Fooling news misinformation classifiers by generating adversarial attacks without altering the semantics of the text

Bachelor Artificial Intelligence
Utrecht University

Matthijs Hornix
m.j.m.hornix@students.uu.nl

A bachelor thesis in the field of
Artificial Intelligence (7,5 ECTS)

Faculty of Humanities
Supervisor: Nguyen, D.P.
Second reader: Prof. Dr. Prakken, H
June 2020

Contents

1 Introduction **2**

2 Related work **3**

3 Methodology **4**

3.1 Dataset 4

3.2 Classifiers 4

3.2.1 Naive Bayes . . . 5

3.2.2 Support vector machine 5

3.2.3 Feed-forward neural network . 5

3.2.4 Long short-term memory 5

3.3 Textual adversarial attacks 6

4 Results **6**

5 Discussion **7**

5.1 Implications 7

5.2 Shortcomings 8

5.3 Future research 9

6 Conclusion **9**

7 Appendix **12**

7.1 Synonym list 12

7.2 Confusion matrices . . . 13

7.3 Accuracy, recall, precision, and F1 scores . . . 14

Abstract

This research looks at the robustness of four different machine learning models (Naive Bayes, SVM, feed-forward neural network, and LSTM), applied to a true or false news classification task. The four classifiers were trained on the same training set but tested on both a regular and altered test set. The altered test set had words replaced with synonyms to investigate if the classifiers were susceptible to semantics-preserving alterations. Naive Bayes based true or false news classification was not able to perform well enough to say it acquired an “understanding” of the content, thus no unambiguous answer could be given, regarding Naive Bayes. SVM and the feed-forward neural network classifiers showed no differences in scores and thus are likely to be insusceptible to the few alterations made. Long short-term memory, however proofed to be susceptible to the alterations and should thus not be implemented as an automatic news classification system, for the classification can be altered without altering the semantics of the input. Even though the SVM and feed-forward neural network classifier’s scores remained unchanged, more research is needed to give definitive answers, regarding the classifiers.

1 Introduction

Vaccines do not cause autism, the general public agrees, however there are people who still believe the retracted article by Wakefield et al. (1998). This article shows the power of misinformation, deliberate or not. Nowadays people are connected to a virtually endless knowledge base and misinformation remains a problem. Misinformation was recently popularized under the term "fake news" during the 2016 US presidential elections and has been, and still is, a hot topic to this day.

Nowadays Facebook alone has more than 2.6 billion monthly active users (Clement, 2020), each capable of sharing their thoughts, factual or not, with the world. With this immense number, that continues to grow, it is essential to raise awareness of potentially false news. This could be done, by for instance, showing a warning sign.

To counter the distribution of misinformation, machine learning systems are being implemented and researched to put an end to this, as is clear from the survey of research conducted by Zhou & Zafarani (2018). They saw that research is mainly focused on one or more of four aspects: false knowledge it carries, writing style, propagation patterns, and credibility of the creators and spreaders.

These four areas have received attention, however an underexplored but important area of research is the robustness of automatic true or false news classifiers. It is critical to make sure there are no possibilities to "fool" the system into misclassifying a truthful article as false, or worse, a false article as truthful.

One of the aspects highlighted by Zhou & Zafarani (2018) is writing style, meaning the machine learning system has learned to base their classification on stylometry features. Stylometry features for example could for example be word choice, sentence structure, etc. There are probably some writing style aspects that correlate with the truthfulness of the text. For example, some low quality news outlets might more often publish misinformation, and perhaps their writing style is somewhat different than the writing style you tend to find in more mainstream outlets (like NYTimes). But even with these correlations, the writing style itself is not a valid signal to say whether a text is true or false. A research that focused, among other things, on this aspect of misinformation detection was done in 2017 by Potthast et al. (2018b). They found that their classifier was able to attain scores of $F_1 = 0.46$ using stylometry features alone. Another study conducted by Curci et al. (2018) looked at the applicability of various machine learning systems to score the reliability of an article's source. They were rather successful as their machine learning systems attained accuracy's between 72.94% and 94.53%.

