Jurnal Teknik Informatika (JUTIF)
Vol. 4, No. 3, June 2023, pp. 611-618

DOI: https://doi.org/10.52436/1.jutif.2023.4.3.868
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IDENTIFYING POSSIBLE RUMOR SPREADERS ON TWITTER USING THE SVM AND FEATURE LEVEL EXTRACTION

Claudia Mei Serin Sitio[1], Yuliant Sibaroni[2], Sri Suryani Prasetiyowati[3]

[1,2,3]Informatics, School of Computing, Telkom University, Bandung, Indonesia
Email: [1]serinsitio@student.telkomuniversity.ac.id, [2]yuliantsibaroni@telkomuniversity.ac.id, [3]srisuryani@telkomuniversity.ac.id

(Article received: January 29, 2023; Revision: May 24, 2023; published: June 26, 2023)

Abstract

In everyday life, many events occur and give rise to various kinds of information, which are also rumors. Rumors can cause fear and influence public opinion about the event in question. Identifying possible rumor spreaders is extremely helpful in preventing the spread of rumors. Feature extraction can be done to expand the feature set, which consists of conversational features in the form of social networks formed from user replies, user features such as following, tweet count, verified, etc., and tweet features with text analysis such as punctuation and sentiment values. These features become instances used for classification. This study aims to identify possible spreaders of rumors on Twitter with the SVM classification model. This instance-based classification algorithm is good for linear and non-linear classification. In the non-linear classification, additional kernels are used, such as linear, RBF, and sigmoid. The research focuses on getting the best model with high performance values from all the models and kernel functions that have been defined. It was found that the SVM classification model with the RBF kernel has a high overall performance value for each data combination with a ratio of the amount of data is 1:1 or the difference is very large. This model gives accurate results with an average of 97.02%. With a wide distribution of data, the SVM classification model with the RBF kernel is able to map the data properly.

Keywords: *account, feature-level extraction, rumors, spreaders, Support Vector Machine, Twitter*.

1. INTRODUCTION

Information is something that is needed by the public . The dissemination of information is very quickly carried out through various media [1]. Social media is one of the media for conveying information that is currently often used. Information has an important role in life because it can influence the actions of society[2].

Many things happen in daily activity, ranging from good phenomena to bad incidents. In the digital era, every information incident can be known very quickly because of the development of social media among the public. Events that become hot topics will be increasingly discussed through social media often used by the public, such as Twitter [2], [3].

The social media of Twitter allows its users to upload tweets and engage with other users through features such as replies and retweets. The ease of interaction built into Twitter means that various kinds of information can be spread without knowing the facts and truth. This resulted in the emergence of rumor spreaders on Twitter [4], [5].

Twitter accounts have many features representing these users' information, such as the number of followers and tweets, verified status accounts, tweets, favorites, and so on. For account identification on Twitter, many are done to detect this information, such as bots or *spammers* as well as *buzzers*. Classification is carried out on existing features of the Twitter dataset with user profiles, social network, and tweet text features. Buzzer detection was performed by expanding the account property features in the form of mean, quartile, and range values of the existing features, such as followers, following, and others [4], [5].

There is some research on rumor detection on Twitter. Those research usually used machine learning techniques, one of which is with a supervised learning approach as in research conducted by Manita Maan[2] and Monu Waskale[6] has used classification algorithms, such as Random Forest and Support Vector Machine (SVM). Both studies analyzed large amounts of tweet copy data to identify and classify rumors. Some common approaches included using features such as sentiment analysis, linguistic patterns, and network analysis to identify rumors and track their spread.

Other studies focusing on Twitter accounts were also conducted with the same approach as detection, such as bots, spammers[7], fake news spreaders, and buzzers. These approaches are analyzed based on user features, such as followers, following, hashtags, URLs, and other features. In spammer detection research[4], classification was carried out with user profile features, social network (reply), and content. Bot detection on Twitter[8] was done to detect two categories of bots, namely, good and bad. Detection

with the expansion of user features is also done to detect Twitter buzzers[9]. Fake news spreaders were also profiled with behavioral analysis on Twitter regarding the 2016 United States presidential election[10]. Since the case was spread with fake news, it also needs identification to deal with the expansion of the spread of fake news and rumors. The identification of rumor spreaders themselves is done little. There are characteristics of spreader rumors that have been analyzed by Bodaghi[11]. On the other hand, Bhavtosh Rath[12] identified the rumor spreaders by utilizing beliefs using RNN. Rumor spreaders identification was also carried out by Shakshi Sharma[13] where the approach used was Supervised Learning with Graph Convolutional Network (GCN) techniques compared to algorithms such as SVM, RF, and LSTM. The GCN model gave an F1-score of 86.4%.

