# All Y0ur Da1a @re 3ncryp1ed: Identifying the Language of Ransomware Notes

**Samuel Miller, Judith Klavans**
University of Maryland, College Park
samm@umd.edu, jklavans@umd.edu

**Abstract**

Research on malicious activity detection has focused on the code in the malicious files themselves, but the human language text files that come with certain types of malware (e.g. ransomware) are often overlooked. We explore and address this gap in malicious activity research. The contribution of this paper is demonstrating the viability of using neural sequence classifiers to distinguish ransomware notes from other files. We show strong performance by tuning on English language ransomware note text and various types of benign files, demonstrating accurate identification of ransomware note text. We present an improvement over other methods of malicious text classification, with an F1-score of 96.8% on content excerpts, compared to the highest F1-score of 92.0% on filenames in prior work (Lemmou et al., 2021). We also revealed some unexpected insights into the characteristics of ransomware notes that, to our knowledge, is novel in the published literature. Finally, we have made publicly available the dataset we curated, for research purposes.

**Keywords:** Corpus Creation, Text Categorisation, Applications, Ransomware, Neural Classifiers, Malware Detection, Cybersecurity

## 1. Introduction

The ability to identify files on a system that indicate malicious activity is an important technique to maintain system security. Ransomware threats are on the rise, and constitute a serious and damaging threat to global balance. In their 2022 mid-year report, Acronis estimated that by 2023, ransomware damages were expected to exceed US $30 billion, with devastating impacts on cities, organizations, and individuals (Acronis, 2022). The global average cost of a single data breach in 2023 according to IBM's Cost of a Data Breach Report 2023 was US $4.45 million (IBM, 2023). According to Sophos' The State of Ransomware 2023 report , the mean cost of a single ransomware attack in 2023 was US $1.54 million, plus an additional US $1.82 million for each data recovery effort (Sophos). SpyCloud's 2023 Ransomware Defense Report found that 81% of organizations in their study were affected by ransomware in some capacity over a 12 month period, and 48% of those that were impacted ended up paying a ransom (SpyCloud, 2023). In addition to the fiscal impacts, hospitals experience increased mortality rates following attacks (Cynerio and Ponemon Institute, 2022), and there are even cases of victims committing suicide following personal attacks (Sawler, 2016).

In the context of the grave implications of ransomware attacks, this paper contributes to a new direction of risk detection and mitigation: training a neural classifier to identify the ransomware notes used to inform users of the attack and ransom payment instructions.

Natural Language Processing techniques offer many technologies and methods to the study of malicious activity detection. Neural classifiers have been used to great success for a variety of text classification tasks (Devlin et al., 2019; Sun et al., 2020). Expanding these technologies to ransomware note classification may add another useful tool for malware prevention: if these files are detected when scanning a download, the ransomware can be isolated and prevented from taking root. It is even more so a useful tool in malware detection: the presence of a ransomware note file on a system is a strong indicator that the system is compromised. Therefore, detection of ransomware note files is a highly relevant task for cybersecurity interests. However, this task is made difficult due to a low quantity of data available and a lack of large public ransomware note datasets.

The contribution of this paper is two-fold. We demonstrate the viability of using neural text classifiers to distinguish ransomware notes from other files. This is an important step forward in the identification of digital human language content indicative of malicious activity. In addition, we curate and make available a cleaned dataset for future research.

## 2. Related Work

In the past, within the field of cybersecurity, the use of Natural Language Processing technologies had been underexplored (Klavans, 2015). However, there is now research that uses NLP techniques for cybersecurity applications, for example, training neural networks to identify malicious internet traffic

(Shenfield et al., 2018; Gao et al., 2020) and malicious URLs (Yuan et al., 2021; Chen et al., 2021). There is also research on a similar task to ours: spam message detection (Jain et al., 2018; Sheikhi et al., 2020). However, there does not seem to have been much research directly on the task of malicious text classification. A related paper on identifying malicious text in emails and public comments does not use ransomware notes (Baccouche et al., 2020). Another related paper uses pictures of ransomware applications (Atapour-Abarghouei et al., 2019).

In terms of ransomware note analysis itself, there is one published research that investigates the content of ransomware note files (Lemmou et al., 2021). In that paper, the names and contents of such files were analyzed. They used the data they accrued to train various machine learning models to classify files as malicious and benign, however only by filename. They also use Latent Semantic Analysis to compare similarities in text across ransomware note content and benign files.

