Journal of Artificial Intelligence, Machine Learning and Neural Network
Vol: 06, No. 02, July-Dec 2026
ISSN: 2799-1172, DOI: 10.55529/jaimlnn.62.73.83

Research Paper

Numbers keep the spam out: Robustness of word- and character-level SMS spam filters to obfuscation and good-word attacks

Muntaser Syed[1*], Salahddine Krit[2]

[1*]Department of Electrical Engineering and Computer Science, Florida Institute of Technology, Melbourne, Florida, USA.

[2]Full Professor of Computer Science, Department of Informatics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco.

Article Info

Article History:
Received: 03 July 2026
Revised: 01 September 2026
Accepted: 20 September 2026
Published: 22 October 2026

Keywords:
Spam Filtering
Adversarial Machine Learning
Text Obfuscation
Good-Word Attack
Character N-Grams
SMS Robustness

ABSTRACT

Spammers routinely disguise messages to evade statistical filters, by obfuscating words with look-alike characters, digits or inserted punctuation, or by appending words typical of legitimate messages. This study measures how six standard spam filters, namely naive Bayes, logistic regression and linear support vector machines on word or character n-gram features, withstand such evasion on the public SMS Spam Collection of 5,572 real text messages. With five random train–test splits, spam in the test set was attacked with leetspeak, homoglyph substitution, character insertion and typographical errors applied to 10–100% of its words, both randomly and targeted at the most spam-indicative words, and with good-word attacks appending 5–20 words typical of legitimate messages. On clean data the filters reached F1-scores of 95–97% with false-positive rates below 0.7%. Random obfuscation of half of the words reduced spam recall by at most 7.3 percentage points, and some obfuscations even increased the recall of character-level naive Bayes. The main reason for this robustness was not the models but the messages: 78.3% of spam but only 0.1% of legitimate messages contained a number of five or more digits, and such non-lexical cues survived word obfuscation. When the filters were restricted to alphabetic words, clean recall hardly changed (92.6–93.6%), but targeted character insertion of half of the words reduced the recall of character-level models to 10–12%. Good-word attacks were the most effective practical attack, reducing recall to 49–71% with 20 appended words. Text normalisation restored recall under character insertion (from 87.8% to 95.0% for character-level logistic regression), and adversarial training with obfuscated spam raised it to 99.9%, but neither defence helped against good-word attacks, and adversarial training made them more effective. Clean accuracy therefore hides a strong dependence on non-lexical cues, and robustness evaluations of spam filters should include both targeted obfuscation and good-word attacks.

Check for updates

Corresponding Author:
Muntaser Syed
Department of Electrical Engineering and Computer Science, Florida Institute of Technology, Melbourne, Florida, USA.
Email: msyed2011@my.fit.edu

Copyright © 2026 The Author(s). This is an open access article distributed under the Creative Commons Attribution License, (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Statistical text classifiers have filtered unwanted e-mail and text messages for more than two decades [1], [2]. Short text messages (SMS) pose particular difficulties because they are brief, informal and full of abbreviations [3], [4], yet simple bag-of-words and character n-gram models with naive Bayes, logistic regression or support vector machines [5], [6] remain strong and widely deployed baselines. Spam filtering is also an adversarial problem: spammers adapt their messages to evade filters [7], [8]. Two families of evasion have been described since the early days of anti-spam research. In obfuscation attacks, spammers alter the spelling of suspicious words, for example writing "FR33", "f.r.e.e" or replacing Latin letters with visually identical Cyrillic ones; in good-word attacks, they append words that are typical of legitimate messages to shift the classifier's decision [9], [10].

Adversarial robustness of text classifiers has since become a major research topic, with character-level perturbations [11], [12], [13], [14], visually similar characters [15], [16] and word substitutions [17] shown to fool modern models, and defences such as robust word recognition [18] and adversarial training [19], [20] proposed. Most of this work concerns neural models and generic text; how much the classic, still widely used spam filters actually lose under realistic obfuscation of short messages, and why, is less well documented. The distinction between word-level and character-level features is especially relevant, because character n-grams are often recommended for their tolerance of spelling variation [6].

This paper evaluates six standard SMS spam filters under five evasion attacks and two defences, using a public collection of real text messages. The contributions are as follows:

A systematic evaluation of word- and character-level naive Bayes, logistic regression and linear SVM filters under leetspeak, homoglyph, character-insertion, typo and good-word attacks at several strengths, with five random splits.