Unlike bots or *spammers*, rumor spreaders need a feature to identify the account, including spreaders or non-spreaders. In this final project, the author identifies the 'possibilities' rumor spreaders on Twitter. The researcher defines them as users who frequently upload rumor tweets, so it is 'possible' that users can become rumor spreaders.

The addition of such intended features resulted in the need for data of users who uploaded tweets multiple times to see their intensity. So, the researcher used the PHEME dataset of nine incidents between 2014 and 2015[4]. There have been five incidents that have data that can be used to identify rumor spreaders on Twitter, namely i) *Charlie hebdo*, ii) *Ottawa shooting*, iii) *Germanwings crash*, iv) *Sydney siege*, and v) *Ferguson*.

The PHEME dataset is converted into a rumor spreaders dataset by extracting tweets and user accounts and calculating intensity values (how often users upload the rumor tweets). The labeling on this approach used *near ground truth* which will only identify accounts that 'might' be rumor spreaders due to the limited appropriate approach.

Our purpose on this research is to identify the 'possibility' of rumor spreaders on Twitter using the SVM classification model. The model will be evaluated with performance measure results of accuracy, precision, recall, and F1-score values. The model can be stated to identify the 'possibility' of rumor spreaders on Twitter.

2. RESEARCH METHOD

The PHEME dataset is the data that have been structured and divided into a directory for each event with two sub-folders, namely rumors and non-rumors. The folder consists of the tweets that are the source and the reactions respond to the tweets. Figure 1 shows the flow of the built system. Pre-processing data was performed to change the shape of the dataset based on tweets to a dataset based on the account of the disseminator.

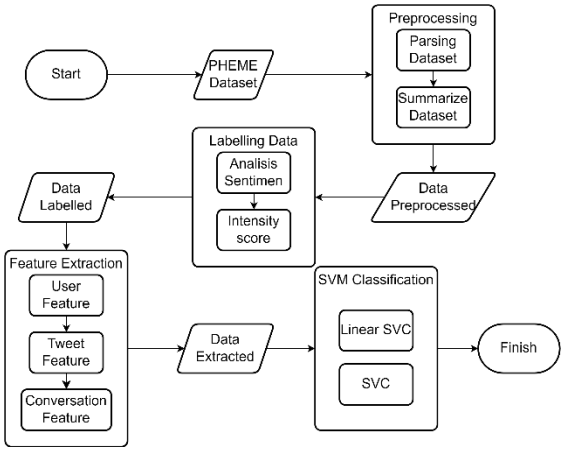

Figure 1. System Flow.

2.1. Datasets

The dataset was collected from the PHEME dataset from the research by Arkaitz Zubiaga[18]. The dataset consists of five events that were widely reported and attracted media attention in 2014 - 2015. The five events involved Charlie Hebdo, Ferguson, Germanwings Crash, Ottawa Shooting, and Sydney Siege which have been annotated as rumors and non-rumors. The data were stored in a JSON format consisting of the source of the tweet and the reaction to the tweet.

Table 1 shows the sharing of rumor and non-rumor annotations for the five events on the dataset.

Table 1. PHEME Dataset Distribution

Events	Rumors	Non-rumors	Total
Charlie Hebdo	458 (22%)	1621 (78%)	2079
Ferguson	284 (24.8%)	859 (75.2%)	1143
Germanwings Crash	238 (50.7%)	231 (49.3%)	469
Ottawa Shooting	470 (52.8%)	420 (47.2%)	890
Sydney Siege	522 (42.8%)	699 (57.2%)	1221
Total	1972 (34%)	3830 (66%)	5802

2.2. Pre-processing

This pre-processing stage was carried out to transform the PHEME dataset into a rumor-spreading dataset used in this study.

