




## Article

# Malcertificate: Research and Implementation of a Malicious Certificate Detection Algorithm Based on GCN

Jingru Liu <sup>1</sup> , Nurbol Luktarhan <sup>2,\*</sup>, Yuyuan Chang <sup>2</sup>  and Wenjie Yu <sup>1</sup> 

<sup>1</sup> School of Software, Xinjiang University, Urumqi 830000, China; ljru@stu.xju.edu.cn (J.L.); yu3022@stu.xju.edu.cn (W.Y.)

<sup>2</sup> College of Information Science and Engineering, Xinjiang University, Urumqi 830000, China; changyy@stu.xju.edu.cn

\* Correspondence: nurbol@xju.edu.cn

**Abstract:** Encryption is widely used to ensure the security and confidentiality of information. Because people trust in encryption technology, a series of attack methods based on certificates have been derived. Malicious certificates protect many malicious behaviors and threaten data security. To counter this threat, machine learning algorithms are widely used in malicious certificate detection. However, the detection efficiency of such algorithms largely depends on whether the extracted features can effectively represent the data. In contrast, graph convolutional networks (GCNs) can automatically extract useful features. GCNs are powerful at fitting graph data, which can improve the effectiveness of learning systems by efficiently embedding prior knowledge in an end-to-end manner. In this paper, we propose an algorithm for detecting malicious digital certificates with GCNs. Firstly, we transform the digital certificate dataset with pem document structure into a corpus of graph structure based on attribute co-occurrence and document attribute relations. Then, we put the graph structure certificate dataset into a GCN for training. The results of the experiment show that GCN is very effective in certificate classification and outperforms traditional machine learning algorithms and extant neural network algorithms. The accuracy of our algorithm to detect malicious certificates is 97.41%. This shows that our algorithm is very effective.

**Keywords:** cyber security; digital certificates; graph convolutional networks; data of the graph structure



**Citation:** Liu, J.; Luktarhan, N.; Chang, Y.; Yu, W. Malcertificate: Research and Implementation of a Malicious Certificate Detection Algorithm Based on GCN. *Appl. Sci.* **2022**, *12*, 4440. <https://doi.org/10.3390/app12094440>

Academic Editors: Leandros Maglaras, Helge Janicke and Mohamed Amine Ferrag

Received: 8 March 2022

Accepted: 24 April 2022

Published: 27 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As digitization plays an increasingly important role in people's lives, security of data and information during transmission has become an important issue. Encryption is widely used to protect the privacy of data as a technique to prevent data leakage. Because information is vulnerable to man-in-the-middle attacks during data transmission, HTTPS protocol is proposed to protect the security and stability of information communication, where SSL and TLS are encrypted communication frameworks. SSL, which is based on the HTTP standard and encrypts data transmitted by TCP, is a protocol layer on top of the TCP protocol and under HTTPS. TLS is an upgraded version of SSL and more secure. Certificate are a digitally signed document used in SSL/TLS protocol to verify identity. They are used to identify each participant in Internet interactions and to protect the confidentiality of information, the certainty of identity, the non-repudiation of transactions as well as the non-repudiation. To a certain extent, certificates reduce the possibility of users being attacked, but they also become a tool for attackers to carry out their attacks, causing great harm to the network. Certificates have also become increasingly important to protect users from cyber attacks.

The Computer Science Research Institute (CSRI) [1] has revealed information about stolen digitally signed certificates being sold and software's digital signature certificates being modified. In these ways, the attacker makes such certificates undetectable by browsers, allowing the malware to bypass the antivirus software and carry out the attack. Hackers

can use the CA certificates and private keys that they have obtained to issue fake certificates, perform SSL hijacking and listen to HTTPS traffic. According to the Anti-Phishing Working Group (APWG) [2] report on phishing activity trends for the third quarter of 2021, 90% of phishing websites use free DV certificates (such as those issued by Cpanel and Let's Encrypt, which do not require user authentication, only the domain name). Malware also often uses certificates to communicate with Command Control servers to avoid detection by traffic analysis tools. The SSL blacklist exposes numerous X.509 certificates. Because malicious certificates protect many phishing sites, they do a huge amount of damage to networks. The question of how to protect users from cyber attacks is becoming more and more important.

Although the certificate format is standardized, there are differences between certificates, and they are not structured data. When machine learning is used to classify certificates, the features of the certificates need to be extracted through feature engineering. However, a GCN does not need to use feature engineering methods to extract features [3–7]. It extracts features from the dataset automatically. Our Cert GCN model is based on an improved Text GCN [8–10] that allows the Cert GCN to convert a pem-structured certificate dataset into a graph-structured form, and then use the GCN to detect malicious digital certificates. In this paper, a GCN algorithm for detecting malicious digital certificates is proposed. The experimental results show that the accuracy of the algorithm is 97.41% for malicious certificates and 93.01% for benign certificates. The contributions of this paper are as follows:

- We design a rules-based method for extracting certificate attributes from documents to build a corpus of certificates. The advantage of this approach is that all useful information in certificates is extracted, saving time in constructing a corpus of certificates, and the constructed corpus of certificates is more comprehensive.
- The certificate graph is constructed by describing the unstructured certificate dataset with a heterogeneous graph. We extract the nodes of the graph and the relationship between nodes from the certificate corpus to construct a graph structure of the certificate that can better represent the certificate data.
- Cert GCN is proposed to coordinate and integrate heterogeneous information in certificate graphs. We use Cert GCN to detect malicious digital certificates and find that the accuracy is very high, up to 97.41%.
- We conduct experiments on the certificate dataset and compare the results with other models to prove the effectiveness of Cert GCN.

The rest of the paper is organized as follows: In Section 2, we describe previous efforts and results in malicious certificate detection. In Section 3, we propose Cert GCN for detecting malicious certificates. In Section 4, we describe our experimental procedure and results. Our experiments include the experimental setting, the dataset, the experiment design, and the analysis of the experimental results. Eventually, the conclusions of the work in this paper are discussed in Section 5.