Evidence that SMS spam filters are surprisingly robust to random word obfuscation, together with an explanation: most spam messages contain non-lexical cues, such as long numbers and currency symbols, that survive word obfuscation.

An ablation showing that when these cues are removed, clean accuracy is almost unchanged but targeted obfuscation becomes devastating, reducing the recall of character-level models to about 10%.

A comparison of text normalisation and adversarial training as defences, showing that both counter obfuscation but neither counters good-word attacks, which adversarial training even amplifies.

2. RELATED WORK

2.1 Statistical Spam Filtering

Cormack [1] and Guzella and Caminhas [2] review statistical approaches to e-mail spam filtering. Linear classifiers on word features, in particular naive Bayes, logistic regression and support vector machines [5], dominate practical systems because they are fast, easy to update and accurate. [6] compared words and character n-grams for spam filtering and found character n-grams competitive or better and less

sensitive to deliberate misspellings. For SMS, [3] showed that short messages require adapted features, and [4] released the SMS Spam Collection used here and reported that linear SVMs and naive Bayes achieve high accuracy.

2.2 Adversarial Attacks on Text Classifiers

[7] Formulated classification as a game between classifier and adversary, and Lowd and Meek [8] studied how an adversary can find low-cost evasions of a linear classifier. In a companion paper, Lowd and Meek [9] showed that appending words indicative of legitimate e-mail can let a large share of spam past naive Bayes and maximum-entropy filters, [10] proposed multiple-instance learning as a countermeasure. [21] and Biggio and Roli [20] review the security of machine learning more broadly. For neural text models, DeepWordBug [11], HotFlip [12], TextBugger [14] and TextFooler [17] generate character- or word-level perturbations, Belinkov and Bisk [13] showed that natural and synthetic noise break neural machine translation, [15] and [16] demonstrated attacks with visually similar or invisible Unicode characters, which motivate the Unicode confusable-character mechanisms [22].

2.3 Defences

Defences fall into two broad groups. Pre-processing defences normalise the input before classification, for example by mapping confusable characters to a canonical form [22] or recognising misspelled words [18]. Training-time defences, notably adversarial training [19], [23], add perturbed examples to the training data so that the model learns to classify them correctly. The relative merits of the two for simple spam filters, and their effect on attacks other than those used in training, are examined in this study.

3. METHODOLOGY

Figure 1 summarises the threat model. The attacker modifies spam messages before they reach the filter, without access to the filter's parameters [9]; the filter is trained on unmodified historical data unless a defence is applied.

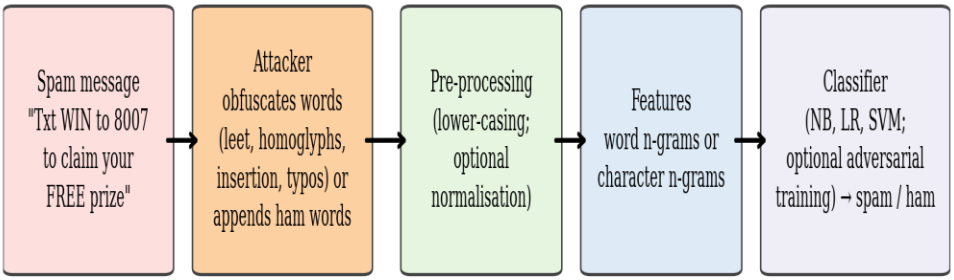


Figure 1. Threat Model and Evaluation Pipeline. The Attacker Modifies Spam Messages; the Filter Pre-Processes the Text, Extracts Word or Character N-Gram Features and Classifies Each Message. Optional Defences Act at the Pre-Processing (Normalisation) or Training (Adversarial Training) Stage

3.1 Data

The SMS Spam Collection [4] contains 5,572 real English text messages, of which 747 (13.4%) are labelled spam and 4,825 legitimate (ham). Spam messages are on average about twice as long as ham messages (139 versus 72 characters) and contain on average 17.2 alphabetic words of three or more letters. Each experiment was repeated with five stratified random splits into 70% training and 30% test data (about 1,672 test messages, including about 224 spam messages, per split).

3.2 Filters

Six filters were built with scikit-learn [24]. Word-level filters used unigrams and bigrams, and character-level filters used character 2- to 5-grams within word boundaries. Multinomial naive Bayes (NB, smoothing 0.1) used raw counts; logistic regression [25] (LR, C = 10) and a linear support vector machine [5], [26] (SVM, C = 1) used sub-linearly scaled TF-IDF weights with class weights balanced for the spam minority. All text was lower-cased.

