HappyKidz: Privacy Preserving Phone Usage Tracking

Benjamin M. Case[1], Marcella Hastings[2], Siam Hussain[3], and Monika Trimoska[4]

[1] *Clemson University*
[2] *University of Pennsylvania*
[3] *University of California San Diego*
[4] *University of Picardie Jules Verne*

Abstract

We propose a smartphone app named *HappyKidz* that allows parents to monitor their child's well-being in a non-invasive way based on measurable behavioral indicators. The app collects behavioral data on smartphone usage, encrypts them with homomorphic encryption, and sends the encrypted data to a server. The server calculates a well-being score for the child using a trained neural network and sends the resulting encrypted score to the parent to decrypt locally. This architecture takes advantage of modern machine learning techniques while maintaining privacy for individual children from the server. Unlike existing apps, it does not directly control, access or report raw behavioral data. In this work, we describe the high-level application and implement a proof of concept of the core neural network logic. We address concerns about the appropriate use of the app and discuss potential barriers to implementation, including collecting appropriate training data and scaling the model to a larger feature set.

1 Introduction

Smartphones have become indispensable parts of our daily lives. Along with adults, children are also using them for both education and entertainment. However, the adverse effects of excessive phone usage have created concerns among parents and social scientists. While these effects are observed in children and adults alike, children are considered to be more susceptible [YR98, BP05, SMO09, HBC17, CRC^{+}16, TZW^{+}19].

To tackle this issue, several apps are designed to allow parents to oversee the phone usage of their children. A study on popular parenting apps in the Google Play Store identified two key features of these apps - remote monitoring and remote locking [KCY+15]. Another study on the acceptance of these apps among children has reported that the ratings given by the children to these apps are significantly lower than those given by the parents [GBUG+18]. According to this study, children felt that the apps were overly restrictive and invasive of their privacy, which negatively impacts their relationship with parents.

The existing apps have two limitations. First, it annoys the children, especially teenagers, who want more control over their lives, thus leading to more complex problems and worsening the situation in many cases. “Hover parenting”, has been associated with increased levels of child anxiety and depression [SLMM+14, LO18, GBUG+18]. Second, it is often difficult to determine when a phone usage pattern becomes unhealthy. Most mental health apps approved by the Anxiety and Depression Association of America [ADA] are targeted toward individual, self-guided management of existing disorders or are designed to be used in tandem with a licensed therapist. Moreover, the signs of depression in the children often go unnoticed by the parents. A poll by the University of Michigan [CFD+19] suggests that two-thirds of parents face barriers in recognizing depression in their own children.

In this work, we aim to help the parents effectively monitor the well-being of their children in a non-invasive way. We propose an app, called *HappyKidz*, that automatically collects usage data from a child’s phone and sends it to a server. The server holds a Machine Learning (ML) model that is trained collaboratively by a large number of parents as well as child psychologists and social scientists to calculate a well-being score of the child. The parents receive a periodic update of the score on their phones. In this way, instead of constantly monitoring the child’s usage, parents only need to intervene if there is a drop in the well-being score.

While this approach solves the above-mentioned limitations of the existing apps, it brings a more crucial issue - protecting the privacy of the child’s data. Allowing the server to view the raw data creates the possibility of corporate misuse, e.g., using knowledge of depressive behaviors to tailor predatory advertisements or selling health data to insurance companies or other partners. In the proposed app, to ensure the privacy of the child the collected data is encrypted locally with Homomorphic Encryption (HE) at the child’s phone before being sent to the server. The server computes the well-being score by performing ML on the encrypted data and sends the encrypted score to the parents’ phone that can decrypt it locally. This allows the parents to benefit from a well-trained ML model that is enriched by the knowledge of other parents and experts without compromising the privacy of their children.

We present a proof-of-concept implementation of the app in this paper. The proposed ML model takes as input the app usage data with the granularity of different app categories and hours of usage. It also takes the sleep pattern of the child since this is considered a strong indicator of the mental health condition [TZW+19]. This data is encrypted with HE using the Microsoft SEAL library [SEA19]. While designing the ML model, we take into account both the precise calculation of the well-being score as well as its efficient execution through the SEAL library. In our evaluation, one inference requires ∼100 ms and deviates by ∼0.0002 from the inference result without encryption. In the current implementation, the training is performed on unencrypted data. Efficient training of any generic ML model on encrypted data is still an open problem. However, we outline a concrete methodology to train on encrypted data.

