International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

CNN METHOD BASED CYBER HARRASMENT DETECTION ON SOCIAL NETWORK

Dr.M.Vargheese(Professor/HOD),

Sobana.S.MCSE department

PSNCET(Autonomous),Tirunelveli,India

ABSTRACT

Online social systems have become an important part of everyday life. Peoples used to share personal content with friends circle in the social networks.Unfortunately, Social Networks provide very little support to prevent unwanted messages on user walls. To overcome this issue, the Deep Neural Network algorithm is proposed in this system.This algorithm automatically block the unwanted text post message efficiently.This method is implemented in java platform with the front end Netbeans compiler and Mysql database as backend.

INTRODUCTION

ONLINE Social Networks (OSNs) allow users to create a public or private profile, encourage sharing information and interests with other users and communicating with each other. As a result, OSNs are being used by millions of people and they are now part of our everyday life. People use OSNs to keep in touch with family, friends, and share personal information, as well as for business purposes. Users of an OSN build connections with their friends, colleagues and people over time. These connections form a social graph that controls how information spreads in the social network. Although there is a dramatic increase in OSN usage – Facebook, for instance, has now 1.55 billion monthly active users, 1.31 billion mobile users, and 1.01 billion daily users1 there are also a lot of security/privacy concerns. One of the main source of these concerns is that OSN users establish new relationships with unknown people with the result of exposure of a huge amount of personal data.

The second issue regards how to model a 'normal behavior'. In doing this, we have to consider that OSN population is really heterogeneous in observed behaviors. However, similar to real world, we expect that similar users (e.g., similar in activity level, gender, education, country, and so on) tend to follow similar rules (e.g., moral, social) with the results of similar behavioral models. Based on this principle, we propose a two-phase risk assessment, where users are first grouped together according to some features meaningful for group identification.

Then, for each identified group, we build one or more normal behavior models. To reach this goal, the key contributions include determining various user features to model normal/anomalous behaviors, and integrating them with probabilistic-clustering approach (the Expectation Maximization algorithm). As it will be illustrated in the paper, we carried out experiments on a real Facebook dataset to show that the proposed two-phase risk assessment outperforms a simplified behavioral-based risk assessment where behavioral models are built over the whole OSN population, without a group identification phase.

Forged Messages in Online Social Networks

Several efforts have been made for more privacy aware Online Social Networks (OSNs) to protect personal data against various privacy threats. However, despite the relevance of these proposals, we believe there is still the lack of a conceptual model on top of which privacy tools have to be designed. Central to this model should be the concept of *risk*. Therefore, in this paper, we propose a risk measure for OSNs. The aim is to associate a risk level with social network users in order to provide other users with a measure of how much it might be risky, in terms of disclosure of private information, to have interactions with them. We compute risk levels based on similarity and benefit measures, by also taking into account the user risk attitudes. In particular, we adopt an active learning approach for risk estimation, where user risk attitude is learned from few required user interactions. The risk estimation process discussed in this paper has been developed into a Facebook application and tested on real data. The experiments show the effectiveness of our proposal Xu Wang et al carried out the analysis by taking a game theoretic approach, where infinitely repeated games are constructed to capture the interactions between a publisher and a network administrator and suppress forged messages in OSNs. Critical conditions, under which the publisher is disincentivized to publish any forged messages, are identified in the absence and presence of misclassification on genuine messages. Closed-form expressions are established for the maximum number of forged messages that a malicious publisher could publish. Confirmed by the numerical results, the proposed infinitely repeated games reveal that forged messages can be suppressed by improving the payoffs for genuine messages, increasing the cost of bots, and/or reducing the payoffs for forged messages. The increasing detection probability of forged messages or decreasing misclassification probability of genuine messages also has a strong impact on the suppression of forged messages.

Online Social Networks (osns) allow users to create a public or private profile, encourage sharing information and interests with other users and communicating with each other. As a result, osns are being used by millions of people and they are now part of our everyday life. People use osns to keep in touch with family, friends, and share personal information, as well as for business purposes. Users of an OSN build connections with their friends, colleagues and people over time. These connections form a social graph that controls how information spreads in the social network.

PROPOSED ALGORITHM

The neural network needs to learn all the time to solve tasks in a more qualified manner or even to use various methods to provide a better result. When it gets new information in the system, it learns how to act accordingly to a new situation.

Learning becomes deeper when tasks you solve get harder. Deep neural network represents the type of machine learning when the system uses many layers of nodes to derive high-level functions from input information. It means transforming the data into a more creative and abstract component.

In order to understand the result of deep learning better, let's imagine a picture of an average man. Although you have never seen this picture and his face and body before, you will always identify that it is a human and differentiate it from other creatures. This is an example of how the deep neural network works. Creative and analytical components of information are analyzed and grouped to ensure that the object is identified correctly. These components are not brought to the system directly, thus the ML system has to modify and derive them.