1. Data Parsing

Data parsing is converting data captured in one particular format into another. The PHEME dataset was formed in JSON format based on the source tweets. Parsing data transformed the data and combined all tweets into CSV-formatted rows of data for each event.

2. Summarize the Data

Summarizing data was done to prepare data for the efficiency of system work when processing the required data. At this stage, the data were filtered based on the tweet copy feature, which consisted of tweets that count punctuation marks and have dot symbols, counting positive and negative words, and sentiment values of the text.

This stage provides results in the form of data with filtered features, namely user features and tweet features, as shown in Table 2.

Table 2. Description of Set Details Feature

Features Sets	Feature	Description
Conversation Feature	threads	Source tweet ID
	in_reply_tweet	ID tweet that replied
	events	The event name of the dataset
	tweet_id	tweet id
	is_source_tweet	Tweet which is the source
	in_reply_user	ID user that did the reply
	user_id	Twitter user id
Featured Tweets	hashtags_count	The number of hashtags on the tweet
	retweet_count	The number of retweets of the tweet
	favorite_count	The number of likes on a tweet
	mentions_count	The number of users mentioned in the tweet
User Features	tweet_count	The number of tweets uploaded by the user
	verified	The Twitter account has been verified
	followers_count	Number of user followers
	friends_count	Number of user friends

2.3. Labelling

In identifying possible rumor spreaders, the researcher took several approaches in labeling by utilizing data labels from the PHEME dataset and features that have been parsed and clean. It was started with sentiment analysis from the user by using the TextBlob API. In the dataset, it was found that the annotations of rumors and non-rumors were in line with the sentiment values of the tweets, namely negative and positive. As shown in Figure 2, at the Charlie Hebdo event, it can be seen that the positive sentiment value of the dataset with the non-rumor label is 30.68%, which is higher than the positive sentiment from the dataset with the rumor label, which is 26.71%. Meanwhile, the dataset with the rumor label also has a high negative sentiment value of 73.29%.

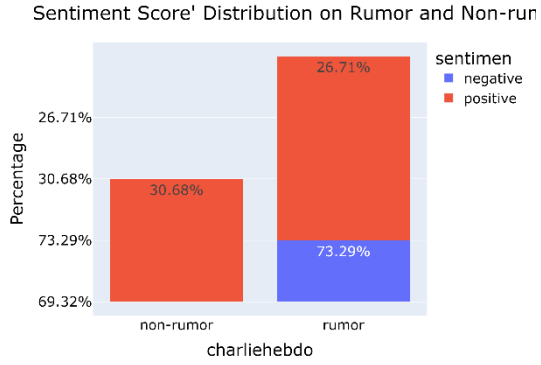

Figure 2. Sentiment Score Distribution on Charlie Hebdo Event

Figure 3 shows the Ferguson event, and it can be seen that the positive sentiment value of the dataset with the non-rumor label is also higher than the dataset with the rumor label, at 32.92% and 28.96%, respectively. Meanwhile, the dataset with the rumor label also has a high negative sentiment value of 71.04%.

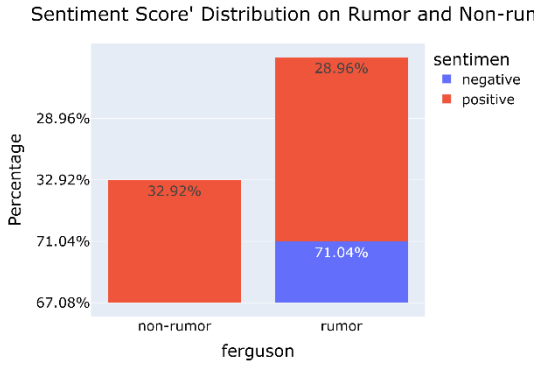

Figure 3. Sentiment Score Distribution on Ferguson Event

At the Germanwings crash event as shown in Figure 4, it can be seen that the positive sentiment value of the dataset with the non-rumor label is 24.81%, and the positive sentiment from the dataset with the rumor label is 23.90%. Then, the dataset with the rumor label also has a high negative sentiment value of 76.10%.