3.3 Attacks

Five attacks were applied to every spam message in the test set *Table 1*. Four obfuscation attacks modify a fraction of the alphabetic words of three or more letters: leetspeak replaces letters with look-alike digits (a→4, e→3, i→1, o→0, s→5, t→7); homoglyphs replace lower-case Latin letters with visually identical Cyrillic letters (a, e, o, c, p, x, y, i); character insertion inserts a full stop between all letters of a word; and typos swap two adjacent letters or delete one. Obfuscation rates of 10%, 25% and 50% of words were used in the main experiment, with words chosen at random. The good-word attack [9] appends 5, 10 or 20 words chosen at random from a list of 100 words that are frequent in legitimate training messages and at least five times more frequent in ham than in spam, such as “sorry”, “home”, “later” and “happy”; it models an attacker who has access to ordinary text messages but not to the filter.

Table 1. Attacks Applied To a Real Spam Message from the Test Set (Obfuscation Rate 25%; 10 Appended Words for the Good-Word Attack)

Attack	Resulting Message
Original	1000’s flirting NOW! Txt GIRL or BLOKE & ur NAME & AGE, eg GIRL ZOE 18 to 8007 to join and get chatting!
Leetspeak	1000’s flirting NOW! Txt G1RL or BLOKE & ur NAME & AGE, eg GIRL Z03 18 to 8007 to j01n and get chatting!
Homoglyphs	1000’s flirting NOW! Txt GIRL or BLOKE & ur NAME & AGE, eg GIRL ZOE 18 to 8007 to join and get chatting! (Cyrillic o and i in “join”)
Character insertion	1000’s flirting NOW! Txt G.I.R.L or BLOKE & ur NAME & AGE, eg GIRL Z.O.E 18 to 8007 to j.o.i.n and get chatting!
Typos	1000’s flirting NOW! Txt IGRL or BLOKE & ur NAME & AGE, eg GIRL ZE 18 to 8007 to oin and get chatting!
Good words	… To join and get chatting! pick made wont happy thk sweet yup office thanx sorry

Homoglyph substitution applies only to lower-case letters, so upper-case words such as “GIRL” remain unchanged.

3.4 Defences

Two defences were applied to the word- and character-level logistic-regression filters. Normalisation pre-processes every message, in training and at test time, with Unicode NFKC normalisation, mapping of the Cyrillic homoglyphs to Latin letters [22], removal of punctuation between letters, and mapping of leetspeak digits back to letters in tokens that also contain letters (tokens with runs of three or more digits, such as phone numbers, are left untouched). Adversarial training adds to the training data one obfuscated copy of every training spam message, produced with a randomly chosen obfuscation attack at a 25% rate. The good-word attack was not used in adversarial training, so it is an unseen attack for this defence.

3.5 Extended Experiments

Two additional factors were examined with the word-level NB and LR filters and the character-level LR and SVM filters. First, targeted obfuscation perturbs the words with the highest spam-to-ham log-frequency ratio in the training data, modelling an attacker who knows which words are typical of spam, and was compared with random obfuscation at rates of 25%, 50% and 100%. Second, a letters-only ablation trained and evaluated the filters on text reduced to alphabetic tokens of three or more characters,

removing digits, currency symbols and short tokens, in order to test whether non-lexical cues explain the filters' robustness.

3.6 Evaluation

Filters were evaluated on clean test data by spam recall, false-positive rate (the proportion of ham classified as spam) and F1-score, and under attack by spam recall on the attacked test spam, since the attacks modify only spam and leave the false-positive rate unchanged. All results are means over five splits with standard deviations; seed-to-seed standard deviations of attacked recall in the main experiment were at most 3.7 percentage points.

4. RESULTS AND DISCUSSION

4.1 Clean Performance

Table 2 shows that all six filters performed well on clean data, with spam recall of 93.8–95.2%, false-positive rates of 0.08–0.59% and F1-scores of 95.1–97.0%. Character-level logistic regression and SVM had the highest F1-scores (96.9–97.0%) and the lowest false-positive rates (0.08–0.17%). Neither defence changed clean performance materially: the largest change in F1-score was −0.4 percentage points (normalised word-level LR), and false-positive rates remained below 0.7%.

