**FINAL YEAR PROJECT**

**GoogleECOM**

Energy Control & Packet Tracing Monitor

**FYP Team**

Khizar Naseer

Ibtisaam Butt

Malik Atif Aziz

Supervised by;

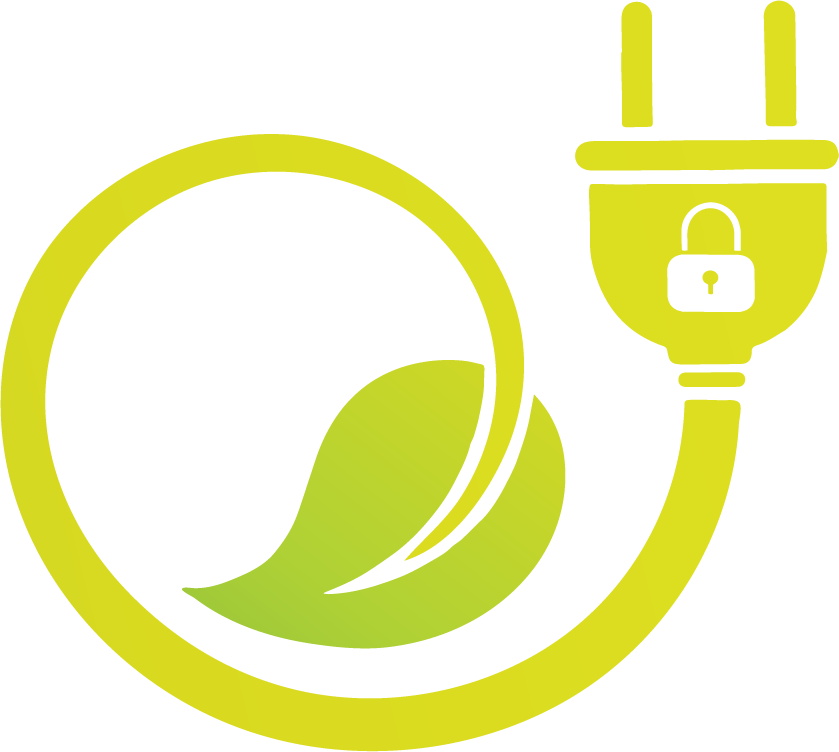
**Dr. Mirza Omer Beg**

Department of Computer Science

National University of Computer and Emerging Sciences

Islamabad, Pakistan

2018





Khizar | Ibtisaam | Atif

**Anti-Plagiarism Declaration**

This is to declare that the above publication produced under the:

Title: **GoogleECOM** is the sole contribution of the authors and no part here of has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. The work presented in the report is our own and it has not been submitted/presented previously to any other institution or organization. We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 17th October, 2018

**Student 1**

Name: Khizar Naseer Butt

Signature:

**Student 2**

Name: Ibtisaam Butt

Signature:

**Student 3**

Name: Malik Atif Aziz

Signature:

Supervisor (Faculty)

Name: Dr. Mirza Omer Beg

Signature:

Table of Contents

[**Chapter 1 Introduction** 5](#_Toc527525960)

[1.1 Problem Domain 5](#_Toc527525962)

[1.2 Research Problem Statement 5](#_Toc527525963)

[**Chapter 2 Literature Review** 6](#_Toc527525964)

[2.1 Using the Middle to Meddle with Mobile 6](#_Toc527525966)

[2.1.1 Summary 6](#_Toc527525967)

[2.1.2 Critical Analysis 7](#_Toc527525968)

[2.1.3Relationship with our Work 7](#_Toc527525969)

[2.2 ReCon: Revealing and Controlling PII Leaks in Mobile Network Systems 8](#_Toc527525970)

[2.2.1 Summary 8](#_Toc527525967)

[2.2.2 Critical Analysis 8](#_Toc527525968)

[2.2.3Relationship with our Work 9](#_Toc527525969)

[2.3 AntMonitor – A System for On-Device Mobile Network Monitoring 10](#_Toc527525972)

[2.3.1 Summary 10](#_Toc527525967)

[2.3.2 Critical Analysis 11](#_Toc527525968)

[2.3.3Relationship with our Work 11](#_Toc527525969)

[2.4 AntShield – On-Device Detection of Personal Information Exposure 12](#_Toc527525975)

[2.4.1 Summary 12](#_Toc527525967)