Deep neural network (DNN)

There are different types of neural networks and the differences between them lies in their work principles, the scheme of actions, and the application areas. Convolutional neural networks (CNN) are mostly used for image recognition, and rarely for audio recognition. It is mostly applied to images because there is no need to check all the pixels one by one. CNN checks an image by blocks, starting from the left upper corner and moving further pixel by pixel up to a successful completion. Then the result of every verification is passed through a convolutional layer, where data elements have connections while others don't. Based on this data, the system can produce the result of the verifications and can conclude what is in the picture.

Fig. Convoultional neural network

The use of deep layers of processing, convolutions, pooling, and a fully connected classification layer opened the door to various new applications of deep learning neural networks. In addition to image processing, the CNN has been successfully applied to video recognition and various tasks within natural language processing.

The DNN is a typical network architecture but includes a novel training algorithm. The DNN is a multilayer network (typically deep, including many hidden layers) in which each pair of connected layers is a restricted Boltzmann machine (RBM). In this way, a DBN is represented as a stack of RBMs.

In the DNN, the input layer represents the raw sensory inputs, and each hidden layer learns abstract representations of this input. The output layer, which is treated somewhat differently than the other layers, implements the network classificat ion. Training occurs in two steps: unsupervised pretraining and supervised fine-tuning

In unsupervised pretraining, each RBM is trained to reconstruct its input (for example, the first RBM reconstructs the input layer to the first hidden layer). The next RBM is trained similarly, but the first hidden layer is treated as the input (or visible) layer, and the RBM is trained by using the outputs of the first hidden layer as the inputs. This process continues until each layer is pretrained. When the pretraining is complete, fine-tuning begins. In this phase, the output nodes are applied labels to give them meaning (what they represent in the context of the network). Full network training is then applied by using either gradient descent learning or back-propagation to complete the training process.

The DNN consists of a set of modules, each of which is a subnetwork in the overall hierarchy of the DNN. In one instance of this architecture, three modules are created for the DNN. Each module consists of an inp ut layer, a single hidden layer, and an output layer. Modules are stacked one on top of another, where the inputs of a module consist of the prior layer outputs and the original input vector. This layering allows the overall network to learn more complex c lassification than would be possible given a single module.The DNN permits training of individual modules in isolation, making it efficient given the ability to train in parallel. Supervised training is implemented as back-propagation for each module rather than back-propagation over the entire network.

SYSTEM DESIGN

ARCHITECTURE

Architecture both the process and the product of planning, designing, and constructing buildings and other physical structures. Architectural works, in the material form of buildings, are often perceived as cultural symbols and as works of art. Historical civilizations are often identified with their surviving architectural achievements.

Use case diagram

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

ER DIAGRAM

An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database. An *entity relationship model*, also called an *entity-relationship (ER) diagram*, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

DFD

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a flowchart which also shows this information.

MODULES

- Network scenario
- Filtering rules
- Online setup assistant for FRS thresholds
- Blocked unwanted message

Network scenario

Given the social network scenario, creators may also be identified by exploiting information on their social graph. This implies to state conditions on type, depth and trust values of the relationship(s) creators should be involved in order to apply them the specified rules. All these options are formalized by the notion of creator specification, defined as follows.

Filtering rules

In defining the language for FRs specification, we consider three main issues that, in our opinion, should affect a message filtering decision. First of all, in OSNs like in everyday life, the same message may have different meanings and relevance based on who writes it. As a consequence, FRs should allow users to state constraints on message creators. Creators on which a

FR applies can be selected on the basis of several different criteria; one of the most relevant is by imposing conditions on their profile's attributes. In such a way it is, for instance, possible to define rules applying only to young creators or to creators with a given religious/political view.

Online setup assistant for FRS thresholds

As mentioned in the previous section, we address the problem of setting thresholds to filter rules, by conceiving and implementing within FW, an Online Setup Assistant (OSA) procedure. OSA presents the user with a set of messages selected from the dataset discussed in Section VI-A. For each message, the user tells the system the decision to accept or reject the message. The collection and processing of user decisions on an adequate set of messages distributed over all the classes allows to compute customized thresholds representing the user attitude in accepting or rejecting certain contents. Such messages are selected according to the following process. A certain amount of non neutral messages taken from a fraction of the dataset and not belonging to the training/test sets, are classified by the ML in order to have, for each message, the second level class membership values.

Blocked unwanted message

Similar to FRs, our BL rules make the wall owner able to identify users to be blocked according to their profiles as well as their relationships in the OSN. Therefore, by means of a BL rule, wall owners are for example able to ban from their walls users they do not directly know (i.e., with which they have only indirect relationships), or users that are friend of a given person as they may have a bad opinion of this person. This banning can be adopted for an undetermined time period or for a specific time window. Moreover, banning criteria may also take into account users' behavior in the OSN. More precisely, among possible information denoting users' bad behavior we have focused on two main measures. The first is related to the principle that if within a given time interval a user has been inserted into a BL for several times, say greater than a given threshold, he/she might deserve to stay in the BL for another while, as his/her behavior is not improved. This principle works for those users that have been already inserted in the considered BL at least one time.