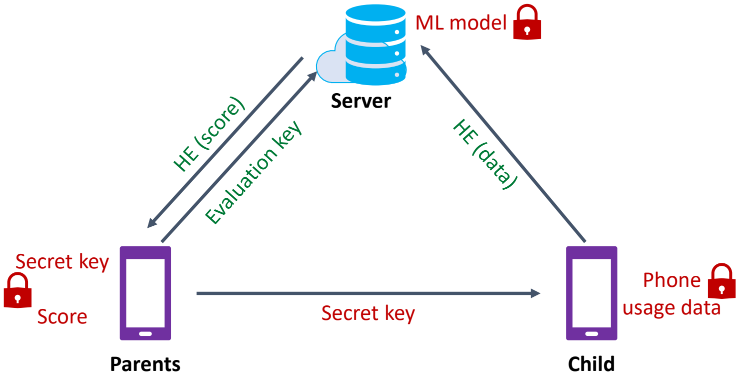

Figure 1: Privacy model

Privacy model. The privacy model of HappyKidz is illustrated in Figure 1. It involves three parties: the parent, the child, and the server. The parent generates the secret key and evaluation key for HE and sends the secret key to the child and the evaluation key to the server. The phone usage data is collected at the child's phone, encrypted with HE using the parent's secret key and sent to the server. This guarantees that the server cannot access the child's data. The server who holds the ML model uses the evaluation key to compute the well-being score. The result generated by the server is an encryption of the well-being score under the parent's secret key. This result is sent to the parent who uses the secret key to decrypt it and learn the well-being score of the child. We assume that the server does not collude with parents to release additional data about the child.

2 Proof of Concept Implementation

We implemented a proof-of-concept version of this app during the 2019 Microsoft Private AI Bootcamp[1]. This section describes the details of the implementation.

2.1 Data Selection and Features

Overall, the HappyKidz app aims to evaluate well-being by measuring quantitative behavioral indicators associated with mental health issues. The proof of

[1] https://www.microsoft.com/en-us/research/event/private-ai-bootcamp/

concept uses two commonly cited indicators: total time spent on phone apps and sleep patterns. Data from these behaviors are collected on the child's phone and consolidated into *features* that are used in the machine learning model.

Various studies have found a correlation between overall social media use and depressive symptoms [HGD$^+$19, TJRM18]. We define three categories of phone apps (social media, education, and games) and divide each day into three time-blocks (school hours, evening hours, and sleep hours). We aggregate the total time a child spends using apps in each category. This breakdown provides insight into appropriate phone usage. For example, a child is welcome to use social media during their evening free time, but excessive use while at school or during sleeping hours is less appropriate.

Sleep also plays a role in adolescent well-being. A variety of studies indicate links between sleep deprivation and behavioral problems in youth. Clarke and Harvey [CH12] suggest that improved sleep quality in adolescents with insomnia correlates with improved moods. We record the time that the child falls asleep and the total duration of sleep each night. This pair is stored locally for three days. Each day, we send the past three nights of sleep data to the model. This accommodates natural fluctuations in bedtimes (e.g. a child may stay up late one night to finish their homework) while still identifying longer-term patterns (e.g. a child goes to bed late every night).

These data provide 15 features each day: 9 from app usage and 6 from sleep data. The data are encrypted and uploaded to the cloud. For discussion of other potential data sources, see Section 3.1.

2.2 Learning Model

The app implements a model consisting of two fully connected (FC) layers. The output of the model is a wellness score between 0 and 1, where a higher score indicates positive behavioral indicators and thus good mental health[2]. In the proof of concept, we trained the model on a simulated feature vector (described in Section 2.1) with hand-labeled wellness scores.

Formally, our model is described as the following function, which takes the input feature vector x of length n:

$$f(x) = b_2 + W_2(s(b_1 + W_1 x)). \tag{1}$$

In this function, $b_1 \in \mathbb{R}^n$, $b_2 \in \mathbb{R}$ are bias vectors, $W_1 \in \mathbb{R}^{n \times n}$ and $W_2 \in \mathbb{R}^{1 \times n}$ are weight matrices, and $s : \mathbb{R}^n \to \mathbb{R}^n$ is the activation function (which operates element-wise on a vector). The bias vectors and weight matrices are generated during training.

We define the activation function s as the square function. It provides high inference accuracy for low-depth ML models [DGBL$^+$16] and is efficient to calculate under homomorphic encryption. Given our two-layer model, this is the most suitable option.

[2] In the parent's app, we will color-code the wellness score for easy interpretation. High scores will be green, low scores will be red.

2.3 Microsoft SEAL Implementation

We implemented the neural net described in Section 2.2 using the Microsoft SEAL [SEA19] homomorphic encryption library. The library supports several protocols and data representations; we used the CKKS scheme [CKKS17] with a multiplicative depth of three.