[2.4.2 Critical Analysis 13](#_Toc527525968)

[2.4.3Relationship with our Work 13](#_Toc527525969)

[2.5 Distilling the Knowledge in a Neural Network 14](#_Toc527525979)

[2.5.1 Summary 14](#_Toc527525967)

[2.5.2 Critical Analysis 14](#_Toc527525968)

[2.5.3Relationship with our Work 15](#_Toc527525969)

[**Chapter 3 Conclusion** 16](#_Toc527525983)

[Conclusion and Our Approach 16](#_Toc527525984)

[**References** 17](#_Toc527525985)

# Chapter 1

# Introduction

## **Problem Domain**

Around 4.57 billion\* Mobile Phones in the world frequently fail to provide users with satisfactory control over and visibility into how third-party applications use their private data. These applications have easy access to huge amounts[[1]](#footnote-1) of sensitive, personally identifiable information and there are several applications which transmit this information over the network and use them for their own endeavours and benefits. Ample amount of research has been done in past 7 to 8 years to gain control of fine-grained information being transmitted over the network and therefore, we aim to gain knowledge from recent findings and to move ahead in this area by enhancing and improving state-of-the-art algorithm for detecting PII leakage and calculating Energy Losses that users have to bear.

## **Research Problem Statement**

Improvement of current state-of-the-art algorithm by introducing new heuristics and/or ensemble methods for identifying Personally Identifiable Information in data packets transmitted over the network. In addition to it, we aim to classify the transmission as: (a) intended to collect information about the user and necessary for the operation of application; (b) intended to collect information about the user but of curious nature; (c) not intended by the user.

Finally, we plan to calculate the consumption of energy associated with the transmission of unintended and intended data packets that will assist in creating awareness and giving further insight to users about their data.

# Chapter 2

# Literature Review

This section contains a review on the work done in the past on Mobile Network Traffic and relation with what we plan to perform during our Research.

## **2.1 Using the Middle to Meddle with Mobile [1]**

This paper was published in CCIS, Northeastern University, Tech. Rep., in December 2012 by Ashwin Rao, A Molavi Kakhki, Abbas Razaghpanah, Amy Tang, Shen Wang, Justine Sherry, Phillipa Gill, Arvind Krishnamurthy, Arnaud Legout, Alan Mislove & David Choffnes. All of them have plenty of work to their name when it comes to monitoring the traffic that a mobile phone generates when it is connected to a network.

### **2.1.1 Summary**

The purpose of this study was to capture entire set of a mobile-device’s traffic over the internet: permitting them to characterize network flows, to interrupt them using software middleboxes, and to facilitate research into new middlebox applications for mobile traffic. The final goal of Meddle was to enable all mobile Internet users to monitor and control their Internet traffic.

Meddle [1] was the first approach to provide visibility and control over network traffic for all access network and most access decides Operating Systems, when introduced in 2012. It remarkably outwitted the work previously done [5][6][7][8] which had a lot of limitations associated with it such as: compromising network coverage, portability and deployability. Meddle compromised none of these and enabled visibility and control of network traffic access carriers, devices and access technologies.

Meddle [1] is a framework that combines virtual private networks (VPNs) with middleboxes to provide an experimental platform that aligns the interests of users and researchers. Meddle relies on VPN tunnels to access the mobile traffic regardless of the device, OS, wireless technology, and carrier. Meddle can thus provide a continuous and comprehensive view of how mobile devices interact with the Internet. Once packets arrive at a Meddle server, they use a variety of middlebox approaches to interpose on mobile-device traffic.

Meddle [1] offers new opportunities for measuring and characterizing mobile traffic, and designing new in-network features to improve the mobile experience. For example, by accessing network traffic regardless of the wireless technology we can analyze how different operating systems and apps offload their traffic from cellular networks to Wi-Fi. To improve the user experience and attract more users to install it, they implemented packet filters to block ads; unlike existing packet filters for mobile devices, the packet filters provided by Meddle do not require jail-breaking the mobile device.

Just like *Ant-Monitor* or *Ant-Shield*, which are part of our study during this project, Meddle worked on the layer where granularity of information is very high along with high number of users (which means that the idea is applicable and practical as compared to ISP Traces and User Traces where either you can get high number of users or fine-grained information at an instance, but not both).