Although these studies sound very promising these numbers may be misleading, since the above mentioned scores were achieved without fact-checking the content against a knowledge base. Thus the classifiers have learned to classify articles based on something that is not the truthfulness of the content.

Semantics-preserving adversarial attacks To test whether such automatic classification systems can be "fooled", we can look at adversarial attacks. An adversarial attack is where an input is modified in such a way that, to a human, the input seems the same, however the machine learning system misclassifies the input. In order to successfully create textual adversarial attacks, we need to alter the input text without changing the semantics of the text. For example the sentences "President Trump won the 2016 elections" and "Donald Trump won the 2016 United States presidential elections" mean, to a human reader, the same. But it could be that these sentences are classified differently by a machine learning system. It is thus important to make sure that these systems are not susceptible to these alterations. This raises the question: **Can a news misinformation classifier's scores be altered by altering the style of the input text?**

Approach In order to test this, I will investigate four different machine learning classifiers that are trained on a dataset of news articles. Then two different test sets of news articles will be used. The contents of the sets will be semantically the same, however one will be altered by substituting words with synonyms. If any of the classifiers is susceptible to these alterations, that would mean that the classifier can be "fooled", which makes the "fooled" classifier unreliable for this task and should thus not be implemented in systems that target this problem.

2 Related work

Adversarial attacks are a well established phenomenon in computer vision research. An adversarial attack is an input altered in such a way that the classifier is led to believe the given input is something else (Kurakin et al., 2017). A clear example of such an attack, which also shows the importance of research into this topic, is the research conducted by Eykholt et al. (2018). They were able to make a trained deep feed-forward neural network (DNN) misclassify a stop sign as a 45 miles per hour sign by applying a few black and white patches to the stop sign. These few alterations were enough to "fool" the DNN classifier.

For text this is more difficult as humans notice substituted words and so for textual adversarial attacks the aim is to change style features whilst keeping the semantics (nearly) identical. Think of typos, synonyms, and sentence structure. These aspects can all be changed so that the sentence is semantically the same, however a computer might classify them differently.

The power of these attacks is clearly shown by Gao et al. (2018); they were able to lower the scores of their spam email machine learning system from 99% accuracy to 40% accuracy and their sentiment analysis machine learning system from 87% accuracy to 26% accuracy. A few of their transformations were to substitute a letter with another, to swap two letters with each other, to delete a letter from the word, and to insert a letter into the word. These transformations are all relatively common typos. So it is not far fetched to think of this happening and thus machine learning systems should not be as susceptible to these as they are, according to Gao et al. (2018).

Much like Gao et al. (2018) used typos to "fool" classifiers, Ebrahimi et al. (2018) used a similar technique.

They created textual adversarial attacks by flipping characters, meaning they swapped one token for another. Similarly Samanta & Mehta (2017) proposed fooling text classifiers by replacing or deleting words that are important for the meaning of the text. This approach is another way to possibly make a text classifier misclassify inputs.

From the above mentioned research it becomes evident that it is critical to continue research in the field of textual adversarial attacks, as it is one way of expanding our knowledge of text classifiers for various applications. This research aims at yet another angle of attack. I will create synonym based adversarial attacks to test whether it is possible to "fool" the classifiers.

3 Methodology

3.1 Dataset

The used dataset is the BuzzFeed-Webis Fake News Corpus 2016 (Potthast et al., 2018a). The choice for this dataset was made because the dataset consists of articles from various left, mainstream, and right publishers as well as being a well-known misinformation dataset for research purposes. The articles in the dataset were gathered by posts made in the same week and were all fact-checked by professional journalists. The dataset consists of articles that are one of four categories:

1. Mostly true
2. Mixture of true and false
3. Mostly false
4. No factual content

For this research, where I look at machine learning to classify false or truthful news, I will only use the articles that are "mostly true" and "mostly false".