As described in Equation 1, the three main operations are a matrix-vector multiplication ($W_1 x$), a square activation function (s), and an inner product ($W_2 s(\cdot)$). The matrix-vector multiplication uses the diagonal method introduced by Halevi and Shoup [HS14]. The square activation function can be computed using a square-in-place homomorphic multiplication. The inner product operation is optimized to use only $O(\log N)$ rotations.

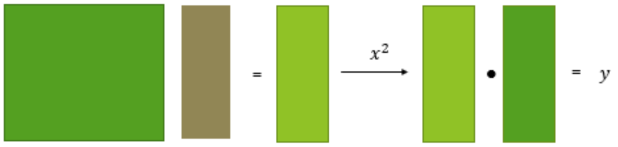

Figure 2: Schema of the inference SEAL implementation

Our model is stored in plaintext as a weight matrix W_1 and a weight vector W_2. When the server receives a batched encoding CKKS ciphertext x, it computes the matrix-vector product $W_1 \cdot x$. To make this more efficient, two preprocessing steps are done. One, on the server-side the diagonals of W_1 are encoded as plaintext vectors. Second, on the child's device, the ciphertext x has the features repeated to fill all the slots. The diagonal method for the matrix-vector multiplication requires us to be able to rotate the slots of a ciphertext. In our implementation, we introduce some temporary ciphertext so that we can get to all the necessary rotations by only rotating one position each time. We can then request just this rotation in the Galois key.

```
// perform the multiplication
Ciphertext temp, temp2;
Ciphertext enc_result;
temp2 = ct; // ct = x

for (int i =0; i < dimension ; i++){
    temp = temp2;
    // multiply
    evaluator.multiply_plain_inplace(temp, ptxt_diag[i]);
    if (i == 0){
        enc_result = temp;
    } else{
        evaluator.add_inplace(enc_result, temp);
    }
```

```
        evaluator.rotate_vector(temp2, 1, galk, temp2);
    }
    evaluator.rescale_to_next_inplace(enc_result);
    enc_result.scale() = pow(2.0, my_scale);
```

Next, we add the bias vector b_1. The activation function $s(x) = x^2$ can be applied by squaring the ciphertext in place followed by relinearization and rescaling.

```
    //add bias vector b1
    encoder.encode(b1, enc_result.parms_id(),scale, b1_plaintext);
    evaluator.add_plain_inplace(enc_result,b1_plaintext);

    //square in place
    evaluator.square(enc_result, enc_result);
    evaluator.relinearize_inplace(enc_result, relin_keys);
    evaluator.rescale_to_next_inplace(enc_result);
    enc_result.scale() = pow(2.0, my_scale);
```

Next, the inner product with the weight vector W_2 can be done by first performing a component wise multiplication and then summing the slots. To do this we need to rotate the ciphertext by powers of 2 rotations; we request these specific Galois keys be created. We follow this up with adding in the final bias correction value b_2.

```
    //multiply in place
    evaluator.multiply_plain_inplace(enc_result, W2);

    // Sum the slots
    Ciphertext temp_ct;
    for (size_t i = 1; i <= encoder.slot_count() / 2; i <<= 1) {
        evaluator.rotate_vector(enc_result, i, galk, temp_ct);
        evaluator.add_inplace(enc_result, temp_ct);
    }

    // add bias value b2
    encoder.encode(b2,enc_result.parms_id(), enc_result.scale(),
         b2_plaintext);
    evaluator.add_plain_inplace(enc_result,b2_plaintext);
```

In the interest of performance, we tried to minimize the size of the CKKS parameters. We chose a polynomial of degree 8192 and a ciphertext modulus with prime factors of sizes $\{60, 30, 30, 60\}$. The communication sizes of the ciphertexts are in Table 1. There are two ways to encrypt the data on the child's device, either using a secret key that is shared with the parent's device or with a public key that corresponds to the secret key on the parent's device.

Encrypting with the secret key saves about a factor of 2 in ciphertext size.

Client to server (encrypting with secret key) (feature vector)	144 KB
Client to server (encrypting with public key) (feature vector)	288 KB
Server to client (wellness score)	130 KB

Table 1: Ciphertext sizes

Since the server needs to compute rotations, it will need a set of Galois keys. We generated the smallest set of Galois keys necessary, which includes rotations $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. The total size of these keys is 7.5 MB. Since the server will also be performing relinearization as part of the homomorphic computation, it will need relinearization keys, which have size 627 KB. The *evaluation key* is the name given to all the key material (relinearization keys and Galois keys) that is needed for the server to run the homomorphic computation. In total the evaluation key has size 8.1 MB but is only generated by the parent's device and sent to the server in the initial setup.