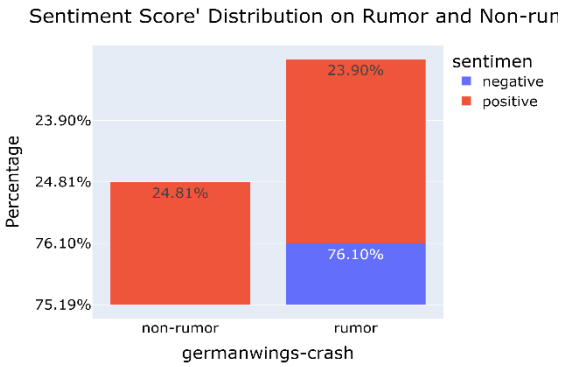

Figure 4. Sentiment Score Distribution on Germanwings crash Event

Figure 5 shows the Ottawa shooting event, and it can be seen that the positive sentiment value of the dataset with the non-rumor label is also higher than the dataset with the rumor label, at 33.98% and 28.28%, respectively. Meanwhile, the dataset with the rumor label also has a high negative sentiment value of 71.72%.

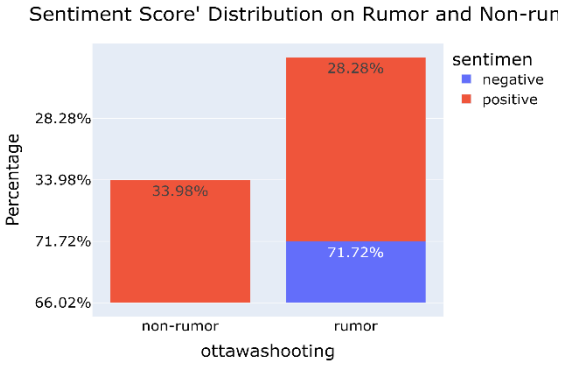

Figure 5. Sentiment Score Distribution on Ottawa shooting Event

At the Sydney siege event as shown in Figure 6, it can be seen that the positive sentiment value of the dataset with the non-rumor label is 32.82% and from the dataset with the rumor label is 30.95%. Then, the dataset with the rumor label also has a high negative sentiment value of 69.05%.

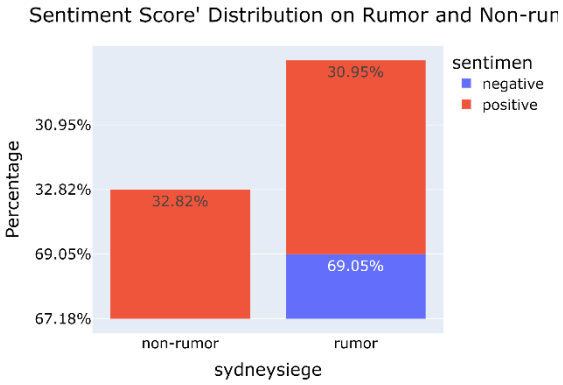

Figure 6. Sentiment Score Distribution on Sydney siege Event

The rumor data are dominated by negative sentiment. Otherwise, the non-rumor data are dominated by positive sentiment. So, the sentiment value can be used as validation for the identification of possible rumor spreaders.

In order to identify possible rumor spreaders, it was also carried out by calculating the intensity value of users spreading rumors on Twitter. The calculation is carried out by equation (6)[4].

$$Score = \frac{\#of\ times\ user\ tweets\ rumor}{Total\ \#\ of\ times\ user\ tweets} \quad (6)$$

The result of calculating the intensity value is between [0,1], where 0 indicates a non-spreader of rumors, and 1 indicates the possibility of rumor spreaders. Amirhosein Bodaghi's research[13] which analyzed the characteristics of rumor spreaders stated that most users are only once involved in the process of spreading rumors and not repetitive activity. Then, a comparison is also made for the sentiment value. So the following comparison is made:

i) if the intensity value < 0.5 and the sentiment value > 0 (positive), labeled 0;
ii) if the intensity value < 0.5 and the sentiment value < 0 (negative), labeled 1;
iii) if the intensity value > 0.5 and the sentiment value > 0 (positive), labeled 1;
iv) if the intensity value > 0.5 and the sentiment value < 0 (negative), labeled 0.