After filtering the dataset down to the "mostly true" and "mostly false" categories, the dataset consists of a total of 1415 articles. As Table 1 shows, the dataset is biased towards the "mostly true" category as opposed to the "mostly false" category. The average word count per article is 572.9 words, whereas articles categorized as "mostly true" had an average word count per article of 588.7 and articles categorized as "mostly false" had an average word count per article of 440.7. Meaning the difference between the averages is $588.7 - 440.7 = 148$, thus showing that articles belonging to the "mostly true" category tend to be significantly longer than articles categorized as "mostly false".

3.2 Classifiers

I will investigate four classifiers that are often implemented in natural language processing (NLP) systems. These four classifiers work fundamentally different, generating results from diverse machine learning models.

The classifiers have access to the document's vector representations calculated by applying Doc2Vec to the texts. Doc2Vec is an algorithm that translates sentences/paragraphs/documents into multidimensional arrays, also called vector representations (Le & Mikolov, 2014). The idea of Doc2Vec, like Word2Vec, is that textual content can be compared to other textual content and, in the case of Doc2Vec, sentences/paragraphs/documents that

Table 1: Statistics of the Buzzfeed-Webis Fake News Corpus.

	Number of articles	**Average number of words**
Total	1415 (100%)	572.9 (0%)
"mostly true"	1264 (89.3%)	588.7 (+2.8%)
"mostly false"	151 (10.7%)	440.7 (-23.1%)

are similar, have vectors that don't differ too much from each other (Mikolov et al., 2013). In this implementation of Doc2Vec an array of length 300 is created for each article.

3.2.1 Naive Bayes

Naive Bayes classifiers rely on probability by applying Bayes' theorem. Naive Bayes classifiers are often used as baseline machine learning models and as stated by Zhang (2004) "Naive Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining".

3.2.2 Support vector machine

Support vector machine (SVM) classifiers use hyperplanes to classify data points. SVMs create an n-dimensional line that separates the various data points into one of two sides of the line. SVMs are often used in text classification as well as image classification because an SVM does not need as much data as other machine learning models to attain high accuracy scores (Joachims, 1998).

3.2.3 Feed-forward neural network

Feed-forward neural networks consist of a number of nodes, each operating in parallel and learning and adapting from experience. Every feed-forward neural network consists of multiple layers. The first being the input layer, second being a number of hidden layers, and lastly the output layer, which gives the result (the prediction) (Specht et al., 1991). The implemented (TensorFlow based) feed-forward neural network, originally used by Curci et al. (2018), consists of an input layer, three dense layers consisting of 300 nodes, three dropout layers with a 40% dropout rate, which finally is fed into a layer from which the predictions are extracted. The structure is as follows: input layer → dense layer → dropout layer → dense layer → dropout layer → dense layer → dropout layer → logits layer → prediction. TensorFlow is an interface, built by Google, for executing machine learning algorithms and is often used in programming machine learning models. These models can then be used for research and real world deployment. TensorFlow is widely used in various fields, such as speech recognition, computer vision, natural language processing, etc. (Abadi et al., 2015).

3.2.4 Long short-term memory

Long short-term memory (LSTM) classifiers are a type of recurrent feed-forward neural networks (RNNs), popular in NLP applications. Proposed by Hochreiter & Schmidhuber (1997) LSTM was a revolution in the field of recurrent feed-forward neural networks. Their LSTM implementation was able to attain a speed and complexity per time step of O(1) along with having more successful runs than a number of related machine learning models. LSTM in NLP uses word order to process input texts. The implemented LSTM classifier, originally used by Curci et al. (2018), employs one hundred nodes and an output layer with a sigmoid activation function.

3.3 Textual adversarial attacks

In the field of textual adversarial attacks little research has been done in comparison to visual adversarial attacks (Akhtar & Mian, 2018). Thus this research aims to extend the knowledge in this relatively untouched field of research.

Table 2 shows two examples of the dataset's original texts and their respective alterations.