Table 2. Clean Test Performance (%, Mean ± SD over Five Splits)

Filter	Defence	Spam Recall	False-Positive Rate	F1-Score
NB (word)	None	94.1 ± 1.9	0.47 ± 0.06	95.5 ± 1.0
LR (word)	None	93.8 ± 1.7	0.54 ± 0.13	95.1 ± 1.2
SVM (word)	None	93.8 ± 1.8	0.41 ± 0.16	95.5 ± 1.4
NB (char)	None	95.2 ± 1.4	0.59 ± 0.08	95.6 ± 0.8
LR (char)	None	95.1 ± 1.5	0.17 ± 0.08	97.0 ± 0.9
SVM (char)	None	94.5 ± 1.3	0.08 ± 0.08	96.9 ± 0.9
LR (word)	Normalisation	93.7 ± 1.3	0.65 ± 0.13	94.7 ± 1.0
LR (char)	Normalisation	95.0 ± 1.3	0.18 ± 0.09	96.9 ± 0.8
LR (word)	Adversarial training	94.0 ± 1.3	0.66 ± 0.18	94.8 ± 0.9
LR (char)	Adversarial training	94.6 ± 1.2	0.10 ± 0.12	96.9 ± 0.9

4.2 Random Obfuscation has Little Effect

Figure 2 and *Table 3* show spam recall under attack. Random obfuscation had only a small effect. With half of the words obfuscated, recall fell by at most 7.3 percentage points (character-level LR under character insertion, from 95.1% to 87.8%), and in most cases by 1–4 points. Word-level filters lost 2–5 points under all four obfuscations. Character-level filters were more robust to leetspeak, homoglyphs and typos than word-level filters, as expected from their tolerance of partial spelling changes [6], but more sensitive to character insertion, which breaks every character n-gram inside a word. Leetspeak even increased the recall of character-level NB, from 95.2% to 98.2% at a 50% rate: digit-letter combinations such as "z03" are themselves characteristic of spam, so the obfuscation added evidence rather than hiding it.

4.3 Good-Word Attacks Are the Most Effective

Appending words typical of legitimate messages was far more effective than obfuscation *Table 3*. With 20 appended words, recall fell to 49.2–71.4%, so that roughly a third to a half of the attacked spam passed the filters. Naive Bayes filters were the most vulnerable (55.1% for words and 49.2% for characters), consistent with the original findings of Lowd and Meek [9], because each appended word contributes an independent log-likelihood term. Discriminatively trained LR and SVM filters were more resistant, and the character-level versions most of all (71.0–71.4%), but none was robust.

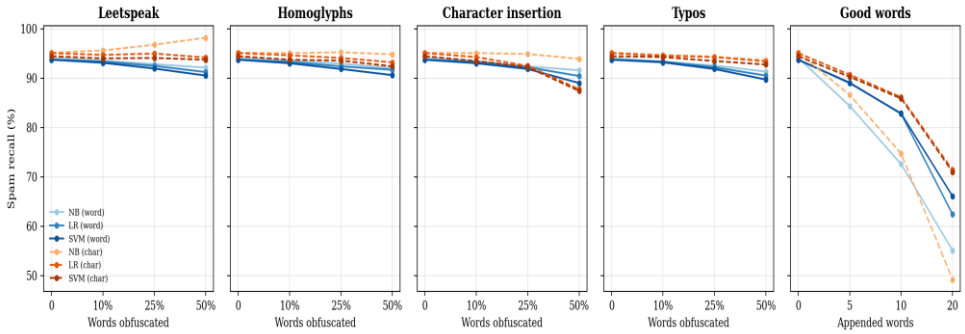


Figure 2. Spam Recall of the Six Filters under Each Attack as a Function of Attack Strength (Mean Over Five Splits). Solid Lines Denote Word-Level and Dashed Lines Character-Level Filters

Table 3. Spam Recall (%) Under The Strongest Random Obfuscation and Under Good-Word Attacks (Mean Over Five Splits)

Attack (Strength)	NB (Word)	LR (Word)	SVM (Word)	NB (Char)	LR (Char)	SVM (Char)
None (clean)	94.1	93.8	93.8	95.2	95.1	94.5
Leetspeak (50%)	92.1	91.2	90.5	98.2	94.2	93.8
Homoglyphs (50%)	92.1	91.7	90.6	94.8	93.2	92.4
Character insertion (50%)	91.6	90.4	89.0	93.9	87.8	87.4
Typos (50%)	91.3	90.5	89.7	93.7	93.4	92.8
Good words (5)	84.4	89.1	89.0	86.6	90.7	90.3
Good words (10)	72.6	82.8	82.9	74.7	86.2	85.9
Good words (20)	55.1	62.4	66.1	49.2	71.4	71.0