## 2. Related Work

Traditional research on malicious certificate detection has focused on feature extraction and detection algorithms. We describe related work in more detail below.

### 2.1. Malicious Certificate Feature Extraction

Extracted certificate features play a very important role in the detection of malicious certificates, and the results of malicious certificate detection algorithms depend heavily on whether the extracted features can describe the data well. Feature extraction of digital certificates is either by manual methods or by using expert rules to extract valid fields. Manual methods of extracting features [11,12] obtain features directly from certificates, digitizing features with one-hot encoding [13], implementing discrete data vectorization with CLE, or processing data to get features with TF-IDF [14]. To address the shortcomings of manual feature extraction, tools for extracting features through expert rules were

designed and implemented. Jiaxin Li et al. designed and employed the VFE [15] method for certificate verification and feature extraction; the VFE extracts features through four parts: base analysis, criteria checking, certificate chain construction, and certificate chain verification. The extracted features were put into classical machine learning models and ensemble learning models for training. The ultimate result is that the ensemble learning model has better results for malicious certificate detection. Dong et al. [16] designed and implemented a certificate detection system containing a certificate downloader, a feature extractor, a classification actuator, and a decision-making section. The feature extractor uses expert knowledge to extract features in this system. This method is easier and less time-consuming, but blacklist expansion is time-consuming and tedious.

## 2.2. Malicious Certificate Detection Algorithm

There are two main types of detection algorithms: one is based on blacklisting malicious digital certificates and the other is based on machine learning [13,17] for malicious digital certificate detection. Ibrahim Ghafr et al. [18] proposed a malicious SSL certificate detection module (MSSLD) in response to APT attacks. It detects APT command and control (C&C) communication of malicious SSL certificate blacklists by matching the blacklist with a certificate or IP. This method is easier and less time-consuming, but the blacklist update is time-consuming and tedious. To solve this problem, it has been proposed that machine learning [19–21] be applied for malicious certificate detection. Akanchha et al. [22] proposed a system for automatically detecting phishing website systems using key attributes of SSL certificates. It uses different machine learning algorithms to do research utilizing extracted SSL certificate features and found that the decision-tree algorithm achieved better classification accuracy than others. Deep neural networks are also widely used in the field of network security, such as malicious certificate detection, backdoor attacks [23–26], and Android malware detection. Ivan Torroledo et al. [27] proposed a method that uses deep neural networks to identify web-based malicious certificates. It successfully identifies legitimate certificates and the malicious patterns used by attackers through the contents of the TLS certificate. The features of SubjectPrincipal and IssuerPrincipal were encoded with one-hot and trained by LSTM, and the results were fused with the other features extracted after training in the Dense/Relu layer. The system had an accuracy of 94.87% for the identification of malware certificates and 88.64% for the identification of phishing certificates.

Although all these models achieve good classification results, they all require feature engineering to extract the features of the certificate. Almost all features need to be identified by industry experts and then the features are manually coded. Eventually, the model can be trained to identify malicious certificates. The effectiveness of the model depends heavily on how well the features describe the data, which leads to the limitations of traditional machine learning methods. This problem is addressed by GCNs [28], which attempt to learn features in the data by themselves, significantly reducing the cost of feature discovery. GCNs achieve the best performance available for the node classification task, and they are also able to effectively take advantage of structural information of neighbors while retaining low-frequency signals. Jie Lu et al. [29] applied graph attention mechanism networks to website fingerprinting, using a GCN to learn intra-process and inter-process features. His work further demonstrates the advantages of GCNs over traditional machine learning methods.

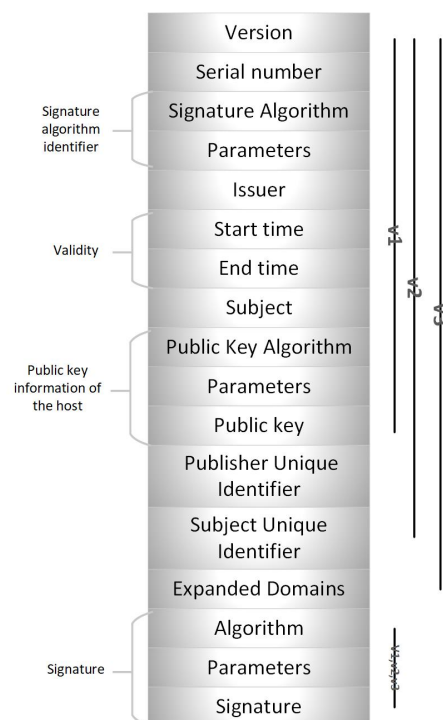
## 3. Method

In this section, we describe how a GCN algorithm can be used to detect malicious digital certificates. To begin with, we explain the process of certificate data pre-processing, and then we describe how the data can be used to detect malicious certificates in GCN.

### 3.1. Data Pre-Processing

#### 3.1.1. SSL/TLS Certificate

The digital certificate is an unencrypted file with a public encryption key that contains organizational details about the certificate owner and the encryption key. The format of the certificate is defined by the X.509 standard. X.509 certificates play an important role in encrypting data transmission between two parties under the HTTPS protocol. X.509 is a signed data structure that binds a public key to a person, computer, or organization and is used in many Internet protocols, including SSL/TLS [30,31]. X.509 is also complex in terms of structure and syntax. Each certificate consists of a sequence of three required fields: *tbcertificate*, *SignatureAlgorithm*, and *SignatureValue*. The first part, the *tbcertificate*, contains the subject, the publisher, and other basic information. Compared to Version 1 certificates, Version 2 certificates have added the *SubjectUniqueID* (subject unique identifier) and *trusteduniqueid* (issuer unique identifier) fields. In addition, extended fields have been added to the Version 3 certificate. The second part, *SignatureAlgorithm*, contains the identifier of the signature algorithm used by the certificate authority (CA) to sign the certificate. The third part, *SignatureValue*, records the digital signature calculated on *TBcertificate*. In a Public Key Infrastructure (PKI), certificates are usually organized together with their issuer into a certificate chain. The structure of the X.509 certificate is shown in Figure 1.