The only other prior work to our knowledge that attempted ransomware note classification based on content is from a presentation at DEFCON (Mager, 2018). While they do show comparable (albeit slightly worse) performance to what we ultimately achieve, their benign data is limited to only the 20 Newsgroups dataset and certain Windows system logs and READme files. This article also does not give full details on how the classifier was implemented (only that it is a Naïve Bayes model).

These works demonstrate that we are not the first to conceptualize the idea of using ransomware note text classification as a proxy for malicious activity detection. However, our work is the first to utilize a transformer-based approach to our knowledge. In addition, we synthesize multiple sources of ransomware note and benign file data to create a new dataset.

## 3. Dataset Creation

### 3.1. Collecting the Data

Acquiring and aggregating data was necessary to assess if neural text classifiers can be trained to identify ransomware notes and other files that indicate malicious activity. Ensuring that the data used is clean, relevant, and robust was likely the most crucial step in this project.

#### 3.1.1. Ransomware Notes

The ransomware note data collected came from two main sources. The first was from the research by Lemmou et al. 2021 (Lemmou et al., 2021). Currently, their archive[1] contains 182 ransom note files (HTML or txt) from 71 distinct ransomware families. The other source of data is from RansomWhere, a crowdsourced ransomware payment tracker, where anyone can submit the ransomware they were infected with and how much bitcoin they were asked to pay (Cable, 2022). The site collects proof that a ransomware event actually occurred when someone supplies data to them, and one of the methods of proof is attaching the ransomware note. The authors of this repository have generously shared the notes collected for the use of this paper. From their data, we were able to filter out 20 additional unique files.

> Attention! All your files, documents, photos, databases and other important files are encrypted! The only method of recovering files is to purchase an unique private key. Only WE can recover your files! You can get the private key by email as well as through a closed TOR network. You can get there by the following ways Main email : argusdecrypt@cock.li Reserved email : argusdecrypt@mailfence.com Follow the instructions If the answer for a long time no Download Tor browser https: www.torproject.org download downloadeasy Install Tor browser Open Tor Browser Open link in TOR browser: http: argusqug6aw25gye.onion Follow the instructions on this page You can get the opportunity to decrypt 1 file for free. ATTENTION! IN ORDER TO PREVENT DATA DAMAGE: DO NOT MODIFY ENCRYPTED FILES

Figure 1: An excerpt from a ransomware note file

After carefully selecting the ransomware note files that we would use (in English and generally clean), the collection consisted of 180 distinct files. See Figure 1 for an excerpt from one of the files.

#### 3.1.2. Benign Files

The process of selecting a comparable data set presented some challenges. We considered collecting spam emails asking for money, passwords or the like. However, the aim of this research is specifically to identify ransomware notes. Spam is generally considered malicious, and this type of message has been well-studied and documented (Jain et al., 2018; Sheikhi et al., 2020). On the other hand, ransomware text has been little studied and thus is the primary focus of this research. Therefore, we focused on collecting other text data which might exist on a device potentially infected with malware. These data were surprisingly difficult to find, given copyright and ethical issues, e.g. data

[1] https://github.com/lemmou/RansomNoteFiles

containing personally identifiable information.

In the data we collected, we had to manually extract the relevant data from the entire file. Data was sampled from 11 datasets to create a somewhat comparable dataset to test the ability of our methods to discriminate ransomware notes from other files. An excerpt from each data source is shown in Table 1.

**Personal Notes** The first data we collected were personal notes: any type of file that the user of a system might write for themselves. We drew upon existing datasets of 10,101 to-do lists (Jauhar et al., 2021) and 1,473 journal/diary entries (Li and Parikh, 2019) to represent this category.

**Informative Files** The second data we collected were informative files that might be downloaded. We represented this category by taking from datasets of around 2,300 bitcoin news articles (Baskar, 2022) and around 57,000 Instructables DIY articles (Larsen, 2020) to represent this. The use of these sources had the added benefit of using vocabulary and language which is similar to that of ransomware notes. In particular, the DIY articles featured step-by-step instructions, common also to ransomware notes, where attackers provide specific steps for those attacked to recover files.