To create textual adversarial attacks the given inputs of the test set need to be altered in such a way that the semantics of the texts are left unscathed, but the sentence structure, word choice, or something similar is altered. To create textual adversarial attacks the choice was made to replace words in the main text of the articles by their synonyms, as is clear from example 1 and 2. This list of synonyms (7.1 Synonym list) was made by hand, based on commonly used words in English and the most common words in the articles' main texts. The synonym list also contains a number of contractions. Whenever multiple synonyms were possible the choice was made to select only the synonym that works best in multiple contexts. This was done to ensure the context would not alter the semantics of the text. The synonym list was created by comparing synonym suggestions from *What is WordNet?* (n.d.) and *The world's favorite online thesaurus!* (n.d.). The synonym list was created keeping in mind each synonym having as little contextual variation as possible, this in order to minimize any accidental alterations to the semantics of the texts. The final synonym list (7.1 Synonym list) contained 34 words and their respective synonyms. Among these synonyms are also 17 contractions as these are identical.

As is clear from the examples, the sentences are altered by replacing words by their synonyms, if the word occurs in the synonym list and a space is preceding and succeeding the word in question. This has been done to ensure no accidental occurrences of words as subpart of another word is replaced.

4 Results

The four classifiers have been trained on the regular train set and then tested against both the regular test set and the altered test set. The altered test set being the set of data where the software has altered the words, but not the semantics, of the main texts.

Table 3 shows the performance of the four different classifiers. Differences in scores are observed for the Naive Bayes classifier and the LSTM classifier, with their scores differing +3.13% and -5.81%, respectively. Notably is that LSTM was the best classifier on the regular test set, however after altering the test set LSTM became the second worst classifier of the four researched classifiers.

Table 2: Examples of dataset alterations.

Original	**Alteration**
“...Syrian military’s announcement of a new offensive in Aleppo. “We can’t go out to the world and say we have an agreement when we don’t,” Secretary of State John Kerry said after meeting...”	“...Syrian military’s announcement of a **brand-new** offensive in Aleppo. “We **cannot** go out to the world and **state** we have an agreement when we don’t,” Secretary of State John Kerry **stated** after meeting...”.
“...“It could be Russia, but it could also be China. It could also be lots of other people,” he said during the first presidential debate. “It also could be somebody sitting on their bed that weighs 400 pounds.”...”	“...“It could be Russia, **although** it could **as well** be China. It could **as well** be lots of other people,”” he **stated** during the **1st** presidential debate. “It **as well** could be somebody sitting on their bed that weighs 400 pounds.”...”

Furthermore Tables 5-9 in the appendix show the results from the confusion matrices, including accuracy, recall, precision, and F1.

Looking at the results mentioned above, as well as at the confusion matrices in the appendix we can draw some conclusions. It is clear that, during this research LSTM proved to be susceptible to the semantic-preserving synonym based adversarial attacks made in this research. With a loss in accuracy of -5.81%. The results also suggest Naive Bayes is susceptible to the above mentioned alterations, however since Naive Bayes performed as poorly as it did, it is unlikely the alterations had any effects on the classifier’s “understanding” of the articles’ content, altered or not.

Surprising was how the feed-forward neural network and SVM implementations were not susceptible to the alterations, especially since both classifiers’ core functionality are so different from one another whereas SVM is more closely related to Naive Bayes and LSTM is more closely related to a feed-forward neural network.

Due to the small size of the dataset, however, it is not possible to give any definitive answers as further research is necessary to give definitive answers.

5 Discussion

5.1 Implications

We have to look at the found results and place them in perspective. For this research we talked about the continuous flood of information and the robustness of automatic news classification systems. As mentioned in the introduction; Facebook alone has more than 2.6 billion monthly active users (Clement, 2020). If we assume that every active user posts one story ev-

Table 3: Overview of the results from the experiments.

	Regular test set	**Altered test set**	**Difference**
Naive Bayes	35.71	38.84	(+3.13%)
Support vector machine (SVM)	90.18	90.18	(0%)
Feed-forward neural network	90.63	90.63	(0%)
Long short-term memory (LSTM)	92.86	87.05	(-5.81%)

ery month, depending on their word choice, an LSTM classifier filtering through those posts could, in a worst-case scenario, misclassify 5.81% more posts. This means that same classifier would then, on a monthly basis, misclassify $0.0581 * 2,600,000,000 = 151,060,000$ posts. This number is on top of the 7.14% ($0.0714 * 2,600,000,000 = 185,640,000$) that already are misclassified if this LSTM classifier was in place.