**Figure 1.** X.509 certificate structure.

#### 3.1.2. Data Preprocessing

Using a rules-based entity extraction approach, we build a heterogeneous certificate graph containing certificate attribute nodes and certificate document nodes. The original dataset for constructing the heterogeneous graph uses benign and malicious certificates (both in pem format). The benign certificates come from the top one million website ranking files from Alexa [32]. The certificate fingerprints of the malicious certificates are obtained from abuse.ch [33]. We search for malicious certificates with SHA1 values corresponding to the fingerprints of previously identified malware certificates. Then, we decrypt the certificate using the openssl toolkit to obtain X.509 standard structured certificate data. We put all the data into the corpus and tag each piece of data. The next step is data cleaning,

and finally, feature values in the certificates are extracted from each certificate's data by a python script that stores the feature values in a new corpus.

### 3.2. Certificate Graph Convolutional Networks (Cert GCN)

Since we wish to use the graph structure to represent the certificate dataset, which contains many discrete pem structure files, all of these certificate documents are packaged into a corpus of certificates, represented in graphical structure. To facilitate the study, this paper gives a formal definition of the structure of a certificate graph, which consists of edges of certificate nodes and certificate nodes.

**Theorem 1.**  $G_{cert}$  is a certificate graph that is composed of nodes, edges, and the adjacency matrix. It is formulated as  $G_{cert} = (V, E, \tilde{A})$ , where  $V = (V_{attribute}, V_{document})$  and  $\tilde{A} = A + I$ .

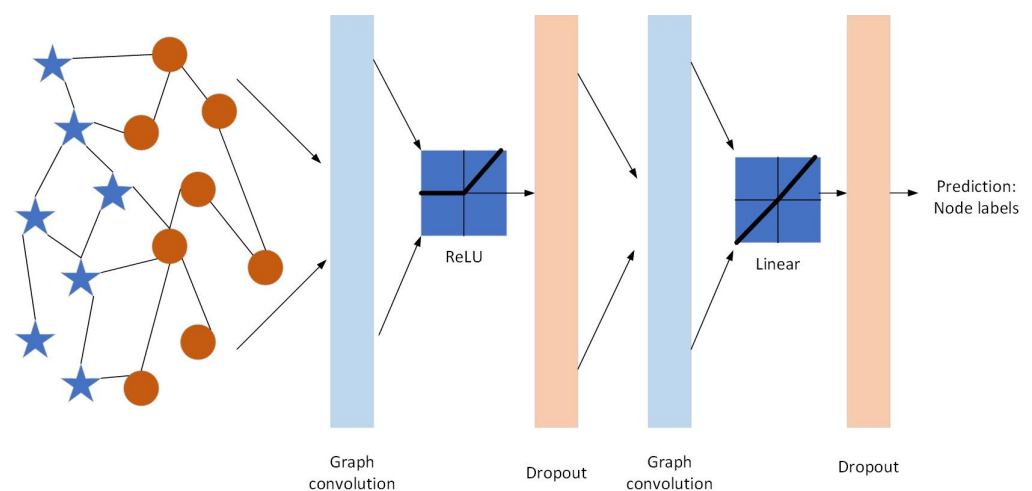
Where  $V_{attribute}$  is the set of certificate attributes, and  $V_{document}$  is the set of document names.  $E$  represents the relationship between nodes (document-to-attribute and attribute-to-attribute). The certificate graph structure is described in terms of the adjacency matrix  $\tilde{A}$  where  $A$  is constructed from the relationships between nodes and edges, and  $I$  stands for the identity matrix, which is added for each node to solve the problem of its own information loss.

We detect malicious certificates using a GCN. To facilitate the study of GCNs for the detection of malicious digital certificates, we define the formula for Cert GCN.

**Theorem 2.** Cert GCN refers to a GCN based on certificate documents and is used for malicious digital certificate detection.  $Z$  represents a two-layer Cert GCN, where  $Z = \text{linear}(\text{BReLU}(BXW_0)W_1)$ .