**Human Speech for Entertainment** The third data we collected were files containing general human speech for entertainment. We represented this by sampling from prose from the Gutenberg Project (Gutenberg) and a video game dialogue dataset (van Stegeren and Theune, 2020). This type of text is similar to ransomware notes in that they are often speaking directly to the reader.

**System Files** The fourth data we collected were system files related to programs and scripts. Here, we utilized a dataset of READme files from 50 software projects (Capiluppi, 2020) and a dataset containing millions of lines of applications and system logs (Zhu et al., 2023). These data live on all systems and it is important to ensure that the model would determine that they are all supposed to be there. The log files are from a variety of systems, including nonmobile operating systems, mobile systems, distributed systems, supercomputers, and server applications (Zhu et al., 2023).

**Communications** The fifth data we collected were communications data. These are especially relevant if a ransom attack was being launched against a mobile device where such data is typically stored locally (though they may be stored locally on nonmobile devices as well). We represented this with a dataset of 627 translated German technical support emails (Rich, 2020). The technical support emails were explicitly chosen for their similarity to ransomware notes in vocabulary and structure. Additionally, because they were translated from German into English, they contain peculiarities of translated text. A limitation this brings up is that we do not have data translated from more languages, as many ransomware notes are written as if translated from another language. (Often a ransomware note composer seems not to be native speaker of English).

**Work-Related Files** The last type of data we wished to represent was school- or job-related files. We were able to represent this text type with math problems (Hendrycks et al., 2021) and resumes. The math dataset is a very comprehensive dataset featuring questions and answers to problems from high school math competitions across many disciplines in mathematics. The resumes are all publicly available, accessed through a Surge dataset (Surge). We only use the publicly available resume text, and remove the annotations to attend to copyright law.

The purpose of using all these diverse data was to help the model train robustly across different types of data. Each text type had its own generally accepted grammar conventions. For example, the prose data used highly standardized grammar, as they are from published works. However, much of the personal works and communications used very informal grammar and shorthand. The data also had varying degrees of "correctness": the instructables often contained typographical errors, as they are generally crowdsourced with little or no editing. All together, given the limitations of finding appropriate available data, the benign portion of the dataset is a close proxy to files that may be found on a system infected with ransomware.

### 3.2. Processing Data

To predict malicious intent from human language alone, each of these files, malicious and benign, needed normalization and detailed sanitizing.

All non-alphanumeric characters except for select punctuation (period, comma, and exclamation point) were removed. This ensured that certain unique features were kept. For example, '!!!' was a common punctuation sequence in ransomware notes that could be considered a malicious feature of written human language, although this particular punctuation sequence is also common to SMS writing (Muayyad and Chiad, 2008; Teh et al., 2015). Other features, such as malformed text and undisplayed binary characters, were stripped from the files. Whitespaces were standardized to a single