After successfully capturing a packet, they use SSL bumping to decrypt and access the plain text of encrypted flows. Meddle’s VPN server can be configured to use a self-generated root certificate used to sign all subsequent certificates issued to participating mobile devices and this allows them to perform SSL traffic decryption using Squid proxy’s SSL bumping feature [13].

For capturing Personally Identifiable Information, they have introduced *ReCon* [10] which uses Meddle [1] on the top to capture and decrypt the packets. *ReCon* [10] has been separately reviewed by us during the literature Review. *ReCon* [10] used WEKA data mining tool to train their classifier. They utilized bag-of-words to separate features from their dataset. To reduce their number of features, *ReCon* [10] set a specific threshold of frequency for each word to be considered as a feature since when PII occurs it rarely occurs just once.

### **2.1.2 Critical Analysis**

There are few limitations associated with Meddle [1], their approach of capturing packets fails for apps that do not trust certificates signed by unknown root authorities. A better approach (SandroProxy library to intercept secure connections and to decrypt) has been used by Ant-Monitor (Reviewed separately).

### **2.1.3 Relationship with our Work**

There work is related to ours when it comes to intercepting the traffic over the network of a mobile phone when it is connected. In the same way, we aim to capture data-packets on the device and to identify if it’s a PII or not by first decrypting it and then calculating energy dosage/consumption during the transmission.

## **2.2 ReCon: Revealing and Controlling PII Leaks in Mobile Network Systems [10]**

This paper was published in MobiSys, in June 2016 by David Choffnes, Jingjing Ren, Ashwin Rao, Martina Lindorfer and Arnaud Legout. They have number of publications to their names in field of Privacy leak detection over the network.

The purpose of this study was to capture the personal information leaks from smartphones over the internet and give user control on it.

### **2.2.1 Summary**

ReCon [10] is a cross platform system that provides user with the view of their Personally Identified Information (PII) leaks and give command over it. User can block, substitute or permit PII. It works for Android, iOS and Windows phone. ReCon [10] used Machine Learning to identify PII leaks. They Performed a Controlled Experiment for collection of their dataset. In the Experiment they tested 100 Apps of Android, iOS and windows phone. They used each application for 5 minutes. ReCon used man in the middle approach to capture data packets.

Recon achieved following high level goals:

* Identify PII in network flow
* Improve awareness by presenting this information to user.
* Improve Classifier Accuracy using user feedback.
* Enable user to modify block or change PII leaks.

ReCon used WEKA data mining tool to train their classifier. They utilized bag-of-words to separate features from their dataset. To reduce their number of features, ReCon set a specific threshold of frequency for each word to be considered as a feature. They build a per domain classifier for those domains which send huge amount of data over the network. Machine learning classifier distinguishes information to be PII or not but rather does not show which content out of the data packet is PII. To resolve this issue, ReCon have a PII type (IMEI, andriodid e.t.c) and key assigned to it. ReCon estimates likelihood of each key to be PII or not by dividing the number of times the key shows up in PII flows with the number of times key show up in overall network flows. ReCon chose to utilize Decision Trees as classifier since DTs maintain great balance among accuracy and time it takes to train them. To additionally enhance their classifier ReCon took feedback from users whether PII was recognized accurately or not and retrain their classifier as needs be.

**2.2.2 Critical Analysis**

ReCon gathered extensive and clean dataset of 100 Applications form each OS (Android, iOS and Windows) which makes their classifier efficient. Their methodology of using Decision Trees gives slightly less accuracy than Blending Decision Trees and K Nearest Neighbors yet it is 7.24 times more affordable in time complexity. On the other hand a better methodology can be used, as ReCon utilize classifier each time it needs to identify PII, it makes process computational costly. They can have predefined strings of PII which can be utilized to distinguish PII by basic string matching as in case of AntMoniter. In the event that it fails they can use the Machine learning Model. Their methodology of identifying PII does not recognize whether PII was intended or not because of which any application which needs to send PII for instance, MAPS uses location (which it is supposed to send), will be identified as PII leak. There can be several approaches to identify whether PII was intended or unintended. One of them is to train a model on descriptions of application and then it can be predicted whether the leaked PII is needed for the working of the particular application or not. Second and a rather simpler approach requires us to check whether the user’s last interaction with that application is up to the particular threshold or not to mark the PII leak as intended or unintended.