The total execution time of the homomorphic circuit is around 100 ms. With these parameters, the floating-point approximation of CKKS gives us about 4 decimal digits of precision. The full code for our implementation can be found at `https://github.com/bmcase/bootcamp`.

3 Soundness and Future Work

Why use HE? When designing a HE application, one must compare against performing this computation locally without homomorphic encryption. We think there are a couple of reasons why using homomorphic encryption is the better option for our application. First, all the data storage can happen on the server and we can periodically send the parent long term statistical summaries of the data and also train other models to look for unhealthy trends in the long term data. Second, it may be the case that the app wants to keep the model from being easily taken by a competitor and charge a monthly subscription for the app. If this is the desire, then it is better to have the model computation done on the server. Keeping the model on the server also gives the app designers more flexibility in updating the model periodically.

How to keep malicious parties from corrupting the ML? One issue any ML application faces is collecting and maintaining accurate training data. There may be adversarial parties who wish to influence the outcome of the model: for example, a game developer may try to reclassify its products as education apps or train the model to associate higher wellness to children with increased game times.

In other use cases for machine learning with homomorphic encryption such as phishing or spam detection, the adversary has control over the target that the model is trying to identify (e.g. the phishing and spam emails), and in such a setting, it is necessary to continually retrain the model to stay up-to-date against modern threats. In our use case, healthy usage of social media, games, and educational apps does not change significantly on a short-term basis. We can train a single model via a large-scale study and use it for an extended time period without compromising its accuracy. Such a study should be done in collaboration with psychologists in settings where children already have devices (some schools have programs to provide students with devices).

The app also depends on secondary data sources, such as app classifications. This data is not used in the machine learning model but is necessary to featurize data or interpret the results. Since this is stored in the clear, we can issue updates to the child and parent apps (e.g. with new classification lists) to combat malicious behavior from app developers.

How to detect if the app is not functioning correctly? Another concern is how to incentivize correct usage by children. This app fails to be useful if, for example, the child has a secondary phone that they use for certain types of behavior. One mitigating factor is to provide high-level data for the parent, such as total time spent in each of three app categories (these statistics could also be computed using homomorphic encryption). If the parent is roughly aware of their child's typical phone usage, they should be able to identify cases where the app data doesn't correlate with the child's behavior patterns.

How to customize the app to irregular schedules? This app is designed to be useful for the average child, but many families have schedules that fall outside the norm. For example, home-schooled children may not have typical 9-3 school hours and varsity athletes may wake up early for team practice. One potential mitigating approach is to allow parents to define custom schedules. They can locally set expected hours for sleep, school, and evening/playtime. These are sent to the child's device and used to define the app usage features.

What is a *good* well-being score? Since the perception of a good well-being score may vary among parents, we do not define a concrete threshold between good and bad. Instead, we divide the scores into four ranges and color code the ranges as red, orange, yellow, and green where red indicates the worst and green indicate the best. Along with the absolute value, the changes in the well-being score is also an important indicator of the mental health of the child.

3.1 Future Work

The proof-of-concept app described in this paper is fairly limited. Future work includes producing higher-quality training data and expanding the machine learning models to provide more useful data.

Training data and features. We need to collect and accurately label real-world, representative data. In a commercial setting, we would collect data via a larger scientific study. Some telecom companies, including Sprint and TracFone, have partnered with public schools to provide free cell phones to students. We could work with such programs to install a preliminary app on the free phones that would collect training data. This study would have to partner with child and education psychologists or other trained professionals who could evaluate the students individually to assess their mental health and assign labels to the data. This type of study would also provide intuition to whether our two-layer model is appropriate for this setting.

Another option for collecting labeled training data is to work in partnership with parents and continually retrain the model. In this setting, we could have parents answer questions on the app regarding their child's well-being. The answers would provide new labels for their child's collected data. This approach has several issues.

- Training a model on encrypted data is prohibitively inefficient, so it would have to be done in the clear on the client-side. This might require the parent to use a more powerful device (e.g. a desktop computer) to answer questions and update the model.
- The server may be able to infer information about the client's data by comparing the model before and after an update. A typical mitigation is to send the model along a chain of parents (each of whom provides an update) before returning it to the server. However, this requires extensive communication and synchronization between individual users of the app.
- There are a variety of approaches for training a model in parallel, but these are not compatible with the privacy requirements of our application. The server would only be able to request updates from one parent (or chain of parents) at a time.
- This provides an avenue for parents to provide arbitrary or incorrect answers. We would have to compute server-side cross-validation after each retraining session to protect the model.