4.4 Why Obfuscation Fails: Non-Lexical Cues

The robustness to obfuscation is explained by the content of SMS spam rather than by the filters. *Figure 3* shows the prevalence of several cues in the full collection. A number with five or more digits, typically a premium-rate phone number or short code, occurred in 78.3% of spam but only 0.1% of ham messages; any digit occurred in 94.8% of spam versus 15.6% of ham; and currency symbols occurred in 34.4% versus 0.4%. These cues are not alphabetic words and are therefore untouched by word obfuscation, and they remain available to both word-level and character-level filters. Among the 25 features with the largest positive weights in a word-level logistic-regression model trained on the full collection, several were such tokens, for example “150p”, “50” and “18”, alongside words such as “txt”, “call”, “free” and “claim”.

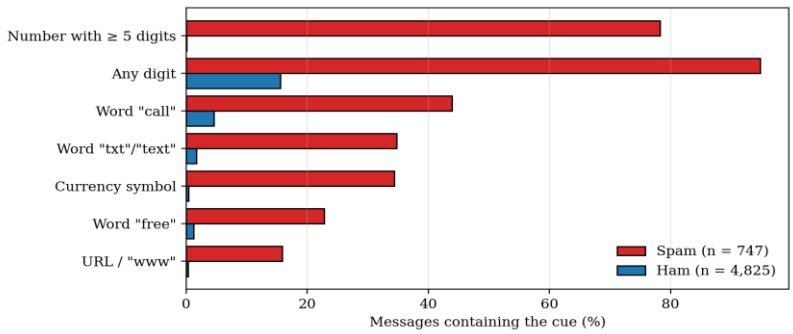


Figure 3. Prevalence of Lexical and Non-Lexical Cues in Spam and Legitimate (Ham) Messages of the SMS Spam Collection. Long Numbers, Digits and Currency Symbols are Almost Exclusive to Spam and are not affected by Word Obfuscation

The letters-only ablation *Table 4* confirms this explanation. Restricting the filters to alphabetic tokens hardly changed their clean recall (92.6–93.6% versus 93.8–95.1% on full text), so a conventional evaluation would not reveal the difference. Under obfuscation, however, the letters-only filters were much more vulnerable. At a 50% random obfuscation rate, character insertion reduced the recall of character-level LR from 88.2% on full text to 60.7% on letters only, and when all words were obfuscated the letters-only character-level filters detected none of the spam under character insertion (0.0%) and only 11.5–11.8% under leetspeak, compared with 48.1–50.3% and 90.7–92.2% on full text.

4.5 Targeted Obfuscation

An attacker who obfuscates the most spam-indicative words, rather than random words, is far more effective *Table 4* and *Figure 4*. On full text, targeting half of the words reduced the recall of word-level LR to 72.8–82.5% depending on the attack, compared with 89.2–91.0% for random obfuscation of the same number of words. Character-level LR remained above 88% under targeted leetspeak, homoglyphs and typos but fell to 73.6% under targeted character insertion. On letters-only text, targeted obfuscation of half of the words was devastating: targeted character insertion reduced the recall of character-level LR and SVM to 12.1% and 10.3%, and targeted leetspeak reduced it to 45.6% and 42.5%. Non-lexical cues therefore act as a safety net that hides the lexical vulnerability of all six filters.

Table 4. Spam Recall (%) Under Random and Targeted Obfuscation Of 50% of Words, On Full Text and on Letters-Only Text (Mean Over Five Splits of the Extended Experiment)

Attack (50% of Words)	LR (Word), Full, Random	LR (Word), Full, Targeted	LR (Word), Letters Only, Targeted	LR (Char), Full, Random	LR (Char), Full, Targeted	LR (Char), Letters Only, Targeted	SVM (Char), Letters Only, Targeted
None (clean recall)	93.8	93.8	93.4	95.1	95.1	93.6	92.6
Leetspeak	90.3	75.7	53.8	94.7	89.2	45.6	42.5
Homoglyphs	91.0	82.5	69.5	93.0	88.2	65.0	61.8
Character insertion	89.6	72.8	50.4	88.2	73.6	12.1	10.3
Typos	89.2	74.8	56.8	93.2	89.5	71.2	68.6

Random obfuscation in the extended experiment used different random draws from the main experiment, so full-text random values differ slightly from *Table 3*.