**2.2.3 Relationship to our Work**

ReCon's study is closely related with our work in identification of PII leaks. We likewise need to identify PII in the first position to additionally give client capacity to control them and a visualization of energy being consumed. ReCon is utilizing machine learning classifier DTs to distinguish PII leaks we likewise need to build any ensemble method to recognize those PII leaks and further arrange them as expected or unintended PII flows.

## **2.3 AntMonitor – A System for On-Device Mobile Network Monitoring & its Application [3]**

This paper was published in arXiv preprint arXiv:1611.04268 in November 2012 by Anastasia Shuba, Anh Le, Emmanouil Alimpertis, Minas Gjoka, Athina Markopoulou. All of them have plenty of work to their name when it comes to monitoring the traffic that a mobile phone generates when it is connected to a network.

### **2.3.1 Summary**

The key emphasis of this paper was on AntMontior system for on-device passive mobile network monitoring. They have introduced a system for collecting large-scale, yet fine-grained network measurements from mobile devices, and for detecting and preventing leakage of private information in real time.

Major goals of their research were:

1. Privacy Leaks Detection and Prevention Module (PII detection).
2. Monitoring Network Performance.
3. Learning Network Behavior.

Their team thoroughly studied the work already done in the same area by (Haystack [2], Privacy Guard [9], Meddle [10]) and worked very well to produce better and efficient result in the following ways:

1. Their application works without root-access or administrator privileges.
2. It runs smoothly as a service app in the background.
3. It scales well with the increasing number of users.
4. It provides fine-grained control data (high-granularity of information being intercepted).
5. It supports real-time analysis on the device.
6. Their application is energy-friendly, along with the best throughput. (It achieves throughput of over 90 Mbps downlink and 65 Mbps uplink, which is 2x and 8x faster than mobile-only baselines and is 94% of the throughput without VPN, while using 2–12x less energy.)

For capturing the traffic (packets), they have used mobile-only VPN approach in contrast to client-server VPN, custom OS or Rooted Phone methodologies which have several disadvantages associated with them. After capturing the encrypted packets, they have used TLS proxy that uses SandroProxy library[11] to intercept secure connections, to decrypt the packets and then re-encrypt them before sending to their intended hosts. For the identification of personally identifiable information (PII), they have a list of strings, in which users can also add any potential PII (such as Mobile#, Citizenship# etc.) which further helps them in Deep Packet Inspection (It is an advanced method of examining and managing network traffic. It is a form of packet filtering that locates, identifies, classifies, reroutes or blocks packets with specific data or code payloads that conventional packet filtering, which examines only packet headers, cannot detect). This list of strings is then used to check if the packet being sent is a PII or not by string matching algorithms in real-time. It also notifies the user and intakes his/her action to allow/block/hash the information being transmitted.

As they had all the packets being transmitted, they used them to measure performance of the network to correlate network-level metrics with other information available on device with added advantage of fine-grained measurements at device or destination level.

### **2.3.2 Critical Analysis**

The only problem with AntMontior is that they used string matching only to identify the packet as PII or not which is not as accurate as if we use Machine Learning techniques. Otherwise, the approach of Mobile-Only VPN to gain access to the data packets gives them the best result as far as the performance of the algorithm and application is concerned.

**2.3.3 Relationship with our Work**

AntMonitor’s research is closely linked with what we plan to work on in the coming academic year. In the same but an enhanced way, we aim to capture data-packets on the device (without sending it to separate logger/server) and to identify if it’s a PII or not. However, there is a problem with their identifier (string matching) which can be easily dodged by a refined attacker; therefore, we need to deploy machine learning techniques by building an ensemble method to recognize and classify PII leaks. Another concern that has been ignored by AntMonitor is distinguishing if the PII being leaked is intended or not (Location needed by Google Maps is an intended PII leak). Therefore, in order to detect the intended or unintended transmission, another classifier will be built. To conclude, their work will surely assist us in several ways during our research.

## **2.4 AntShield – On-Device Detection of Personal Information Exposure [4]**

This paper was published in arXiv:1803.01261v1 [cs.NI] on 3 March 2018 by Anastasia Shuba, Evita Bakopoulou, Milad Asgari Mehrabadi, Hieu Le, David Choffnes, and Athina Markopoulou. Every one of them have a lot of work to their name with regards to checking the activity that a cell phone produces when it is associated with a network.