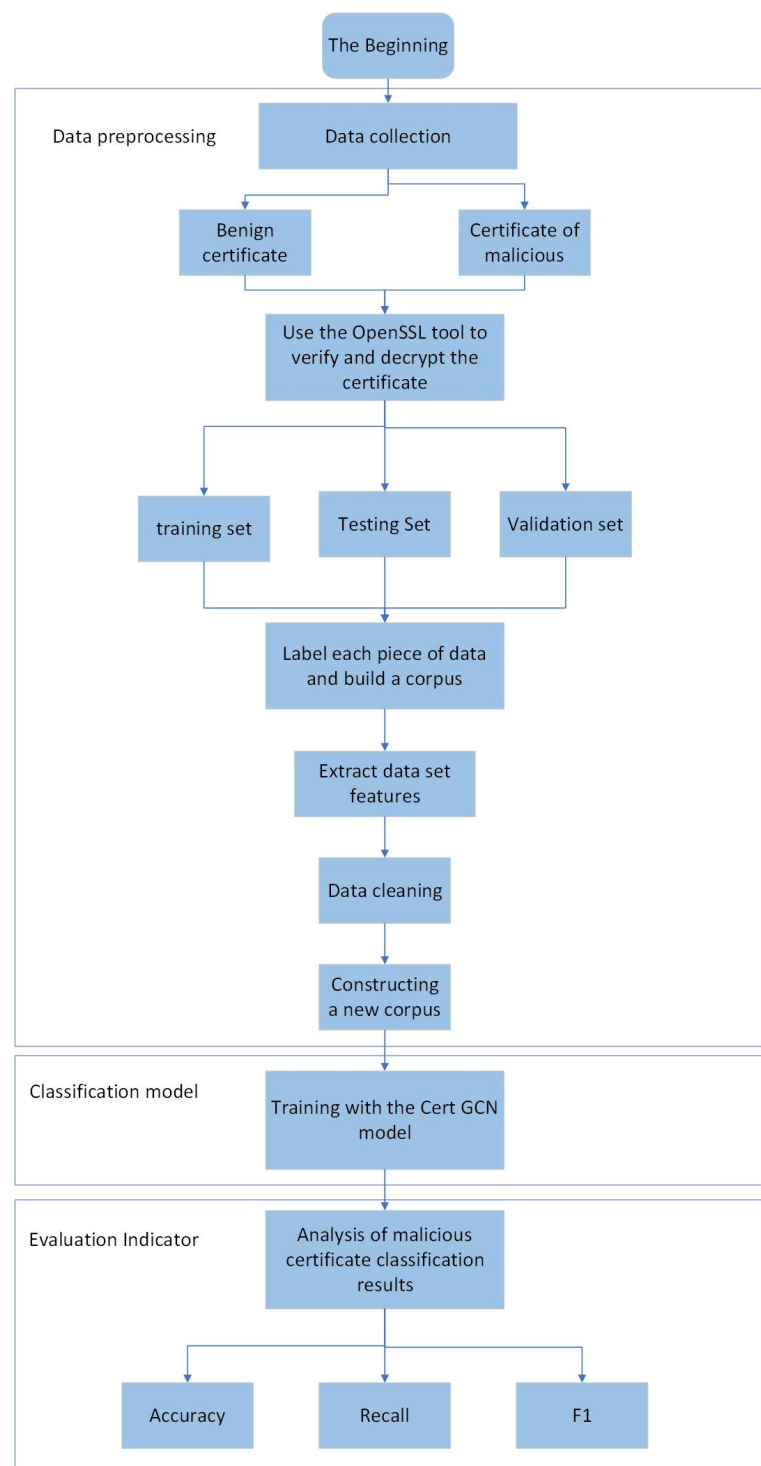
Where  $B = D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}}$ ,  $\tilde{A}$  is the adjacency matrix mentioned in Theorem 1,  $D$  is the degree matrix of  $A$  ( $D_{ii} = \sum_j A_{ij}$ ), and  $X$  is the input layer representation feature matrix. Each node  $i$  has feature  $x_i$ . The characteristic matrix of a node can be represented by the matrix  $X_{N \times M}$ , where  $N$  is the number of nodes and  $M$  denotes the number of features per node.  $W_0$  is the weight matrix for the first layer and  $W_1$  is the weight matrix for the second layer. The first layer of the activation function is ReLU. The second layer is the linear activation function.

In this paper, we constructed a dataset of certificate graph structures and put the dataset into the GCN layer. Each layer of the network consists of a convolutional layer, an activation function layer, and a dropout layer. The first layer of the activation function is ReLU. The second layer is the linear activation function. The constructed Cert GCN model is shown in Figure 2.



**Figure 2.** Schematic of Cert GCN.

The flowchart for malicious digital certificate detection is shown in Figure 3.



**Figure 3.** Malicious digital certificate detection flowchart.

## 4. Experimental Method

### 4.1. Experimental Environment

The software framework for use in the experiments was TensorFlow 2.6. The environment is established on a computer with the Windows 10 operating system and 8G of RAM. The code is edited using the Python 3.8 toolkit and the Anaconda integrated environment.



#### 4.2. Dataset

We use the openssl toolkit to verify whether the collected certificate files fit the X.509 certificate format standard so that the data can be cleaned [14]. The number of malicious and benign certificates changes after the cleaning, as shown in Table 1. Through using features of the certificate by means of rules-based entity extraction, we build a heterogeneous certificate graph containing certificate attribute nodes and certificate document nodes, and the features are described as shown in Table 2.

**Table 1.** Sample size of the dataset.

	Benign Certificates	Malicious Certificates
<b>Before decryption</b>	14,388	2267
<b>After decryption</b>	6438	1623

**Table 2.** Sample features from certificates.

Feature	Description
Version	Version information of the certificate.
Signature Algorithm	The signature algorithm of the certificate, the algorithm used to obtain the signature, corresponds to the OID.
Issuer	The issuer of the certificate, which generally adopts X.500 format, usually contain fields such as CN, O, L, S, C, E, G and OU.
Validity Not Before	The start date of the certificate validity period.
Validity Not After	The expiration date of the certificate validity period
Subject	The identifiable name of the certificate owner, generally with fields such as CN, O, L, S, C, E and G.
Public Key Algorithm	Public key signing algorithm for certificates.
RSA Public-Key	Public key encryption key for certificates—RSA
X509v3 Authority Key Identifier	The authorization key identifier for the certificate.
X509v3 Subject Key Identifier	The key identifier of the certificate subject.
X509v3 Subject Alternative Name	Optional name of certificate user.
X509v3 Extended Key Usage	Extended key usage.
X509v3 Certificate Policies	Certificate strategy.
Authority Information Access	Authorizer information access for certificates, including the URL of the certificate authority and the URL of the online certificate status protocol.
X509v3 CRL Distribution Points	URL information for the CRL distribution point.
X509v3 Key Usage: Critical	Key usage of the certificate.
X509v3 Basic Constraints: Critical	Whether the certificate is a CA certificate.
CT Precertificate SCTs Version	Certificate transparency version.
CT Precertificate SCTs Timestamp	Certificate transparency timestamp.
CT Precertificate SCTs Signature	Signature algorithm for certificate transparency

#### 4.3. Evaluation Indicator

In this paper, the effectiveness of the model is evaluated by accuracy, recall, and F1 value. The accuracy formula is  $\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN}$ . The formula for the recall rate is  $\text{Recall} = \frac{TP}{TP+FN}$ . The F1 formula is  $\text{F1} = \frac{2\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ .

#### 4.4. Experiment Design

The certificate corpus is obtained by pre-processing. The heterogeneous graph is constructed by using the extracted features and certificate document names as graph nodes in the certificate graph structure dataset. The weight between certificate document nodes and feature nodes is determined by the TF-IDF of the feature in the pem document, where TF is the frequency of the word in the document and IDF is the index of the inverse text frequency. The calculation formulas are as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

$$idf_i = \lg \frac{|D|}{|j : t_i \in d_j|} \quad (2)$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i \quad (3)$$

The relationship between two feature nodes is represented by the mutual information of the nodes. PMI [10] is a common metric for word association, and the formula for PMI is as follows:

$$\text{PMI}(i, j) = \log \frac{p(i, j)}{p(i)p(j)} \quad (4)$$

$$p(i, j) = \frac{\#W(i, j)}{\#W} \quad (5)$$

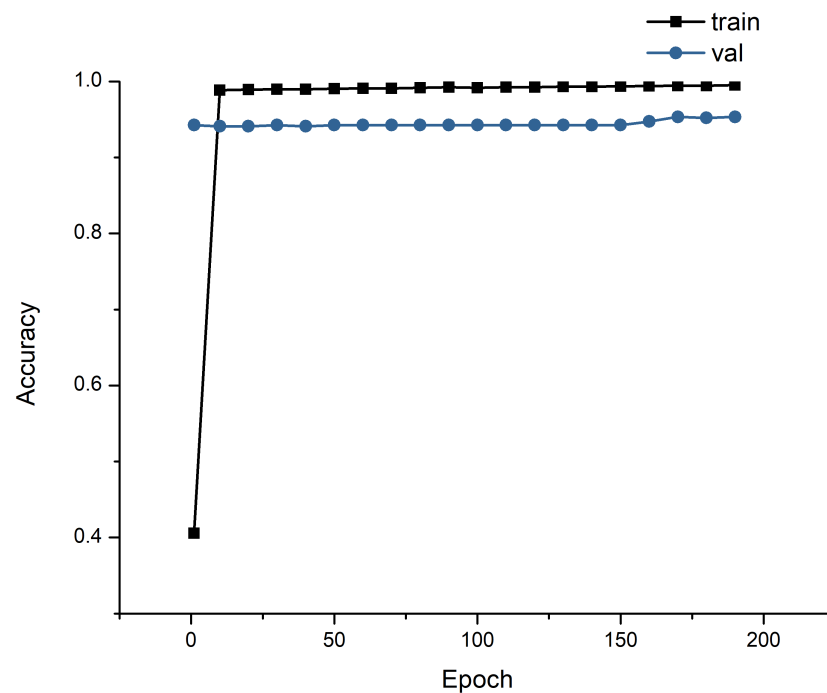
$$p(i) = \frac{\#W(i)}{\#W} \quad (6)$$

$\#W(i)$  represents the number of nodes  $i$  contained in the sliding window in the corpus, and  $\#W(i, j)$  represents the number of nodes  $i$  and  $j$  contained in the sliding window in the corpus.  $\#W$  represents the number of sliding windows in the corpus. These are the formulas for calculating PMI. The positive value of PMI indicates a high similarity of words in the corpus, while a negative value indicates low similarity. The weight of the edge between the node and itself is 1. The rest of the nodes have a weight of 0. The adjacency matrix of the certificate graph is then constructed. For the certificate GCN, the convolutional embedding size of the first layer was 200, the number of training sessions was set to 200, the learning rate was set to 0.005, dropout was set to 0.5, and weight decay was set to  $5 \times 10^{-4}$ .

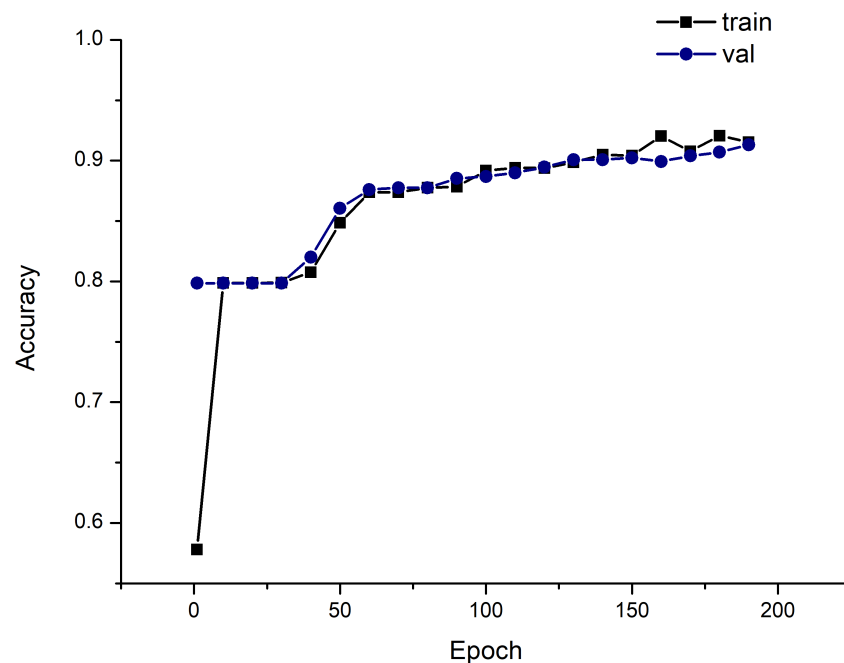
#### 4.5. Experimental Results

Comparing Cert GCN with Text GCN, the difference is that Cert GCN constructs graph certificate datasets using the attributes of the certificates and the names of the certificate documents as nodes, whereas Text GCN processes certificate datasets by constructing the corpus using the words in the certificate file and the certificate documents as nodes. They also use different activation functions and have different learning rates. Figures 4 and 5 depict the accuracy of the training and validation sets of Cert GCN and Text GCN [10], respectively, applied to malicious and benign certificates.