The above mentioned results suggest that the robustness of an SVM or feed-forward neural network classifier outweigh the ≈ 2% performance increase of an LSTM classifier.

5.2 Shortcomings

The results are a first indication that an LSTM news classifier might be not as robust as other options, however we cannot be completely certain an SVM or feed-forward neural network news classifier is insensitive to synonym alterations. This is because the used test set was small ($N = 224$) and the synonym list does not include all possible synonyms and/or the most effective words for an adversarial attack.

The choice was made for semantics-preserving alterations based on synonyms. This was done to ensure the text would, to a human, seem like it could have been officially published, for instance, in a mainstream newspaper. Whereas introducing typos or flipping characters could, to a human, come across as less reliable. This choice, however brought along some difficulties as words are context-dependent, meaning their meaning could change depending on the context. An effort was made to ensure the semantics of each synonym were identical, regardless of context, think of contractions, for example. But it could be some synonyms have different semantics in some contexts, causing small changes in the semantics of the text.

Since the test set was very small and biased towards "mostly true", it is difficult to interpret the results as a classifier that would always predict "true" would still attain an accuracy score of ≈ 90%.

We must keep these shortcomings in mind looking at the results from the research.

5.3 Future research

One of the main aspects that could be further researched is a larger test set as well as more synonyms. An interesting research could be conducted by looking at a list of synonyms and finding highly effective adversarial attack words. These words could then be used in a similar research applied to possibly more different news classifiers or a specific system that has already been implemented to make sure this system is not “foolable”.

6 Conclusion

Taking into account the shortcomings of this research we cannot give a definitive answer to the question: “Can a news misinformation classifier’s scores be altered by altering the style of the input text?” for all researched classifiers.

From the results we can however conclude that an LSTM news classifier is susceptible to semantics-preserving alterations to the inputs. The Naive Bayes implementation also showed differences in scores between the normal and the altered test set. But since the scores of the classifier were as low as they were it is likely to not be due to a change in the “understanding” of the classifier.

Even though the SVM and feed-forward neural network classifiers’ scores showed no differences in scores, it cannot be definitively said that these classifiers are insusceptible to semantics-preserving alterations. To give a definitive answer, a larger test set must be used as well as more synonyms.

This research was aimed at being a steppingstone for further research into this topic. Nowadays terms like “fake news” make people weary of all sorts of information, causing people to distrust media and causing uncertainty as to what is true or false. Automatic news classification systems could be a major step against this distrust, however we must ensure these systems work correctly before having media consumers, and publishers alike, trust them blindly.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2015). *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* Retrieved from http://download.tensorflow.org/paper/whitepaper2015.pdf

Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, *6*, 14410–14430.

Clement, J. (2020, April). *Facebook: active users worldwide.* Retrieved from https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

Curci, J., Khara, K., Pillai, A., & Qin, R. (2018, May). *Fake news detection.* Retrieved from https://github.com/FakeNewsDetection/FakeBuster/blob/master/Project\%20Report.pdf (Accessed: 2020-06-21)

Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018, July). HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 31–36). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P18-2006 doi: 10.18653/v1/P18-2006

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., … Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1625–1634).

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 ieee security and privacy workshops (spw)* (pp. 50–56).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137–142).

Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial examples in the physical world. *ICLR Workshop*. Retrieved from \url{https://arxiv.org/abs/1607.02533}

Le, Q., & Mikolov, T. (2014, 22–24 Jun). Distributed representations of sentences and documents. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (Vol. 32, pp. 1188–1196). Bejing, China: PMLR. Retrieved from http://proceedings.mlr.press/v32/le14.html

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). *Efficient estimation of word representations in vector space.* Retrieved from http://arxiv.org/abs/1301.3781

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2018a, February). *Buzzfeed-webis fake news corpus 2016.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.1239675` doi: 10.5281/zenodo.1239675

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2018b, July). A Stylometric Inquiry into Hyperpartisan and Fake News. In I. Gurevych & Y. Miyao (Eds.), *56th annual meeting of the association for computational linguistics (acl 2018)* (p. 231-240). Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P18-1022`

Samanta, S., & Mehta, S. (2017). Towards crafting text adversarial samples. *CoRR*, *abs/1707.02812*. Retrieved from `http://arxiv.org/abs/1707.02812`

Specht, D. F., et al. (1991). A general regression neural network. *IEEE transactions on neural networks*, *2*(6), 568–576.