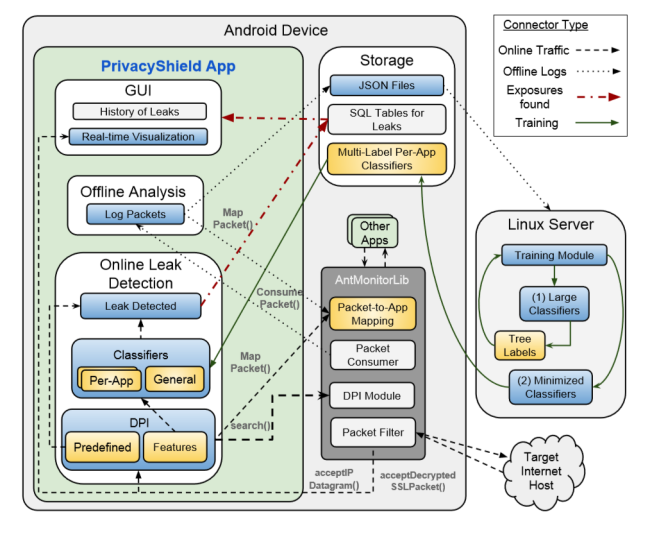
### **2.4.1 Summary**

With the advent of time, modern mobile phone devices have provided us with opportunities and facilities we could never think of. But with goodness comes a price. Our mobile phone devices have ingress to potentially sensitive, personal user data. With the boom of world wide web internet, a huge number of mobile applications transmit huge amount of data over the network. This data can be subject to as personally identifiable information which this particular paper has referred to as PII. AntShield- On-Device Detection of Personal Information Exposure, integrates the usefulness of *on-device* packet monitoring along with the recognition of identifying unknown PII. This particular system is the best state of the art in terms of system performance and accuracy, even out performing Recon [10].

Their research contributions as follows:

* A complete system to detect PII exposure via remote Virtual Proxy Network server in real time (~1 ms). They have used AntMonitor [3] library and its resource plug-ins to build this system which assimilate the draw backs that are developed due to other techniques such as customized OS or rooting a phone.
* The PIIs are identified by passing a network packet through 2 protocols (classifiers/ models). To identify a packet containing a potential PII, they use a string-matching algorithm heuristic to classify the packet. Also, their second classifier, multi-label classification model uses binary relevance with decision tress to identify unknown PIIs. The modular system of their application allows them to scale this feature if a better method is introduced.
* Their team built upon the dataset provided by Recon [10]. This richer than previously available dataset constitutes of information over UDP and TCP leaks as well along with HTTP leaks. They collected and analyzed this dataset with manual and automatic testing.

The core functionality of their system is PII exposure detection. This is done by maintaining calls to AntMonitor Library as depicted in the diagram below. Each intercepted packet sent over the network is analyzed with predefined leaks and then passed to the classifier for feature extraction. Their application gives the user the ability to choose whether to block the packet or allowed to continue to its remote destination.

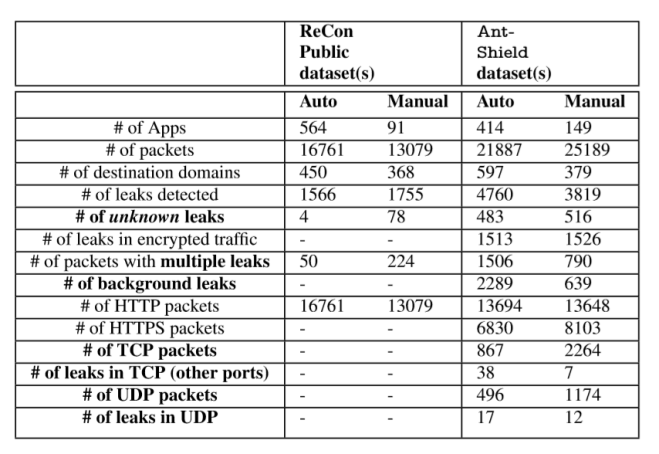


They also worked in giving interested and motivated users the ability to run the module offline after logging consumer packet over a period of time. Obviously, this would be computationally expensive but this can help the user reveal the ground truth.