| |
|---|
| **Instructable**: Step 3: Step Two Connect one alligator clip to the negative on the Voltage tester, and the other alligator clip to the positive. Step 4: Step Three Set the Voltage tester to 1 volt DC. Rechargeable AA batteries need 12 volts for them to charge, this will allow you to see that you generate at least 1 volt. Make sure that the voltage is on DC and NOT AC |
| **Prose**: And you, niece, do me the favor to sigh and cry to your heart's content for the next ten years; for your confounded mania of sniveling, greatly as it annoys me, is preferable to these mad fits of rage. Because in the matter of Rosario and Jacinto I say to you, resignation? Because when every thing is going on well you turn back and allow Senor de Rey to get possession of Rosario. And how am I going to prevent it? Dona Perfecta is right in saying that you have an understanding of brick. |
| **BTC News Article**: But this emotional bearishness against crypto isn't just limited to the anticrypto crowd. Even the crypto market has an illogical emotional problem with certain assets. One of those assets is Litecoin. Litecoin is a fork of Bitcoin BTCUSD. It has 4 times the circulating supply, faster block times, and much cheaper transactions. While it has historically spent quite a bit of time as a Top 5 coin by market cap, it is currently placed at 20. |
| **Technical Support Email**: Dear Sirs and Madames I edited my XLS and Xtreme XLS with the process editing program. last week my PC had a defective one, so I had to reinstall the program. Since The messages appear again and again: No creation devices could be recognized in your system and Driver software for USB 1658 device needs to be installed. I have the download already performed X times for this installation process manager yellow white; unsuccessful. What am I doing wrong? |
| **READme**: Writing your own WebSocket Client The org.javawebsocket.client.WebSocketClient abstract class can connect to valid WebSocket servers. The constructor expects a valid ws: URI to connect to. Important events onOpen, onClose, onMessage and onError get fired throughout the life of the WebSocketClient, and must be implemented in your subclass. |
| **Math**: In triangleABC with side lengths AB 13, AC 12, and BC 5, let O and I denote the circumcenter and incenter, respectively. A circle with center M is tangent to the legs AC and BC and to the circumcircle of triangleABC. What is the area of triangleMOI? |
| **Logs**: Sun Dec 04 06:01:21 2005 notice workerEnv.init ok etc httpd conf workers2.properties<br>Sun Dec 04 06:01:30 2005 error modjk child workerEnv in error state 6<br>Sun Dec 04 06:01:42 2005 notice jk2init Found child 32352 in scoreboard slot 9<br>Sun Dec 04 06:02:01 2005 notice workerEnv.init ok etc httpd conf workers2.properties |
| **Game Dialogue**: The deal was struck, and the Heroes swiftly and cleverly set about stealing from the treethanes. From one they stole a prized bow. Another they tricked into handing over a valued necklace in exchange for all of the most valuable thing in the world that they could hold in their hands. The most valuable thing in the world, as any one knows, is air for without it we could not live. |
| **Resume**: Provided employees with tools to maintain and increase service levels to both internal and external customers. Increased employee knowledge by assisting with development and implementation of productawareness program. Served as InFlight Training Instructor. Emergency Safety First Aide, CPR Automated External Defibrillator Emergency Procedures Crew Resource Management to Line Holding Pilots Flight Attendants Security Serving Customer Service Aircraft Specifics I.O.E. Qualified. Education Bachelor of Arts : Psychology Social Sciences The University of Louisville |
| **Todo List**: camping: dvd movies sticks for smores baseball and glove kayak and paddle quinoa salad tp wipes chemicals dog leash harness cat food water water backpack dirty laundry bag juices projector stove with fuel colouring book bluetooth speaker mattress air mattress, pump pie crust therma rest picnic chairs waterproof phone holder play mat time sheet crystals wheel chock |
| **Journal Entries**: I went to my MRI to find out about the spot on my pancreas. I did not like it because I felt trapped in the machine. When I got home I had a vertigo attack that I think was caused by being in the MRI for so long. |

Table 1: Example excerpts from each data source used

| Obfuscated Text | Clarified Text |
|---|---|
| Remembe**tyT7ggBn03**r! The worst si**cdPSxiEKS**tuation already happ**M4cv4**ened and now the future of your files dep**rp**ends on your determ**FbH9DSsfs**ination and speed of your actions | Remember! The worst situation already happened and now the future of your files depends on your determination and speed of your actions |

Table 2: An excerpt from a Ransomware note file which inserted random bits of gibberish to obfuscate it

| Obfuscated Text | Clarified Text |
|---|---|
| If y\'eeu still w\'e0nt t\'ee try t\'ee d\'e5crypt th\'e5m by y\'eeurs\'e5lf pl\'e5\'e0s\'e5 m\'e0k\'e5 \'e0 b\'e0ckup \'e0t first b\'e5c\'e0us\'e5 th\'e5 d\'e5\'f1rypti\'een will b\'e5c\'eem\'e5 imp\'eessibl\'e5 in c\'e0s\'e5 \'eef \'e0ny ch\'e0ng\'e5s insid\'e5 th\'e5 fil\'e5s. | If you still want to try to decrypt them by yourself please make a backup at first because the decryption will become impossible in case of any changes inside the files. |

Table 3: An excerpt from a Ransomware note file which replaced random letters with escape codes to obfuscate it

space between words, and newline characters to no more than one consecutive instance. All HTML-based files (encompassing many files, including both some ransomware notes and benign files) were run through an HTML parser, html2text, so only the human language in the body of files was used, and not HTML tags nor undisplayed notes. Markdown files were converted to HTML with Markdown, then parsed as HTML files into plaintext. We cleaned all data via tailored Python scripts which are made available.

#### 3.2.1. Ransomware Note-Specific Processing

Some ransomware note files required extra cleaning. Ransomware authors often attempted to obscure their notes from detection by randomly inserting s and s of apparently randomly generated gibberish. They appear to rely on a parser to remove all tags with certain classes. This was manually reimplemented to properly sanitize these files. Table 2 displays an excerpt which shows this phenomenon.