2.4. Feature Extraction

This feature extraction process was done to expand the properties of the user, tweet, and conversation features. User features and tweets used aggregate and statistical functions in each feature. The function consists of mean value (*mean*), total (*sum*), and variance (*var*) to calculate the distribution of data. The calculation is carried out by equations (7) and (8)[19].

$$Mean = \bar{X}\frac{\sum_i^N X_i}{N} \quad (7)$$

$$Var = \frac{1}{N}\sum_i^N (X_i - \bar{X}) \quad (8)$$

Description:
N : amount of data,
X_i : data i

The features additional was done from the previously named features plus each calculation as shown in Table 3

Table 3. Feature Extraction Results

Feature ID	Featured	Means	Sum	Var
F1	favorite_count	F1_mean	F1_sum	F1_var
F2	retweet_count	F2_mean	F2_sum	F2_var
F3	hashtags_count	F3_mean	F3_sum	F3_var
F4	tweet_count	F4_mean	F4_sum	F4_var
F5	is_rumor	Nan	F5_sum	Nan
F6	hasperiod	F6_mean	F6_sum	F6_var
F7	number_punct	F7_mean	F7_sum	F7_var
F8	negative wordcount	F8_mean	F8_sum	F8_var
F9	PositiveWord Count	F9_mean	F9_sum	F9_var
F10	sentiment score	F10_mean	F10_sum	F10_var

The conversation feature is used to calculate the user's network by adding the largest user-to-user conversation size of the dataset. Correlation between users are calculated by building graphs based on interactions in the form of replies. The result of this additional feature is the largest number of network components and diameters.

2.5. SVM Classification

After the data feature has been extracted, then the data were normalized first to create the numerical data on the dataset and have the same range of values (scale). This process is performed on the X data feature by using equation (9)[19].

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (9)$$

Description:
X_{max} : maximum data X
X_{min} : minimum data X

The normalized data was then divided into train data and test data. Training data is used to train *classifiers* to recognize the characteristics of users who may be rumor spreaders and non-rumor

spreaders. Data testing was used in trials of the resulting classification model and determined the performance of the classification model by comparing the results of the model classification on each data in testing data with actual labels.

The classification model used in this study is the Support Vector Machine (SVM). SVM is a supervised learning algorithm with the idea of describing a line that will divide the data into two classes[15]. SVM aims to find the optimal dividing hyperplane by maximizing the margin (gap distance) of the training data[16]. Where the margin is twice the distance between the hyperplane and the nearest data point of each class. So, there will not be any data points in the margin.

SVM, which is an *instance-based* approach, performs linear and non-linear classifications or can be quadratic, cubic, and higher-order equations[17]. The SVM then calculated the value of the divisor hyperplane called the hyperplane soft margin with equation (1)[16].

$$g : (w^T x) + b = 0 \quad (1)$$

Then the calculation of the distance to the nearest data point of each class is g. Distances are defined by equations (2) and (3)[16].

$$\left(w^T x^{(i)}\right) + b \geq 1 \; to \; y = 1 \quad (2)$$

$$\left(w^T x^{(i)}\right) + b \leq 1 \; to \; y = -1 \quad (3)$$

The optimal hyperplane will have the largest margin because it classifies the train data into the correct class and is generally good for unseen data. The determination of the best *hyperplane* is done by maximizing the margin of the training data by minimizing the numerator in the margin formula of equation (4)[16].

$$Ar\ g\ Min\ \frac{1}{2} - w^T w \quad (4)$$

$$s.t.\left(w^T x^{(i)} + b\right) y_i \geq 1, i = 1,2, \dots, N$$

Description:
x : data points
w : vector parameters
b : field scalar

In higher dimensions, it is non-linear by converting the *dot product* input into *feature space* (Φ) i.e. the Kernel function in equation (5)[17].

$$K(x, y) = \Phi(x) \cdot \Phi(y) \quad (5)$$

This study used three kernels, namely linear, Radial Basis Function (RBF), and sigmoid.

In SVM, there are several kernel functions, namely Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid. In this study, the classification was carried out using the *Scikit-Learn* library, namely the SVC and Linear SVC functions.

