance when analyzing language while utilizing the TF-IDF NP process for the inputs, and our models produced good results but not exactly as good as we would have hoped. We believe that some of this was limited due to the data as it is difficult to match crime to Twitter data to make a predictive model which is an area that could be improved. There should be more robust data sets that actually have a direct connection between the two which could foster more accurate results. While our models can be useful for law enforcement agencies to implement into their own systems there needs to be continued research as labeling someone as criminal based on their social media use could be costly unless you are accurate.

References

[1] Y. Wang, S. Liu, and S. D. Young, "The relationship between social media data and crime rates in the ...," Sage Journals, https://journals.sagepub.com/doi/10.1177/2056305119834585 (accessed Mar. 19, 2024)

[2] N. Ashraf, D. Mahmood, M. A. Obaidat, G. Ahmed, and A. Akhunzada, "Criminal behavior identification using social media forensics," MDPI, https://www.mdpi.com/2079-9292/11/19/3162 (accessed Mar. 18, 2024).

[3] S. Tam and Ö. Ö. Tanriöver, "Multimodal Deep Learning Crime Prediction Using Tweets," IEEE explor, https://ieeexplore.ieee.org/document/10231346.

[4] E. B. CEYHAN, Ş. SAĞIROĞLU, R. CESUR, and A. KERMEN, "Determination of potential criminals in social network," Gazi University Journal of Science, https://dergipark.org.tr/en/pub/gujs/issue/28464/303378 (accessed Mar. 25, 2024).

[5] A. Tundis, A. Jain, M. Muhlhauser, and G. Bhatia, "Similarity Analysis of Criminals on Social Networks: An Example on Twitter," IEEE explor, https://ieeexplore.ieee.org/document/8847028 (accessed Mar. 25, 2024).

[6] X. Lombo, O. Olaide, and A. E.-S. Ezugwu, "Crime Detection and Analysis from Social Media Messages Using Machine Learning and Natural Language Processing Technique," ResearchGate, https://www.researchgate.net/publication/362233726_Crime_Detection_and_Analysis_from_Social_Media_Messages_Using_Machine_Learning_and_Natural_Language_Processing_Technique (accessed Mar. 26, 2024).

[7] Toronto Police, "Major crime indicators open data," Toronto Police Service Public Safety Data Portal, https://data.torontopolice.on.ca/datasets/0a239a5563a344a3bbf8452504ed8d68_0/explore?location=43.537887%2C-78.225994%2C8.68 (accessed Mar. 26, 2024).

[8] M. KazAnova, "Sentiment140 dataset with 1.6 million tweets," Kaggle, https://www.kaggle.com/datasets/kazanova/sentiment140 (accessed Mar. 26, 2024).

[9] "Stats about all US cities - real estate, relocation info, crime, house prices, cost of living, races, home value estimator, recent sales, income, photos, schools, maps, weather, neighborhoods, and more," City, https://www.city-data.com/ (accessed Mar. 26, 2024).

[10] S. Sadiq, "Cyber troll dataset," Zenodo, https://zenodo.org/records/3665663 (accessed Mar. 26, 2024).

[11] A. Simha, "Understanding TF-IDF for Machine Learning," Capital One, https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/ (accessed Mar. 26, 2024).

[12] R. Kulshrestha, "Latent dirichlet allocation(lda)," Medium, https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2 (accessed Mar. 26, 2024).

[13] S. E. R, "Understand random forest algorithms with examples (updated 2024)," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (accessed Mar. 26, 2024).

[14] A. Saini, "Guide on Support Vector Machine (SVM) algorithm," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/ (accessed Mar. 26, 2024).

[15] I. Muzaffar, "What are LSTM models?," Educative, https://www.educative.io/answers/what-are-lstm-models (accessed Mar. 26, 2024).

[16] Statistical Modeling, "What are the advantages and disadvantages of using random forests for natural language processing tasks?," Random

Forests for NLP: Pros and Cons, https://www.linkedin.com/advice/3/what-advantages-disadvantages-using-random (accessed Mar. 26, 2024).

[17] E. D. Canedo and B. Mendes, "Software Requirements Classification Using Machine Learning Algorithms," ResearchGate, https://www.researchgate.net/publication/344331184_Software_Requirements_Classification_Using_Machine_Learning_Algorithms (accessed Mar. 26, 2024).

[18] P. Srivatsavaya, "LSTM-implementation, advantages and diadvantages," Medium, https://medium.com/@prudhviraju.srivatsavaya/lstm-implementation-advantages-and-diadvantages-914a96fa0acb (accessed Mar. 26, 2024).

[19] *Text mining package [R package TM version 0.7-12]* (2024) *The Comprehensive R Archive Network*. Available at: https://cran.r-project.org/package=tm (Accessed: 14 April 2024).

[20] Chowdhury, K.R. (2020) *TF-IDF using R, Medium*. Available at: https://medium.com/@kounteyo1998/tf-idf-using-r-19fe750a9d15 (Accessed: 14 April 2024).

[21] *Package randomforest* (no date) *CRAN*. Available at: https://cran.r-project.org/package=randomForest (Accessed: 14 April 2024).

[22] *GLM: Fitting generalized linear models* (no date) *RDocumentation*. Available at: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm (Accessed: 14 April 2024).

[23] Le, J. (2018) *Support Vector Machines in R tutorial*, *DataCamp*. Available at: https://www.datacamp.com/tutorial/support-vector-machines-r (Accessed: 13 April 2024).

[24] Finnstats (2021) *LSTM network in R: R-bloggers*, *R*. Available at: https://www.r-bloggers.com/2021/04/lstm-network-in-r/ (Accessed: 13 April 2024).

[25] *Classification models: Top 10 evaluation metrics for classification* (2023a) *Explorium*. Available at: https://www.explorium.ai/blog/machine-learning/top-10-evaluation-metrics-for-classification-models/ (Accessed: 14 April 2024).