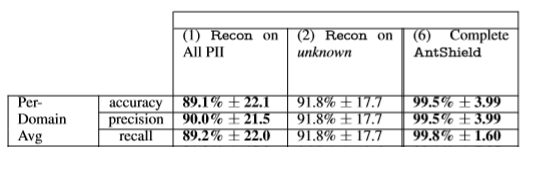
The AntShield application comes with pre-trained classifiers and does not log any data unless the user wishes to. Therefore, this application is space efficient when it comes to storage on device/or server. Along with it their application uses 100MB RAM which is acceptable as compared to many popular applications e.g, *Facebook* uses 200 MB of RAM.

### **2.4.2 Critical Analysis**

AntShield packet capture mechanism has several advantages compared to other state of the art mechanisms: (a) They were able to map packet to its source and destination accurately; (b) They picked up understanding into TLS, UDP, and general TCP trafﬁc, along with HTTP.



Their evaluation matrix consists of binary classification: F-measure, specificity and recall. The details are as follows:



### **2.4.3 Relationship with our Work**

AntShield is the latest advancement in the research area we wish to pursue. They are nearest to our work, in this manner and we will utilize it as our standard for correlation all through our work. The key difference would lie in the classification methodology and the eventually the energy report we wish to achieve. Their technique creates new routes for distributed learning of individual data leakage which presents one of a kind framework challenges and learning opportunities.

## **2.5 Distilling the Knowledge in a Neural Network [17]**

This paper was published in arXiv:1503.02531 on 9 Mar 2015 by [Geoffrey Hinton](https://arxiv.org/search/stat?searchtype=author&query=Hinton%2C+G), [Oriol Vinyals](https://arxiv.org/search/stat?searchtype=author&query=Vinyals%2C+O), [Jeff Dean](https://arxiv.org/search/stat?searchtype=author&query=Dean%2C+J).

### **2.5.1 Summary**

Cumbersome models generalize the data much better than small models but they are difficult to deploy because of their computational complexity. It is possible to compress the knowledge in a cumbersome (very deep network or ensemble of different networks) model to a small neural network. During distillation, they use soft targets (softmax results of cumbersome model) as ground truth to train a smaller network. Using soft targets instead of hard targets provides much better information of relative probabilities among different classes. For example, in case of image classification it gives you information that one type of fish is more closely related to another type of fish than different types of dogs.

They use modified version of softmax when training a model cumbersome or shallow network) for distillation. It is given as:

S = ezi/Tezj/T

Where zi is the resultant logit of neuron i and T is temperature constant. Increasing the temperature increases the affect of classes having very low probabilities by producing reasonably soft targets. Fig (a) shows the result of different values of T on probabilities given by softmax.

After computing softmax, they use modified cross entropy loss by using pi to be equal to the target that cumbersome model produces on that specific data point.

L=pi\*log (qi)

Then they take derivative of this function and put it equal to zero. It turns out that they are trying to make qi = pi. Training ensemble of neural networks might be very costly so one thing that we can do is to train specialist models on subsets of data and then use distillation loss to transfer their knowledge to the small network. This method also acts as a regularizer and reduces over-fitting on data.

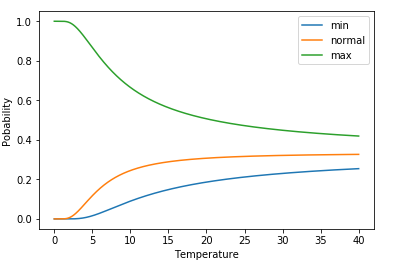


Fig (a): Effect of temperature on softmax scores

### **2.5.2 Critical Analysis**

This method works very well for classification. On MNIST dataset small network of two layers with 800 RELUs on each gives 146 errors on test set but these errors go to 67 after distilling the knowledge from a deep network.  Omitting the digit 3 from training set and distilling the knowledge from a deep network, with bias adjusted, gives 14 test set errors.

This paper does not discuss the variant of loss that can be used for regression purpose. Also, this paper introduces 3 new hyper-parameters in the models (T, weight for cross entropy and distillation, and bias correction for missing training data).

### 

### **2.5.3 Relationship with our Work**

Most of state-of-the-art algorithms and application face the problem of having to settle between either accuracy or computational cost. In this regard, we aim to train our classifier be it Decision trees or Convolutional Neural Networks, and then convert it into a smaller neural network using distillation loss and deploy it in our final product. This would significantly help us in saving computational resources and at the same time maintaining, if not increasing, the accuracy/ output of the system.