The content of plaintext ransomware notes was often obfuscated by assigning certain escape codes to letters. This is shown in Table 3. In a similar manner to the HTML notes, gibberish escape codes were randomly inserted as well, but this is not shown.

### 3.3. Creating the Dataset

The data were aggregated, labelled as either benign or malicious, then chunked into sequences of 128 words. Ransomware note length ranged from a few brief sentences to over 1,200 words. Benign files ranged from a single sentence to hundreds of pages. Chunking yielded sequences of approximately 300-400 tokens after tokenization, ensuring all sequences fit into DistilBERT, which has a 512 maximum token limit per sequence (for discussion on model choice, see section 3.4). This prevents truncation of any sequence: preventing any data loss is critical in a low-resource context. Since the length of the sequences was often not evenly divisible by 128, any sequence with less than 10 words was discarded. This decision was made after examining the data and deciding that the results would not be negatively impacted.

All together, the dataset consisted of 2,531 files, as broken down in Table 4. The benign data were balanced for an equal number of chunks per data source, wherever possible: (sometimes, there is not enough data for a source to have as many chunks as the others). Since, for the most part, there was more of each of the benign data types than ransomware note data, we oversampled the benign files, then retroactively removed chunks at random, so that the dataset composition was as balanced as possible (i.e. an equal number of chunks from each data source). As a whole, however, the dataset is necessarily unbalanced since the number of ransomware note chunks is approximately 10% of the number of benign chunks.

The data was randomly split into training, validation, and test sets, as shown in Table 5. All chunks yielded from any particular file went into only one of the sets to minimize data overlap.

### 3.4. Training the Model

The distilbert-base-cased model was used. DistilBERT is a version of the BERT-based neural models. This model was chosen due to its wide use in text classification tasks (Sanh et al., 2020), and because it is a relatively small and fast model: only 4GB of GPU was available for use in this paper. While this did impose some limitations on maximum sequence length, we felt that this was the best option given the hardware limitations. The cased model (distinguishing between upper and lower case words) was used because letter case is

| | Files | | Chunks | |
|---|---|---|---|---|
| | Num. | Perc. | Num. | Perc. |
| **Prose** | 22 | .01 | 394 | .09 |
| **Support Emails** | 420 | .17 | 394 | .09 |
| **BTC Articles** | 31 | .01 | 353 | .08 |
| **Instructables** | 83 | .03 | 394 | .09 |
| **Todo Lists** | 293 | .12 | 147 | .03 |
| **Journal Entries** | 504 | .2 | 394 | .09 |
| **READmes** | 41 | .02 | 293 | .07 |
| **Resumes** | 84 | .03 | 394 | .09 |
| **Game Dialogue** | 378 | .15 | 394 | .09 |
| **Math Problems** | 459 | .18 | 394 | .09 |
| **System Logs** | 16 | .01 | 394 | .09 |
| **Benign Total** | **2351** | **.93** | **3945** | **.91** |
| **Ransom Notes** | **180** | **.07** | **394** | **.09** |
| **Total** | **2531** | **1.00** | **4339** | **1.00** |

Table 4: Number of files and the chunks derived from them in the dataset

| | Files | | Chunks | |
|---|---|---|---|---|
| | Number | Percent | Number | Percent |
| **Train** | 1772 | .70 | 3144 | .73 |
| **Valid** | 377 | .15 | 604 | .14 |
| **Test** | 382 | .15 | 591 | .14 |
| **Total** | **2531** | **1.00** | **4339** | **1.00** |

Table 5: Data split of files and the chunks derived from them in the dataset

an important feature for ransomware note files, as it is often employed to grab attention.

The model was trained using the following hyperparameters: learning_rate=1e-5, batch_size=32, epochs=10, weight_decay=0.02. The batches were split into 8 gradient accumulation steps of 4 items each to reduce memory usage, as necessary due to hardware limitations. A low learning rate and high weight decay were used to ensure smooth training and to reduce the potential for overfitting. Training was ended after epoch 10, because after that, the model began to increasingly overfit, regardless of hyperparameter setting.