SYSTEM TESTING

When a system is developed, it is expected that it performs properly. In practice, however, some errors always occur. The main purpose of testing an information system is to find the errors and correct them. A successful test is one, which find an error. The main objectives of the system testing are

- To ensure during the operation that the system will perform as per specified in the design phase.
- To make sure that the system meets user requirements during operations.
- To verify that the controls incorporated in the system functions as intended.
- To see that if correct inputs are fed into the system, it provides perfect output.
- To verify that during operation incorrect input processing and output will be deleted.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. If the testing conducted successfully, it will uncover errors in the software. As a secondary benefit, testing demonstrates that the software functions appear to be working according to specification and that performance requirements appear to have been made.The scope of the system test should include both manual operations and computer operations system testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.System testing is the process of checking if the developed system is working according to the original objectives and requirements. All testing needs to be conducted in accordance to the test conditions specifies earlier.This will ensure that the test coverage meets the requirement and that testing is done in semantic manner. System testing accounts for the largest percentage of technical effort in the software development phase.

There are three aspects of system testing:

- Unit testing
- Validation testing
- Sub System testing

UNIT TESTING

The application is tested in small units like functions so as to check that the function achieves the desired functionality when correct input is supplied. In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use.

Integration Testing

Integration testing is to test the project by connecting each module. The admin has to check whether the connection between the modules is correct.

`Validation testing

The application is tested to check how it responds to various kinds of input given. The user should be intimated of any kind of exceptions in a more understandable manner so that debugging becomes easier.
At the culmination of the black box testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected. Next stage is the validation testing and it can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in the manner that can be reasonably expected by the user. When an user enters incorrect inputs it should not display error messages, instead it should display helpful messages enabling user to use the tool properly. The tool is tested with test data as well as live data and has been found to work properly in the networked environment.

Verification Testing

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process. In the verification process, after giving the username and password the data is verified with the data in the database.

In most ways, System Security behaves in ways that are typical to rogue ant-virus programs. Infections typically occur by way of Trojan Horse, and the Trojan is hidden in some ordinary, harmless-looking thing you come across online. System Security infections are often the result of a Trojan hidden in a video codec, free download, or even a click-through pop-up screen on a website. That means that if your computer is infected with System Security, it is a surprise to you. This is all part of the scam's attempt to get you to think that System Security has some legitimate connection with Windows.

TEST CASE

A **test case**, in software engineering, is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a *test oracle*. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as *test scripts*, particularly when written - when they are usually collected into test suites.

MESSAGE BLOCKING SYSTEM

- Step 1: Social Network allocation
- Step 2: Friends registration
- Step 3: BL-> datalist
- Step 4: X->New text
- Step 4: send X to a friend
- Step 5: If (X==BL)
– then
– Block X
- Step 7: else
– Allow X to a friend
- Step 8: If repeated X
- Step 9: Block the sender of X

CONCLUSION

A two-phase risk assessment approach able to assign a risk score to each OSN user. This risk estimation is based on user's behavior under the idea that the more this diverges from what it can be considered as a 'normal behavior', the more the user should be considered risky. Experiments carried out on a real Facebook, google play and twitter dataset show the effectiveness of our proposal. We demonstrated here as some of our proposed work with the user registrations and the different network topologies and the admin server page.

Future work:

In future, this work can be extended according to several directions. An interesting future work is the extension of the proposed two-phase risk assessment so as to make it able to perform a continuous monitoring and estimation of risk scores.

Moreover, we plan to revise the risk assessment model so as to being deployable in Decentralized Online Social Networks, which are characterized by the absence of a central source of data to be analyzed. This will require to investigate decentralized data mining algorithms to gather user features.

REFERENCES

[1] Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. Privacy in social networks how risky is your social graph? In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pages 9–19. IEEE, 2012.

[2] Christa SC Asterhan and Tammy Eisenmann. Online and faceto- face discussions in the classroom: A study on the experiences of'active'and'silent'students. In Proceedings of the 9th international conference on Computer supported collaborative learning-Volume 1, pages 132–136. International Society of the Learning Sciences, 2009.

[3] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us automated identity theft attacks on social networks. In Proceedings of the 18th international conference on World wide web, pages 551–560. ACM, 2009.

[4] Yazan Boshmaf, Konstantin Beznosov, and Matei Ripeanu. Graphbased sybil detection in social and information systems. In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pages 466–473. IEEE, 2013.

[5] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Ler´ıa, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. ´Integro: Leveraging victim prediction for robust fake account detection in osns. In Proc. of NDSS, 2015.

[6] George Danezis and Prateek Mittal. Sybilinfer- detecting sybil nodes using social networks. In NDSS, 2009.

[7] Vacha Dave, Saikat Guha, and Yin Zhang. Measuring and fingerprinting click-spam in ad networks. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, pages 175–186. ACM, 2012.

[8] Vacha Dave, Saikat Guha, and Yin Zhang. Viceroi catching clickspam in search ad networks. In Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security, pages 765–776. ACM, 2013.

[9] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Compa- detecting compromised accounts on social networks. In NDSS, 2013.

[10] Mohammad Reza Faghani and Hossein Saidi. Malware propagation in online social networks. In Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on, pages 8–14. IEEE, 2009.