# Chapter 3

# Conclusion and Our Approach

Our proposed approach is an on-device network monitoring paradigm which uses a Virtual Private Network setup on the device to route the packets from the application to their server domain. This precludes the necessity for trust agreement and gives full autonomy to the user. Our system would consist of the following steps:

1. We build on AntMonitor Library for intercepting packets on the device.
2. Our system would then search for pre-defined strings in the data packets using string matching heuristics or regular expressions.
3. For unknown Personal Identifiable Information (PIIs), we would build a classifier which would identify a packet as PII in *real time*.
4. We aim to improve the current state of the art algorithms used for detecting PIIs by introducing distillation loss and converting the complex classifier into a deep neural network with less complexity but same, if not improved, accuracy. This new ensemble method would be less computationally expensive and would drastically improve efficiency.
5. We also aim to classify the packets as: (a) intended and required; (b) intended but skeptical; (c) not intended.
6. This would require Operating system calls (which do not require rooting a phone) such as RAM information, applications loaded and application nature/ behavior to classify the transmitted packet into its specific type.
7. In the end, we aim to calculate the consumption of energy used for the transmission of these PIIs.
8. For this purpose, we would require a regression model built on an energy dataset to give us an idea of how much energy is being utilized for the needless transmission.
9. This information would be displayed to user in the form of values for approximate energy consumed and energy graphs.

Conclusion:

We aim to empower the user to give them full autonomy and control over their mobile phones when it comes to transmission of their personal data over the network. In order to achieve this objective, we propose a VPN based approach which uses classifiers (heuristics/ ensemble methods) and regression models to classify the data leakage and give an insight about energy consumption to the user.

# **References**

1. A. Rao, A. M. (n.d.). *Using the Middle to Meddle with Mobile.* Technical report, Northeastern University, Dec. 2013.
2. A. Razaghpanah, N. V.-R. (n.d.). Haystack: In Situ Mobile Traffic Analysis in User Space. arXiv preprint arXiv:1510.01419, Oct. 2015.
3. A. Shuba, A. L. (n.d.). Antmonitor: Network trafﬁc monitoring and real-time prevention of privacy leaks in mobile devices. In ACM Mobicom Demo and Short Paper (and best demo in S3), September 2015.
4. Anastasia Shuba, E. B. (n.d.). AntShield – On-Device Detection of Personal Information Exposure. arXiv:1803.01261v1 [cs.NI] on 3 March 2018.
5. CHALLEN, G. (n.d.). *Phonelab testbed*. Retrieved from http://www.phone-lab.org.
6. CHEN, X. J. (n.d.). Network Performance of Smart Mobile Handhelds in a University Campus WiFi. In Proc. of IMC (2012).
7. ENCK, W. G.-G. (n.d.). A. N. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In Proc. of OSDI.
8. GERBER, A. P. (n.d.). S. Speed Testing without Speed Tests: Estimating Achievable Download Speed from Passive Measurements. In Proc. of IMC (2010).
9. Hengartner., Y. S. (n.d.). PrivacyGuard: A VPN-based Platform to Detect Information Leakage on Android Devices. In Proc. of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, Oct. 2015.
10. J. Ren, A. R. (n.d.). Recon: Revealing and controlling pii leaks in mobile network trafﬁc. . In ACM MobiSys), volume 16, 2016.
11. *Secure Android Proxy.* (n.d.). Retrieved from https://code.google.com/archive/p/sandrop/
12. SOMMERS, J. A. (n.d.). P. Cell vs. WiFi: On the Performance of Metro Area Mobile Connections. In Proc. of IMC (2012).
13. *Squid-in-the-middle SSL Bump.* (n.d.). Retrieved from http://wiki.squid-cache.org/Features/SslBump.
14. *strongSwan VPN Client.* . (n.d.). Retrieved from https://play.google.com/store/apps/details?id=org.strongswan
15. VALLINA-RODRIGUEZ, N. S. (n.d.). J. Breaking for Commercials: characterizing Mobile Advertising. . In Proc. of IMC (2012).
16. WANG, Z. Q. (n.d.). M. An Untold Story of Middleboxes in Cellular Networks. In Proc. of ACM SIGCOMM (2011).
17. Geoffrey Hinton, O. V. (n.d.). Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

1. Statistics according to Statista. https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/ [↑](#footnote-ref-1)