2.6. Evaluation Metrics

In this study, the model performance measurements were carried out based on F1-score, precision, recall, and accuracy. F1-score was derived from harmonic mean of precision and recall. The best score taken from F1-score is 1 and the worst score is 0. Equations (10), (11), (12) and (13) show the calculation of performance values based on the confusion matrix[19].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

$$\frac{1}{F1} = \frac{1}{2}\left(\frac{1}{Precision} + \frac{1}{Recall}\right) \quad (13)$$

Where TP (true positive) is positive prediction data and positive actual data, FP (false positive) is positive prediction data and negative actual data, and FN (false negative) is negative prediction data and positive actual data. These predictive and actual classifications are set forth in a commonly used matrix for binary classification[20]. More details can be seen in

Table 4.

Table 4. Confusion Matrix

		Actual Values	
		Positives	Negatives
Predict Values	Positives	TP	FP
	Negatives	FN	MR

3. RESULT AND DISCUSSION

In this study, a test scenario was carried out by creating a combination of train data and test data from each event. Therefore, a comparison of the performance size of the model for all combinations of event train data one and another event test data was obtained. The created combination consists of:

1) Data Train and Data Test at *Charlie Hebdo* events
2) Train Data and Test Data on *Germanwings Crash* event
3) Data Train on *Germanwings Crash* and Data Test on *Charlie Hebdo* event
4) Data Train on *Germanwings Crash* and Data Test on *Ottawa Shooting* event
5) Data Train on *Ottawa Shooting* event and *Ferguson* event

The five combinations above produce test results as shown in

Table 5.

Table 5. Combination Results of Data Train and Charlie Hebdo Test Data

Classification Models	Accuracy	Precision	Recall	F1-score
LinearSVC	99.02%	96.74%	99.40%	98.05%
SVM, kernel=linear	**99.40%**	**97.91%**	**99.74%**	**98.82%**
SVM, kernel=RBF	97.23%	89.82%	99.26%	94.30%
SVM, kernel=sigmoid	92%	78.97%	88.49%	82.46%

The test results as shown in Table 5 show the classification results with a combination of *Charlie Hebdo* event train data. In this combination, overall accurate results were obtained from accuracy, precision, recall, and F1-score for all SVM classification models tested. This is because the same data size is 38268 row data. Thus, the data were splitted and recombined to produce balanced data. *Germanwings Crash* as much 4489 row data.

Table 6. Combination Results of Data Train and Data Test Germanwings Crash

Classification Models	Accuracy	Precision	Recall	F1-score
LinearSVC	**99.86%**	**99.52%**	**100%**	**99.76%**
SVM, kernel=linear	99.32%	97.61%	100%	98.79%
SVM, kernel=RBF	94.96%	83.25%	98.86%	90.39%
SVM, kernel=sigmoid	92.51%	80.38%	92.31%	85.93%

Similar to the combination of training data and test data on *Charlie Hebdo*, this combination in the *Germanwings Crash* event also results in high accuracy of all models, up to 99.86% using the LinearSVC model as shown in Table 6. The amount of data will affect the performance of the model where in *Germanwings Crash*, there are 4489 *rows* of data. *Germanwing Crash* event data, which is much less than *Charliue Hebdo* event data, provides higher performance results.

Table 7. Combination Results of Germanwings Crash Train Data and Charlie Hebdo Test Data

Classification Models	Accuracy	Precision	Recall	F1-score
LinearSVC	57.26%	**99.96%**	37.44%	54.48%
SVM, kernel=linear	57.95%	97.96%	37.82%	54.88%
SVM, kernel=RBF	**97.12%**	92.62%	**96.02 %**	**94.29 %**
SVM, kernel=sigmoid	90.04%	85.00%	73.17%	78.65%

In Table 7, when the train data from one event is compared with the other event test data, it provides different performance values. As in the training data from the *Germanwings Crash* event and the test data from the *Charlie Hebdo* event, it provides different performance values for each classification model. Both events provided test data accuracy results of 57.26% for the LinearSVC classification model; 57.26% for SVM classification models with linear kernels; 97.12% for SVM classification model with RBF kernel; and 90.83% for SVM classification model with sigmoid kernel. This difference in accuracy values can be caused by the vast difference in data size between the *Germanwings Crash* event and the *Charlie Hebdo* event. Where the size ratio is about 1:6. So that when data is split, between the training data and the test data is not balanced.