Wakefield, A. J., Murch, S. H., Anthony, A., Linnell, J., Casson, D. M., Malik, M., … others (1998). *Retracted: Ileal-lymphoid-nodular hyperplasia, non-specific colitis, and pervasive developmental disorder in children.* Elsevier.

What is wordnet? (n.d.). The Trustees of Princeton University. Retrieved from `https://wordnet.princeton.edu/`

The world's favorite online thesaurus! (n.d.). Retrieved from `https://www.thesaurus.com/`

Zhang, H. (2004). *The optimality of naive bayes.* American Association for Artificial Intelligence.

Zhou, X., & Zafarani, R. (2018). Fake news: A survey of research, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*.

7 Appendix

7.1 Synonym list

Table 4: Overview of the results from the experiments.

Original	**Synonym**
but	although
say	state
like	alike
some	a few
now	immediately
only	merely
said	stated
new	brand-new
president	President of the United States
debate	public debate
also	as well
first	1st
think	consider
voter	elector
told	explained
former	erstwhile
want	desire
aren't	are not
I'm	I am
can't	cannot
I've	I have
didn't	did not
isn't	is not
we're	were
don't	do not
let's	let us
he'll	he will
she'll	she will
you'll	you will
weren't	were not
they'll	they will
couldn't	could not
wouldn't	would not
won't	will not

7.2 Confusion matrices

The rows are the actual classifications and the columns the predicted classifications.

Table 5: Statistics from the Naive Bayes classifier's confusion matrices.

Naive Bayes (regular test set)	True	False	Sum
True	65	137	202
False	7	15	22
Sum	72	152	
Naive Bayes (altered test set)	True	False	Sum
True	71	131	202
False	6	16	22
Sum	77	147	

Table 6: Statistics from the support vector machine (SVM) classifier's confusion matrices.

SVM (regular test set)	True	False	Sum
True	202	0	202
False	22	0	22
Sum	224	0	
SVM (altered test set)	True	False	Sum
True	202	0	202
False	22	0	22
Sum	224	0	

Table 7: Statistics from the feed-forward neural net (FFNN) classifier's confusion matrices.

FFNN (regular test set)	True	False	Sum
True	201	1	202
False	20	2	22
Sum	221	3	
FFNN (altered test set)	True	False	Sum
True	201	1	202
False	20	2	22
Sum	221	3	

Table 8: Statistics from the long short-term memory (LSTM) classifier's confusion matrices.

LSTM (regular test set)	True	False	Sum
True	206	1	207
False	15	2	17
Sum	221	3	
LSTM (altered test set)	True	False	Sum
True	195	0	195
False	29	0	29
Sum	224	0	

7.3 Accuracy, recall, precision, and F1 scores

Table 9: Accuracy (Acc.), Recall (Rec.), Precision (Prec.), and F1 from the classifiers. If the column header contains an (a) then it is about the altered test set. NA means no value could be calculated.

	Acc.	Acc. (a)	Rec.	Rec. (a)	Prec.	Prec. (a)	F1	F1 (a)
NB	35.71%	38.84%	90.28%	92.21%	32.18%	35.15%	47.45%	50.90%
SVM	90.18%	90.18%	90.18%	90.18%	100%	100%	94.84%	94.84%
FFNN	90.63%	90.63%	90.95%	90.95%	99.50%	99.50%	95.04%	95.04%
LSTM	92.86%	87.05%	93.21%	87.05%	99.52%	100%	96.26%	93.08%