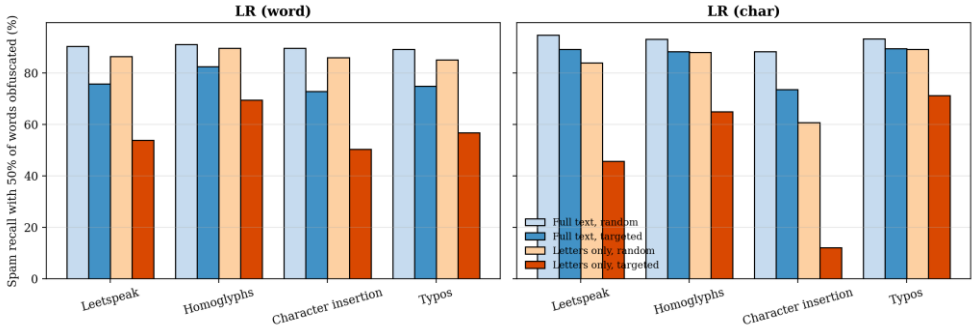


Figure 4. Spam Recall With 50% of Words Obfuscated, For Random and Targeted Obfuscation on Full Text and on Letters-Only Text, For Word-Level (Left) and Character-Level (Right) Logistic Regression

4.6 Defences

Figure 5 and *Table 5* compare the two defences at the maximum attack strength. Normalisation undid the obfuscations it was designed for: under character insertion, recall of character-level LR rose from 87.8% to 95.0% and that of word-level LR from 90.4% to 93.7%, essentially the clean level, and recall under

homoglyphs and leetspeak also returned to clean values. Typos, which normalisation does not address, were unaffected. Adversarial training was even more effective against the obfuscations it had seen, raising the recall of character-level LR under character insertion to 99.9% and under leetspeak to 97.7%, above its clean recall; the model had learned that obfuscation itself indicates spam. Neither defence helped against the good-word attack. Adversarial training made it more effective, lowering recall with 20 appended words from 71.4% to 60.1% for character-level LR and from 62.4% to 53.9% for word-level LR, and normalisation also lowered the recall of word-level LR under this attack (to 55.7%).

Table 5. Spam Recall (%) Of Logistic-Regression Filters with and Without Defences at the Maximum Attack Strength (50% of Words Obfuscated; 20 Appended Words)

Attack	LR (Word), no Defence	LR (Word), Normalisation	LR (Word), Adversarial Training	LR (Char), no Defence	LR (Char), Normalisation	LR (Char), Adversarial Training
Leetspeak	91.2	93.4	93.3	94.2	95.1	97.7
Homoglyphs	91.7	93.7	92.8	93.2	95.0	95.5
Character insertion	90.4	93.7	92.0	87.8	95.0	99.9
Typos	90.5	89.7	92.5	93.4	93.3	94.0
Good words (20)	62.4	55.7	53.9	71.4	71.2	60.1

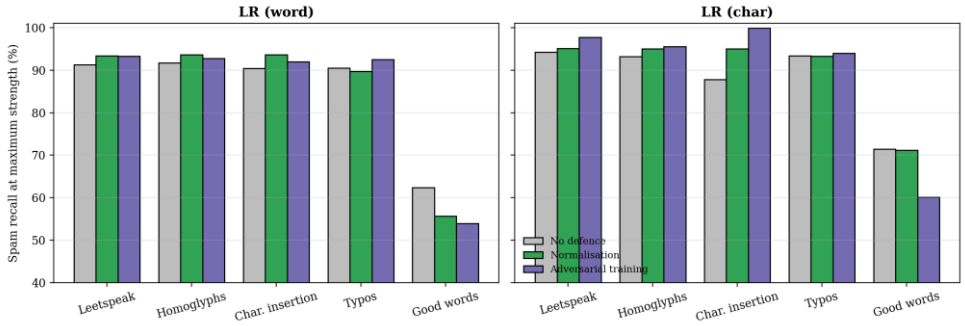


Figure 5. Spam Recall at the Maximum Attack Strength for Word-Level (Left) and Character-Level (Right) Logistic Regression without Defence, With Text Normalisation and With Adversarial Training

4.7 Discussion

4.7.1 Principal Findings

On a public collection of real text messages, six standard spam filters proved surprisingly robust to random word obfuscation, losing at most 7.3 percentage points of recall with half of the words obfuscated. The robustness, however, stemmed largely from non-lexical cues, above all long phone numbers and short codes, which appear in most spam and almost no legitimate messages. When those cues were removed, clean accuracy barely changed, yet targeted obfuscation reduced detection to about 10%. The most effective attack that leaves these cues intact was the good-word attack, which let a third to a half of spam through with 20 appended words and was not countered by either defence. These results agree with earlier findings on good-word attacks [9], [10] and on the tolerance of character n-grams to misspellings [6], and they add a clear demonstration that clean accuracy can hide which cues a filter depends on.