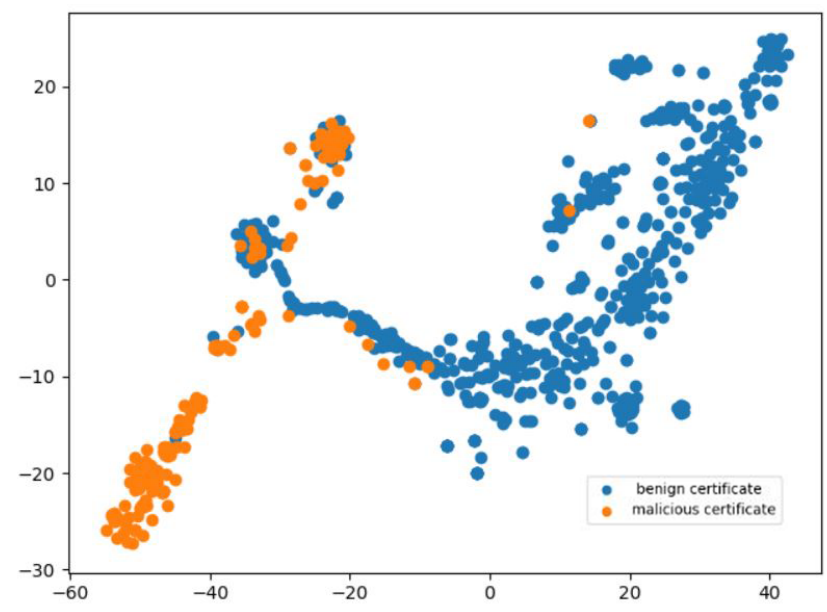


**Figure 4.** Classification accuracy of Cert GCN model on the training set and validation set of certificates.

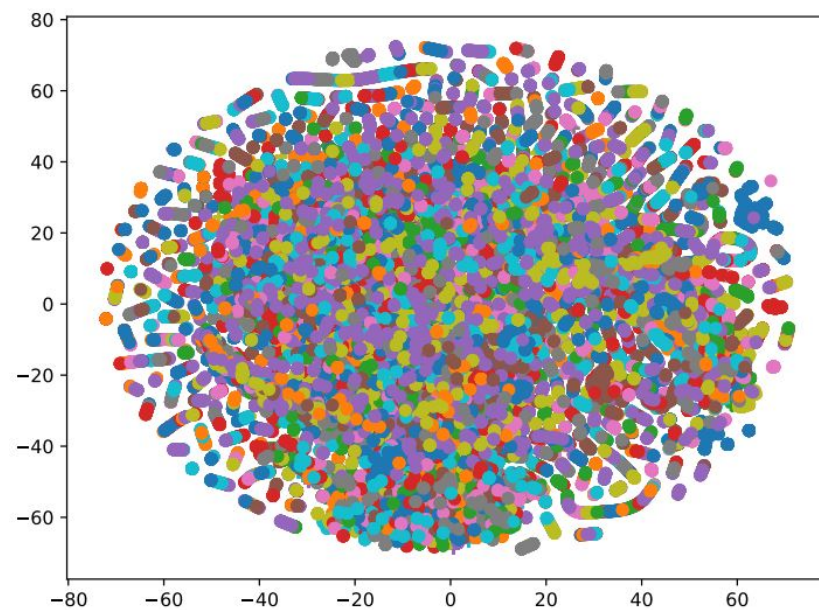


**Figure 5.** Classification accuracy of Text GCN model on the training set and validation set of certificates.

As the line graph shows, Cert GCN is 8% more accurate than Text GCN at the point at which the training set has the highest accuracy. Cert GCN is 4% more accurate than Text GCN at the point of highest validation set accuracy. In Cert GCN, the accuracy of the test set is close to 99%, the accuracy of the validation set is close to 95%, and the model is more stable. In Text GCN, the highest accuracy is 92% in the test set and 91% in the validation set, but the model is not yet stable, and there is an upward trend in the accuracy of the model. The fitting speed of Text GCN is slower than that of Cert GCN and the accuracy is not as high as that of Cert GCN. Figures 6 and 7 show the visualization of certificate node classification after training the model.



**Figure 6.** Visual classification of certificate documents by Cert GCN, in which yellow points are malicious certificate nodes and blue points are benign certificate nodes.



**Figure 7.** Visual classification of certificate attributes by Cert GCN. Different colors in the figure represent different documents, and each point represents an attribute of the document. There are 8061 digital certificate files in total.

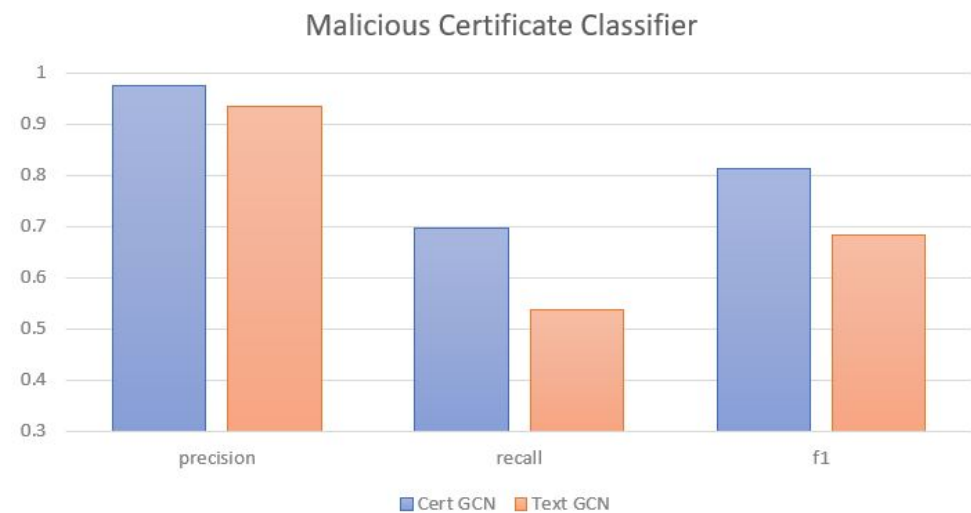
Tables 3 and 4 compare the evaluation of Cert GCN and Text GCN for detecting malicious and benign certificates. Figure 8 depicts the accuracy, recall, and F1 values of Cert GCN and Text GCN for malicious certificate classification. The diagram shows Cert GCN is 3.86% more accurate than Text GCN, and Cert GCN has 16.05% higher recall and 13.05% higher F1 than Text GCN. Figure 9 depicts the results of the two algorithms for benign certificate detection. The figure shows Cert GCN is 3.42% more accurate than Text GCN, and Cert GCN has 0.46% higher recall and 2.085% higher F1 than Text GCN. Cert GCN achieves good classification results.