Table 8. Combination Results of Germanwings Crash Train Data and Ottawa Shooting Test Data

Classification Models	Accuracy	Precision	Recall	F1-score
LinearSVC	60.24%	**99.91 %**	40.76 %	57.90 %
SVM, kernel=linear	60.49%	**99.91 %**	40.92 %	58.05 %
SVM, kernel=RBF	**97.67%**	92.37 %	99.04 %	95.59 %
SVM, kernel=sigmoid	90.04%	85.62 %	79.55 %	82.47 %

The similar results is found in the combination of training data from Germanwings *Crash* event and test data from *Ottawa Shooting* event as shown in Table 8. The combination provided test data accuracy results of 60.24% for the LinearSVC classification model; 60.49% for SVM classification model with linear kernel; 97.67% for SVM classification model with RBF kernel; 90.04% for SVM classification model with sigmoid kernel. The size ratio is about 1:3.

Table 9. Combination Result of Ottawa Train Data Shooting and Ferguson Test Data

Classification Models	Accuracy	Precision	Recall	F1-score
LinearSVC	98.94%	97.85%	98.28%	98.06 %
SVM, kernel=linear	**99.22%**	**99.53%**	97.66%	98.58 %
SVM, kernel=RBF	97.95%	94.05%	**98.42 %**	96.18 %
SVM, kernel=sigmoid	87.94%	77.87%	78.10%	77.99 %

However, it did not happen in events with data sizes that are not much different as in the *Ottawa Shooting* event and *the Ferguson* event with a ratio of about 1:1. Table 9 show the high accuracy for the combination. The training data on the *Ottawa Shooting* event and the test data on the *Ferguson* event obtained excellent performance values which defined classification models.

4. DISCUSSION

Based on all the test result, the SVM classification model with the RBF kernel has a high overall performance value. Where for the combination of training data and test data at the *Charlie Hebdo* event gave the results of training data accuracy of 97.23%. Then, the *Germanwings*

Crash event is 94.96%. A combination of training data on *Germanwings Crash* events and test data on Charlie *Hebdo* events is 97.12%. A combination of training data on *the Germanwings Crash* events and test data on *the Ottawa Shooting* events is 97.67%. Last, a combination of training data on the *Ottawa Shooting* event and test data on the *Ferguson* event is 98.13%.

As a result, the RBF kernel function has the highest average accuracy of 97.02% when compared to the linear and sigmoid kernels. Whether it's a combination of training data and test data from the same event and it's definitely 1:1, the comparison of training data for one event with other event test data is still 1:1 and the comparison of training data for one event with other event test data with a difference as far apart as 1:6, the model provides high accuracy. This shows that the RBF kernel function is able to map data well on data with a wide distribution.

5. CONCLUSION

Based on the test results and analysis of the five combinations, the SVM classification model can be used to identify possible rumor spreaders. Feature extraction with aggregate and statistical calculations makes the SVM classification model highly accurate. The *labeling* approach with sentiment analysis and user intensity is also the cause of the high accuracy value. The experimental results show that the SVM classification model with the RBF kernel has high performance for all combinations of training data and test data on events with the same or different size ratios. The classification model with the RBF kernel gives high accuracy, precision, recall, and F1-score performance values, both for training data and test data from the same event or a comparison of the size of training data and test data from different events, namely 1:1, or the difference in comparison of the size of the training data is the test data from very different events. so that the classification model with the RBF kernel in this case works well in mapping data with a wide distribution of data.

The suggestion for further research is to multiply the dataset with *tweet* repetition from the same *user* and use all parameters in full with a balanced label comparison.

REFERENCES

[1] L. (Monroe) Meng, T. Li, X. Huang, dan S. (Kevin) Li, "Lift the veil of rumors: the impact of the characteristics of information sources on the effectiveness of rumors spreading," *Internet Res.*, vol. 32, no. 1, 2022, doi: 10.1108/INTR-11-2020-0620.