4.7.2 Implications

For practitioners, four points follow. First, a filter's robustness depends on the data as much as on the model; in SMS, spammers must include a contact number or short code for their scheme to work, which gives filters a cue that is costly to remove. Spam in other channels, such as e-mail with links hidden behind text or messaging platforms without phone numbers, may lack such cues, and the letters-only results suggest how vulnerable filters would then be. Second, simple normalisation of confusable characters,

punctuation and digit substitutions [22] is cheap, does not harm clean performance and neutralises the obfuscations it anticipates, so it should be standard. Third, adversarial training against obfuscation works well for the attacks it has seen but can increase vulnerability to different attacks, here the good-word attack; defences should therefore be evaluated against attacks outside their training set [20]. Fourth, good-word attacks remain the main threat to bag-of-words filters and require dedicated countermeasures, such as segment-based or multiple-instance classification [10], or features that are harder to dilute.

4.7.3 Limitations

The study used a single public SMS collection in English, which is relatively small and dates from the early 2010s; spam in other languages, channels and years may differ. The attacks were synthetic and passive, without feedback from the filter; adaptive attackers who query the filter [8] would be more effective. Homoglyph substitution was applied only to lower-case letters, which underestimates its potential. The normalisation defence mapped leetspeak digits back to letters in mixed tokens, which also altered some legitimate tokens such as "1st"; this affected training and test data equally. Only linear filters with n-gram features were studied; neural and transformer-based filters may behave differently, although they are known to be vulnerable to similar perturbations [11], [16], [17]. Finally, the targeted attack used word statistics from the training data, which represents a knowledgeable but not omniscient attacker.

5. CONCLUSION

Six word- and character-level SMS spam filters were evaluated under five evasion attacks and two defences on 5,572 real text messages. Random obfuscation of up to half of the words reduced spam recall by no more than 7.3 percentage points, because most spam carries non-lexical cues, such as long phone numbers, that obfuscation does not touch; without these cues, clean accuracy was almost unchanged but targeted obfuscation reduced detection to about 10%. Good-word attacks were the most effective, letting up to half of the spam through, and were not stopped by normalisation or adversarial training, the latter of which made them more effective. Clean accuracy therefore reveals little about a filter's robustness, and evaluations should include targeted obfuscation, good-word attacks and ablations of the cues a filter relies on. Future work will examine neural filters, adaptive attackers and other languages and messaging channels.

Acknowledgments

The authors thank the creators of the SMS Spam Collection for making it publicly available, the developers of scikit-learn, and the researchers whose work is cited in this article.

Funding Information

This research has received no external funding.

Author Contribution Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Muntaser Syed	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Salahddine Krit		✓	✓		✓	✓	✓	✓		✓	✓		✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

Conflict of Interest Statement

No conflict of interest.

Informed Consent

Not Applicable.

Ethical Approval

Not Applicable. The study used a publicly available, anonymised research dataset.

Data Availability

This study draws on the publicly available SMS Spam Collection. No new datasets were created. The analysis code is available from the corresponding author on reasonable request.

REFERENCES

[1] G. V. Cormack, "Email spam filtering: A systematic review," Foundations and Trends in Information Retrieval, vol. 1, no. 4, pp. 335–455, 2008. doi.org/10.1561/1500000006

[2] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," Expert Systems with Applications, vol. 36, no. 7, pp. 10206–10222, 2009. doi.org/10.1016/j.eswa.2009.02.037

[3] G. V. Cormack, J. M. Gómez Hidalgo, and E. Puertas Sánz, "Spam filtering for short messages," in Proc. 16th ACM Conf. Information and Knowledge Management, 2007, pp. 313–320. doi.org/10.1145/1321440.1321486

[4] T. A. Almeida, J. M. Gómez Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in Proc. 11th ACM Symp. Document Engineering, 2011, pp. 259–262. doi.org/10.1145/2034691.2034742

[5] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Machine Learning: ECML-98, Lecture Notes in Computer Science, vol. 1398. Berlin: Springer, 1998, pp. 137–142. doi.org/10.1007/BFb0026683