**Table 3.** Evaluation of model results of malicious certificate detection by Cert GCN and Text GCN.

Model	Accuracy	Recall	F1
Cert GCN	97.41%	69.75%	81.29%
Text GCN	93.55%	53.70%	68.24%

**Table 4.** Evaluation of model results of benign certificate detection by Cert GCN and Text GCN.

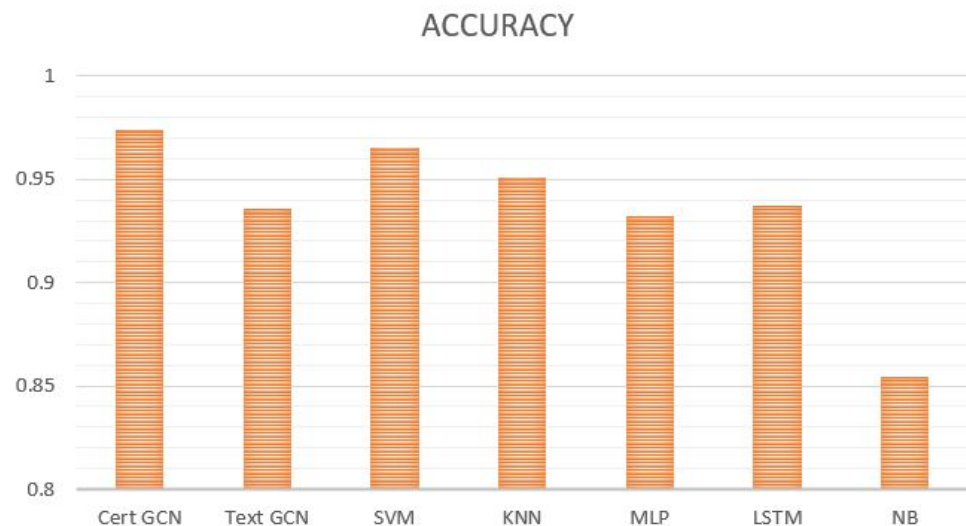
Model	Accuracy	Recall	F1
Cert GCN	92.89%	99.53%	96.10%
Text GCN	89.47%	99.07%	94.02%

**Figure 8.** Accuracy, recall, and F1 values for Cert GCN and Text GCN malicious certificate classifiers.**Figure 9.** Accuracy, recall, and F1 values for Cert GCN and Text GCN benign certificate classifiers.

In this paper, the accuracy of Cert GCN is compared with that of SVM, KNN, MLP, LSTM, NB, and Text GCN. We find that the accuracy of Cert GCN is better than that of all these other algorithms, and Cert GCN achieves 97.41% accuracy in detecting malicious certificates. The model with the lowest prediction accuracy is NB, which has an accuracy of 85.42%, as shown in Table 5 and Figure 10.

**Table 5.** Accuracy of different models for malicious certificate detection.

	Cert GCN	Text GCN	SVM	KNN	MLP	LSTM	NB
Accuracy	97.41%	93.55%	96.53%	95.10%	93.21%	93.72%	85.42%

**Figure 10.** Accuracy of various algorithms in predicting malicious certificates.

We found that Cert GCN does not perform better with more layers in classification compared with other neural networks. On the contrary, the more layers of Cert GCN convolution layer used, the lower the accuracy rate. Two-layer Cert GCN is the best for malicious certificate detection. There are also different results in the training process using different learning rates. If the learning rate is too large, the network will not converge and will hover around the optimal value. If the learning rate is too small, the network will converge too slowly and will fall into local extreme value convergence and not find the real solution. Experiments show that a learning rate of 0.005 works best. The maximum Chebyshev polynomial degree is set to 5, with a decrease in accuracy below and above 5. Dropout works best with a value of 0.5.

## 5. Conclusions

In this paper, we propose a new approach to malicious certificate detection (Cert GCN). The certificate dataset is processed by using certificate attributes and certificate documents as nodes of the graph and using attribute co-occurrence information as edges between nodes. We construct a heterogeneous graph for the certificate corpus. Cert GCN transforms the certificate detection problem into a node classification problem. A classification model was constructed based on GCN. The experimental results show that Cert GCN outperforms Text GCN and several other machine learning algorithms in terms of accuracy. Using Cert GCN, we address the problem with machine learning algorithms being unable to handle unstructured data and must therefore rely on whether the features extracted by feature engineering can accurately represent the data.

However, with the emergence of new malicious certificates, there is a growing problem of model degradation. There is still a space for improvement to enhance the accuracy of detection. The certificate dataset is relatively small and has data imbalance problems. Improvement of future work can: (1) address the problem of model degradation in malicious certificate detection by improving the model; (2) expand the dataset of benign and malicious certificates to alleviate the problem of low data size and data imbalance; (3) introduce other detection techniques of graph neural networks to improve the accuracy of malicious certificate detection.

**Author Contributions:** Conceptualization, J.L. and N.L.; methodology, J.L. and N.L.; software, J.L.; validation, J.L., Y.C. and W.Y.; formal analysis, J.L.; investigation, J.L. and W.Y.; resources, J.L. and Y.C.; data curation, J.L. and Y.C.; writing—original draft preparation, J.L.; writing—review and editing, J.L., W.Y. and Y.C.; visualization, J.L.; supervision, Y.C.; project administration, N.L.; funding acquisition, N.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded in part by National Social Science Fund of China under Grant 20&ZD293, and as part of the Innovation Environment Construction Special 355 Project of Xinjiang Uygur Autonomous Region under Grant PT1811.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

GCN	Graph Convolutional Network
SVM	Support Vector Machine
KNN	K Nearest Neighbors
MLP	Multilayer Perceptron
LSTM	Long Short-Term Memory
NB	NaiveBayesian Classification
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