## 4. Results

The results, as shown in Table 6, indicate that DistilBERT is reliably able to predict if a 128-word excerpt is either from a ransomware note or from a benign file, within a binary classification task. Note that in Table 6, each of the runs result in over 0.96 F1, demonstrating the high effectiveness of the classifier on the data. Since this dataset is unbalanced, the discrepancy between accuracy and F1 reflects the underlying structure of the dataset. Were the model to predict that everything was benign, the accuracy still would have been around 0.90 (recall Table 4, where roughly 90% of chunks were benign). This bias can be learned by the model, resulting in inflated accuracy scores.

Although these results may seem surprisingly good, they represent a modest improvement upon the previous methods for classifying on just filename: Random Forests, Support Vector Machine, Decision Tree, Naive Bayes, & Logistic Regression all achieve F1 scores of 0.890-0.920 (Lemmou et. al, 2021). However, content is far more robust in revealing obfuscation than filename, which can be changed to practically anything with little consequence to the attack. Since the victim ultimately needs to be able to read the ransomware note, these results represent better performance for a classification method that is far more powerful in practice. Lemmou et al. 2021 do apply Latent Semantic Analysis for similarity detection on their corpus, suggesting that they might have achieved similar results had they performed a classification task in addition to similarity detection.

Mager 2018 (Mager, 2018) trained a Naive Bayes classifier on ransomware note content using different data, achieving an F1 Score of .901, a precision of 0.823, and a recall of 0.994. Here also, we have a marked improvement in F1, due to a higher precision: our false positive rate is lower. However, it is of note that our recall is slightly lower, and in practical applications, a higher recall to the detriment of precision may be beneficial. Catching all malicious text along with some benign debris may be a more desirable outcome than letting some malicious text slip through.

For both of these comparisons, it is important to note that their results represent classification on whole files (directly or proxied via filename), and on different data. In our results, no file had more than one chunk classified incorrectly. If we classify each file by averaging the scores (0 for incorrect, 1 for correct) for each of their chunks, then rounding to 0 or 1, 100% of all files in the test set would be classified correctly in all 3 runs. This figure is technically more comparable to previous studies in terms of data type, but post-processing removes some ability to directly compare model performance.

### 4.1. Analysis of Incorrect Predictions

Upon investigation of the sequences that the model classified incorrectly, a few main features stood out. One feature was chunk length. Recall that chunk length was restricted to 10-128 words to prevent

| | | Acc. | F1 | Prec. | Rec. |
|---|---|---|---|---|---|
| Run 1 | Valid. | 0.997 | 0.981 | 1.000 | 0.962 |
| | Test | 0.995 | 0.973 | 0.965 | 0.982 |
| Run 2 | Valid. | 0.997 | 0.981 | 1.000 | 0.962 |
| | Test | 0.993 | 0.965 | 0.948 | 0.982 |
| Run 3 | Valid. | 0.998 | 0.991 | 1.000 | 0.981 |
| | Test | 0.993 | 0.965 | 0.948 | 0.982 |
| **Avg** | **Valid.** | **0.997** | **0.984** | **1.000** | **0.968** |
| | **Test** | **0.994** | **0.968** | **0.954** | **0.982** |

Table 6: Accuracy, F1 Score, Precision, and Recall for 3 Runs of the Model

data loss through truncation (see 3.3), given hardware limitations (see 3.4). Nevertheless, the model had difficulty with some of the shorter sequences. This may be caused by those sequences not having enough features to properly classify them. The other features seemed to involve the style of language employed, implying that the model was able to at least mimic an understanding of some linguistic choices that ransomware note authors typically make. Another major feature that stood out is that all false positives employed the grammatical second person, as shown in Table 7.

The only false negative in the test set, predicted incorrectly in all runs, is shown in 8.

| Source | Example |
|---|---|
| Game Dialogue | Then you are not worthy of further access. You will be rejected as unsuitable |
| Game Dialogue | Did she send you to find me? Please do not tell her you saw me! |
| Support Email | Error code 27 indicates that my programs are already registered under a different email address. Although my new email address walterfalkenbernd@einsundeins.de is registered with you, I have no way of changing the registration numbers for the new address. When trying to change, you go around in circles. Can you delete the old address walterfalkenbernd@hotmail.de? Or please let me know as soon as possible how I can work with Warehouse again. |

Table 7: Examples of False Positive Chunks. All employ the grammatical second person.