[6] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words versus character n-grams for anti-spam filtering," International Journal on Artificial Intelligence Tools, vol. 16, no. 6, pp. 1047–1067, 2007. doi.org/10.1142/S0218213007003692

[7] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2004, pp. 99–108. doi.org/10.1145/1014052.1014066

[8] D. Lowd and C. Meek, "Adversarial learning," in Proc. 11th ACM SIGKDD Int. Conf. Knowledge Discovery in Data Mining, 2005, pp. 641–647. doi.org/10.1145/1081870.1081950

[9] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in Proc. 2nd Conf. Email and Anti-Spam (CEAS), 2005. personal.utdallas.edu/~mxk055100/courses/dmsec_files/125.pdf

[10] Z. Jorgensen, Y. Zhou, and M. Inge, "A multiple instance learning strategy for combating good word attacks on spam filters," Journal of Machine Learning Research, vol. 9, pp. 1115–1146, 2008. jmlr.org/papers/v9/jorgensen08a.html

[11] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in Proc. IEEE Security and Privacy Workshops, 2018, pp. 50–56. doi.org/10.1109/SPW.2018.00016

[12] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: White-box adversarial examples for text classification," in Proc. 56th Annu. Meeting of the Association for Computational Linguistics (Short Papers), 2018, pp. 31–36. doi.org/10.18653/v1/P18-2006

[13] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in Proc. 6th Int. Conf. Learning Representations, 2018. doi.org/10.48550/arXiv.1711.02173

[14] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," in Proc. Network and Distributed System Security Symp., 2019. doi.org/10.14722/ndss.2019.23138

[15] S. Eger et al., "Text processing like humans do: Visually attacking and shielding NLP systems," in Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics, 2019, pp. 1634–1647. doi.org/10.18653/v1/N19-1165
[16] N. Boucher, I. Shumailov, R. Anderson, and N. Papernot, "Bad characters: Imperceptible NLP attacks," in Proc. IEEE Symp. Security and Privacy, 2022, pp. 1987–2004. doi.org/10.1109/SP46214.2022.9833641
[17] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," in Proc. 34th AAAI Conf. Artificial Intelligence, 2020, pp. 8018–8025. doi.org/10.1609/aaai.v34i05.6311
[18] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," in Proc. 57th Annu. Meeting of the Association for Computational Linguistics, 2019, pp. 5582–5591. doi.org/10.18653/v1/P19-1561
[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in Proc. 3rd Int. Conf. Learning Representations, 2015. doi.org/10.48550/arXiv.1412.6572
[20] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," Pattern Recognition, vol. 84, pp. 317–331, 2018. doi.org/10.1016/j.patcog.2018.07.023
[21] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," Machine Learning, vol. 81, no. 2, pp. 121–148, 2010. doi.org/10.1007/s10994-010-5188-5
[22] Unicode Consortium, "Unicode Technical Standard #39: Unicode Security Mechanisms." www.unicode.org/reports/tr39/
[23] C. Szegedy et al., "Intriguing properties of neural networks," in Proc. 2nd Int. Conf. Learning Representations, 2014. doi.org/10.48550/arXiv.1312.6199
[24] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. jmlr.org/papers/v12/pedregosa11a.html
[25] D. R. Cox, "The regression analysis of binary sequences," Journal of the Royal Statistical Society: Series B (Methodological), vol. 20, no. 2, pp. 215–242, 1958. doi.org/10.1111/j.2517-6161.1958.tb00292.x
[26] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995. doi.org/10.1007/BF00994018

How to Cite: Muntaser Syed, Salahddine Krit (2026). Numbers keep the spam out: Robustness of word- and character-level SMS spam filters to obfuscation and good-word attacks. Journal of Artificial Intelligence, Machine Learning and Neural Network, 6(2), 73-83. https://doi.org/10.55529/jaimlnn.62.73.83

BIOGRAPHIES OF AUTHORS



Muntaser Syed is a researcher in the Department of Electrical Engineering and Computer Science at the Florida Institute of Technology, Melbourne, Florida, USA. His work covers deep learning, transfer learning and natural language processing, including topic modelling and text classification. He is interested in how learning systems behave under imperfect supervision and adversarial input, and in the empirical study of their failure modes.



Prof. Salahddine Krit is a Full Professor of Computer Science in the Department of Informatics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco. His research and teaching centre on computer and network security, embedded and distributed systems, and the engineering of secure communications. He has a particular interest in how security and robustness considerations shape the design of practical computing systems, and in the application of machine learning to filtering and detection problems where adversarial behaviour is expected.