[2] I. C. Hsu dan C. C. Chang, "Integrating machine learning and open data into social Chatbot for filtering information rumor," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 1, 2021, doi: 10.1007/s12652-020-02119-3.

[3] R. Dekker, G. Engbersen, J. Klaver, dan H. Vonk, "Smart Refugees: How Syrian Asylum Migrants Use Social Media Information in Migration Decision-Making," *Soc. Media Soc.*, vol. 4, no. 1, 2018, doi: 10.1177/2056305118764439.

[4] S. Sharma dan R. Sharma, "Identifying Possible Rumor Spreaders on Twitter: A Weak Supervised Learning Approach," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2021-July, 2021, doi: 10.1109/IJCNN52387.2021.9534185.

[5] M. Maan, M. K. Jain, S. Trivedi, dan R. Sharma, "Machine Learning Based Rumor Detection on Twitter Data," *Emerg. Technol. Comput. Eng. Cogn. Comput. Intell. IoT. ICETCE 2022. Commun. Comput. Inf. Sci.*, vol. 1591, 2022, doi: https://doi.org/10.1007/978-3-031-07012-9_23.

[6] A. Kaur dan A. Sinha, "Multi-contextual spammer detection for online social networks," *J. Discret. Math. Sci. Cryptogr.*, hal. 777–786, 2021, doi: https://doi.org/10.1080/09720529.2020.1794517.

[7] B. Dixon, *Social Media for School Leaders: A Comprehensive Guide to Getting the Most Out of Facebook, Twitter, and Other Essential Web Tools*. 2012.

[8] M. Waskale dan P. Jain, "Rumors Detection on Twitter Using Machine Learning Techniques," 2019.

[9] D. Koggalahewa, Y. Xu, dan E. Foo, *An unsupervised method for social network spammer detection based on user information interests*, vol. 9, no. 1. Springer International Publishing, 2022. doi: 10.1186/s40537-021-00552-5.

[10] A. Ramalingaiah, S. Hussaini, dan S. Chaudhari, "Twitter bot detection using supervised machine learning," *J. Phys. Conf. Ser.*, vol. 1950, no. 1, 2021, doi: 10.1088/1742-6596/1950/1/012006.

[11] Yuliant sibaroni dan Sri Suryani Prasetiyowati, "Buzzer Detection on Indonesian Twitter using SVM and Account Property Feature Extension," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, hal. 663–669, 2022, doi: 10.29207/resti.v6i4.4338.

[12] M. Cardaioli, S. Cecconello, M. Conti, L. Pajola, dan F. Turrin, "Fake News Spreaders Profiling through Behavioural Analysis Notebook for PAN at CLEF 2020," *CEUR Workshop Proc.*, vol. 2696, no. September, hal. 22–25, 2020.

[13] A. Bodaghi dan J. Oliveira, "No The characteristics of rumor spreaders on Twitter: A quantitative analysis on real data," *Comput. Commun.*, vol. 160, hal. 674–687, 2020, doi: https://doi.org/10.1016/j.comcom.2020.07.017.

[14] B. Rath, W. Gao, J. Ma, dan E. Al., "Utilizing computational trust to identify rumor spreaders on Twitter," *Soc. Netw. Anal. Min.*, 2018, doi: https://doi.org/10.1007/s13278-018-0540-z.

[15] V. Piccialli dan M. Sciandrone, "Nonlinear optimization and support vector machines," *Ann. Oper. Res.*, vol. 314, no. 1, hal. 15–47, 2022, doi: 10.1007/s10479-022-04655-x.

[16] V. Vapnik, *The Nature of Statistical Learning Theory*. 1995. doi: http://dx.doi.org/10.1007/978-1-4757-2440-0.

[17] S. Džeroski, *Data Mining*. 2008. doi: 10.1016/B978-008045405-4.00153-1.

[18] A. Zubiaga, M. Liakata, dan R. Procter, "Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media," 2016.

[19] S. Agarwal, *Data mining: Data mining concepts and techniques*. 2014. doi: 10.1109/ICMIRA.2013.45.

[20] D. Valero-Carreras, J. Alcaraz, dan M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, no. December 2022, hal. 106131, 2023, doi: 10.1016/j.cor.2022.106131.