## 5. Discussion

The principal result that we demonstrate here is that, to neural classifiers, ransomware note text is meaningfully distinct from other common types of text. This may seem obvious, but given the lack of previous literature on the subject, it warrants a formal investigation. However, the implications of these results may not be as obvious as one might think.

There is the direct implication: neural classifiers can be used as a highly accurate means to automatically detect malicious activity on a system. As demonstrated by Mager, it is possible to build programs that use a ransomware note classifier to detect, alert, and mitigate potential intrusions in real-time (Mager, 2018). The use of neural classifiers in this way promises to make a contribution to the wide array of other technologies currently developed to mitigate ransomware attacks and control malicious actors.

In the larger context of enabling the detection and prevention of ransomware attacks, we envision a couple of possible applications for our model. First, such a model might be able to determine if there is currently any malware lying dormant during a system scan. Second, the model could be used to aid antivirus download scans, checking files for ransomware note text before they are allowed to interact with a system.

However, there are additional implications beyond the scope of ransomware mitigation. Ransomware attacks often prey on panic and distress. They have been described as a form of psychological warfare that exploits victims mentally, in addition to locking up their physical data (Sawler, 2016). In this paper, we take a first step towards using neural classifiers to computationally represent the particular language employed by attackers to exploit their victims. These results show that there is a meaningful way to computationally identify ransomware notes among various benign files by using the linguistic idiosyncrasies in the exploitative nature of ransomware note text. Ransomware notes are not the only form of exploitative text out there. Aside from the aforementioned work on scam emails and text messages, there is research that uses NLP-based classification techniques to identify cyberbullying (Raj et al., 2022) and cybergrooming (Isaza et al., 2022).

The current research adds to the prior successes of using NLP techniques to identify exploitative and malicious human language.

| False Negative Example |
| --- |
| Greetings,<br>We'd like to apologize for the inconveniences, however, your computer has been locked. In order to unlock it, you have to complete the following steps: 1. Buy iTunes Gift Cards for a total amount of 400.00<br>2. Send the gift codes to the indicated email address 3. Receive a code and a file that will unlock your computer. Please note: The nominal amount of the particular gift card doesn't matter, yet the total amount have to be as listed above. You can buy the iTunes Gift Cards online or in any shop. The codes must be correct, otherwise, you won't receive anything. After receiving the code and the security file, your computer will be unlocked and will never be locked again. Sorry for the inconveniences caused. |

Table 8: The only ransomware note text chunk classified incorrectly.

## 6. Contributions

We have presented novel results for the classification of ransomware notes. We achieve strong performance by tuning a pre-trained sequence classifier. From this, a strong, but narrow conclusion emerges: DistilBERT is able to almost perfectly distinguish between English ransomware note text and text from several types of benign files.

The cleaned dataset and code used in this paper are available on GitHub to support future research for the development of cyberattack mitigation tools.

## 7. Acknowledgements

This research was completed as course requirement fulfillment for the Advanced Cybersecurity Experience for Students (ACES) minor at the University of Maryland, College Park, supervised by Dr. Judith Klavans.

The collection of ransomware note files was made possible in part due to the generous sharing of data from RansomWhere by Jack Cable. In many ways, the work of Yassine Lemmou paved the way for this paper, laying the groundwork for ransomware note detection with Machine Learning, especially in regards to ransomware note collection. Lastly, we acknowledge the valuable feedback received from Dr. Smaranda Muresan.

## 8. Ethics Statement

All data collected and used exists in the public domain or is licensed for academic use, with the exception of the Instructables data whose copyrights belong to the respective authors. The project was approved by the ACES program at UMD.

## 9. Bibliographical References

Acronis. 2022. Acronis mid-year cyberthreats report 2022. Technical report, Acronis.

Amir Atapour-Abarghouei, Stephen Bonner, and Andrew Stephen McGough. 2019. A king's ransom for encryption: Ransomware classification using augmented one-shot learning and bayesian approximation. In *2019 IEEE International Conference on Big Data (Big Data)*, page 1601–1606, Los Angeles, CA, USA. IEEE.

Asma Baccouche, Sadaf Ahmed, Daniel Sierra-Sosa, and Adel Elmaghraby. 2020. Malicious text identification: Deep learning from public comments and emails. *Information*, 11(6):312.

Bala Baskar. 2022. Bitcoin - news articles text corpora. Accessed 2023.