## References

- Computer Science Research Institute. Available online: <https://cfwebprod.sandia.gov/cfdocs/CSRI/> (accessed on 5 December 2021).
- Phishing Attack Trends Report–2Q 2021. Anti-Phishing Working Group. Available online: <https://apwg.org/trendsreports/> (accessed on 21 February 2022).
- Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [\[CrossRef\]](#)
- Yang, Z.; Cohen, W.; Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 40–48.
- Cao, M.; Yuan, J.; Xu, M.; Yu, H.; Wang, C. Local Structural Aware Heterogeneous Information Network Embedding Based on Relational Self-Attention Graph Neural Network. *IEEE Access* **2021**, *9*, 88301–88312. [\[CrossRef\]](#)
- Wei, L.; Ye, X.; Xue, Y.; Sakurai, T.; Wei, L. ATSE: A peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. *Brief. Bioinform.* **2021**, *22*, bbab041. [\[CrossRef\]](#) [\[PubMed\]](#)
- Arora, S. A survey on graph neural networks for knowledge graph completion. *arXiv* **2020**, arXiv:2007.12374.
- Hu, H.; Yao, M.; He, F.; Zhang, F. Graph Neural Network via Edge Convolution for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
- Xiao, S.; Wang, S.; Dai, Y.; Guo, W. Graph neural networks in node classification: Survey and evaluation. *Mach. Vis. Appl.* **2022**, *33*, 4. [\[CrossRef\]](#)
- Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
- Bagaria, S.; Balaji, R.; Bindhumadhava, B. Detecting malignant tls servers using machine learning techniques. *arXiv* **2017**, arXiv:1705.09044.
- Mishari, M.A.; De Cristofaro, E.; Defrawy, K.E.; Tsudik, G. Harvesting SSL certificate data to identify web-fraud. *arXiv* **2009**, arXiv:0909.3688.
- Chen, C.; Diao, W.; Zeng, Y.; Guo, S.; Hu, C. DRLgencert: Deep learning-based automated testing of certificate verification in SSL/TLS implementations. In Proceedings of the 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), Madrid, Spain, 23–29 September 2018; pp. 48–58.
- Song, D.; Nurbol, W.Y. Malicious digital certificate detection method based on Catboost. *J. Chin. Comput. Syst.* **2021**, *30*, 1–8.
- Li, J.; Zhang, Z.; Guo, C. Machine learning-based malicious X. 509 certificates' detection. *Appl. Sci.* **2021**, *11*, 2164. [\[CrossRef\]](#)

16. Dong, Z.; Kapadia, A.; Blythe, J.; Camp, L.J. Beyond the lock icon: Real-time detection of phishing websites using public key certificates. In Proceedings of the 2015 APWG Symposium on Electronic Crime Research (eCrime), Barcelona, Spain, 26–29 May 2015; pp. 1–12.
17. El-Alfy, E.S.M. Detection of phishing websites based on probabilistic neural networks and K-medoids clustering. *Comput. J.* **2017**, *60*, 1745–1759. [[CrossRef](#)]
18. Ghafir, I.; Prenosil, V.; Hammoudeh, M.; Han, L.; Raza, U. Malicious ssl certificate detection: A step towards advanced persistent threat defence. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017.
19. Drichel, A.; Drury, V.; von Brandt, J.; Meyer, U. Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–12.
20. Dong, Z.; Kane, K.; Camp, L.J. Detection of rogue certificates from trusted certificate authorities using deep neural networks. *ACM Trans. Priv. Secur. (TOPS)* **2016**, *19*, 1–31. [[CrossRef](#)]
21. Gu, X.; Gu, X. On the detection of fake certificates via attribute correlation. *Entropy* **2015**, *17*, 3806–3837. [[CrossRef](#)]
22. Akanchha, A. *Exploring a Robust Machine Learning Classifier for Detecting Phishing Domains Using SSL Certificates*; Dalhousie University: Halifax, NS, Canada, 2020.
23. Kwon, H.; Kim, Y. BlindNet backdoor: Attack on deep neural network using blind watermark. *Multimed. Tools Appl.* **2022**, *81*, 6217–6234. [[CrossRef](#)]
24. KWON, H. Multi-Model Selective Backdoor Attack with Different Trigger Positions. *IEICE Trans. Inf. Syst.* **2022**, *105*, 170–174. [[CrossRef](#)]
25. Kwon, H.; Lee, S. Textual Backdoor Attack for the Text Classification System. *Secur. Commun. Netw.* **2021**, 2021. [[CrossRef](#)]
26. Kwon, H. Defending Deep Neural Networks against Backdoor Attack by Using De-trigger Autoencoder. *IEEE Access* **2021**. [[CrossRef](#)]
27. Torroledo, I.; Camacho, L.D.; Bahnsen, A.C. Hunting malicious TLS certificates with deep neural networks. In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, Berkeley, CA, USA, 19 October 2018; pp. 64–73.
28. Hamrouni, A.; Ghazzai, H.; Alelyani, T.; Massoud, Y. Low-Complexity Recruitment for Collaborative Mobile Crowdsourcing Using Graph Neural Networks. *IEEE Internet Things J.* **2021**, *9*, 813–829. [[CrossRef](#)]
29. Lu, J.; Gou, G.; Su, M.; Song, D.; Liu, C.; Yang, C.; Guan, Y. GAP-WF: Graph Attention Pooling Network for Fine-grained SSL/TLS Website Fingerprinting. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
30. Oh, S.; Kim, E.; Kim, H. Empirical analysis of SSL/TLS weaknesses in real websites: Who cares? In Proceedings of the International Workshop on Information Security Applications, Jeju Island, Korea, 25–27 August 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 174–185.
31. Paulson, L.C. Inductive analysis of the internet protocol TLS. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **1999**, *2*, 332–351. [[CrossRef](#)]
32. Alexa an amazon.com Company. Available online: <https://www.alexa.com/> (accessed on 1 October 2021).
33. A Research Project at Bern University of Applied Science. Available online: <https://abuse.ch> (accessed on 21 February 2022).