Jack Cable. 2022. Ransomwhere: A crowdsourced ransomware payment dataset.

Andrea Capiluppi. 2020. DATA – collection of ReadMe files. Accessed 2023.

Zuguo Chen, Yanglong Liu, Chaoyang Chen, Ming Lu, and Xuzhuo Zhang. 2021. Malicious url detection based on improved multilayer recurrent convolutional neural network model. *Security and Communication Networks*, 2021:1–13.

Cynerio and Ponemon Institute. 2022. The insecurity of connected devices in healthcare 2022. Technical report, Cynerio and Ponemon Institute.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805 [cs].

Minghui Gao, Li Ma, Heng Liu, Zhijun Zhang, Zhiyan Ning, and Jian Xu. 2020. Malicious network traffic detection based on deep neural networks and association analysis. *Sensors*, 20(5):1452.

Project Gutenberg. Project gutenberg. Accessed 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*. Accessed 2023.

IBM. 2023. Cost of a data breach report 2023. Technical report, IBM.

Gustavo Isaza, Fabián Muñoz, Luis Castillo, and Felipe Buitrago. 2022. Classifying cybergrooming for child online protection using hybrid machine learning model. *Neurocomputing*, 484:250–259.

Gauri Jain, Manisha Sharma, and Basant Agarwal. 2018. Spam detection on social media using semantic convolutional neural network:. *International Journal of Knowledge Discovery in Bioinformatics*, 8(1):12–26.

Sujay Kumar Jauhar, Nirupama Chandrasekaran, Michael Gamon, and Ryen W. White. 2021. Mslatte: A dataset of where and when to-do tasks are completed. *arXiv preprint 2111.06902*. Accessed 2023.

Judith L. Klavans. 2015. Cybersecurity - what's language got to do with it?

Liam Larsen. 2020. Instructables diy - all projects. Accessed 2023.

Yassine Lemmou, Jean-Louis Lanet, and El Mamoun Souidi. 2021. In-depth analysis of ransom note files. *Computers*, 10(11):145.

X. Alice Li and Devi Parikh. 2019. Lemotif: Affective visual journal. Accessed 2023.

Mark Mager. 2018. Rapid anomaly detection via ransom note file classification.

Omran Muayyad and Muayyad Chiad. 2008. Structural and linguistic analysis of sms text messages. 7:1–13.

Mitushi Raj, Samridhi Singh, Kanishka Solanki, and Ramani Selvanambi. 2022. An application to detect cyberbullying using machine learning and deep learning techniques. *Sn Computer Science*, 3(5):401.

Jordan Rich. 2020. German technical support emails. Accessed 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv:1910.01108 [cs].

David R. Sawler. 2016. Ransomware: Psychological warfare in the cyber realm.

Saeid Sheikhi, Mohammad Taghi Kheirabadi, and Amin Bazzazi. 2020. An effective model for sms spam detection using content-based features and averaged neural network. *International Journal of Engineering*, 33(2).

Alex Shenfield, David Day, and Aladdin Ayesh. 2018. Intelligent intrusion detection systems using artificial neural networks. *ICT Express*, 4(2):95–99.

Sophos. The state of ransomware 2023. Technical report, Sophos.

SpyCloud. 2023. The 2023 spycloud ransomware defense report. Technical report, Spycloud.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? ArXiv:1905.05583 [cs].

Surge. Public resume categorization. Accessed 2023.

Phoey Lee Teh, Paul Rayson, Irina Pak, and Scott Piao. 2015. Sentiment analysis tools should take account of the number of exclamation marks!!! In *Proceedings of the 17th International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '15, New York, NY, USA. Association for Computing Machinery.

Judith van Stegeren and Mariët Theune. 2020. Fantastic Strings and Where to Find Them: The Quest for High-Quality Video Game Text Corpora. In *Intelligent Narrative Technologies Workshop*. AAAI Press. Accessed 2023.

Jianting Yuan, Guanxin Chen, Shengwei Tian, and Xinjun Pei. 2021. Malicious url detection based on a parallel neural joint model. *IEEE Access*, 9:9464–9472.

Jieming Zhu, Shilin He, Pinjia He, Jinyang Liu, and Michael R. Lyu. 2023. Loghub: A large collection of system log datasets for ai-driven log analytics. ArXiv:2008.06448 [cs].