In the future, we may also wish to add more features to the app. For example, the SleepCycle [Sle20] app computes a "quality" score that correlates with the measured amount of deep or REM sleep. We also wish to incorporate more granular app categories or time blocks, or data beyond sleep and phone usage. The maximum ciphertext size in the CKKS implementation is much larger than our current input vector, so it is technically simple to add more features.

Expanded models. In the future, we would like to make longer-term evaluations about overall mental health. There are two potential approaches: We can store daily scores on the parent's phone and report monthly averages and trends. Alternately, we can store encrypted features on the cloud and train new models on the aggregates to produce long-range wellness scores. These would likely be more informative and less reactive than day-to-day snapshots.

The current app architecture provides flexibility to evaluate more complex models on the cloud. The simple two-layer network may not be appropriate for use on real data, but we can train and evaluate larger and more useful models that incorporate more data and advanced ML techniques.

4 Conclusion

We have presented the HappyKidz app that allows parents to monitor the well-being of their children through phone usage and sleep patterns. Contrary to the existing parental apps, which researchers have found to be unpopular among children, this app does not control or report phone usage of the children directly to the parents. Instead, it calculates a well-being score of the child using an ML model that is trained collaboratively by a large group of parents and experts and is deployed on a server. The app protects the privacy of the child's data by encrypting it with HE. Our proof-of-concept implementation shows that computation of the well-being score on encrypted data is practical in terms of computation time and memory usage.

References

[ADA] ADAA reviewed mental health apps. `https://adaa.org/finding-help/mobile-apps`. Accessed: 3 December 2019.

[BP05] Adriana Bianchi and James G Phillips. Psychological predictors of problem mobile phone use. *CyberPsychology & Behavior*, 8(1):39–51, 2005.

[CFD+19] Sarah J. Clark, Gary L. Freed, Sekhar Deepa, Dianne C. Singer, Acham Gebremariam, and Sara L. Schultz. Recognizing youth depression at home and school. *Mott Poll Report*, 35(2), November 2019.

[CH12] Greg Clarke and Allison G Harvey. The complex role of sleep in adolescent depression. *Child and Adolescent Psychiatric Clinics*, 21(2):385–400, 2012.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437. Springer, 2017.

[CRC+16] Yolanda Linda Reid Chassiakos, Jenny Radesky, Dimitri Christakis, Megan A Moreno, Corinn Cross, et al. Children and adolescents and digital media. *Pediatrics*, 138(5):e20162593, 2016.

[DGBL+16] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. Technical Report MSR-TR-2016-3, February 2016.

[GBUG$^+$18] Arup Kumar Ghosh, Karla Badillo-Urquiola, Shion Guha, Joseph J LaViola Jr, and Pamela J Wisniewski. Safety vs. surveillance: what children have to say about mobile apps for parental control. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[HBC17] Elizabeth Hoge, David Bickham, and Joanne Cantor. Digital media, anxiety, and depression in children. *Pediatrics*, 140(Supplement 2):S76–S80, 2017.

[HGD$^+$19] Taylor Heffer, Marie Good, Owen Daly, Elliott MacDonell, and Teena Willoughby. The longitudinal association between social-media use and depressive symptoms among adolescents and young adults: An empirical reply to twenge et al.(2018). *Clinical Psychological Science*, 7(3):462–470, 2019.

[HS14] Shai Halevi and Victor Shoup. Algorithms in HElib. In *Annual Cryptology Conference*, pages 554–571. Springer, 2014.

[KCY$^+$15] Minsam Ko, Seungwoo Choi, Subin Yang, Joonwon Lee, and Uichin Lee. Familync: facilitating participatory parental mediation of adolescents' smartphone use. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 867–878, 2015.

[LO18] Eun Jee Lee and Yolanda Ogbolu. Does parental control work with smartphone addiction?: A cross-sectional study of children in south korea. *Journal of addictions nursing*, 29(2):128–138, 2018.

[SEA19] Microsoft SEAL (release 3.4). `https://github.com/Microsoft/SEAL`, October 2019. Microsoft Research, Redmond, WA.

[Sle20] Sleep Cycle AB. Sleep cycle: Sleep analysis & smart alarm clock. `https://www.sleepcycle.com/`, 2020.

[SLMM$^+$14] Holly H Schiffrin, Miriam Liss, Haley Miles-McLean, Katherine A Geary, Mindy J Erchull, and Taryn Tashner. Helping or hovering? the effects of helicopter parenting on college students' well-being. *Journal of Child and Family Studies*, 23(3):548–557, 2014.

[SMO09] Mercedes Sánchez-Martínez and Angel Otero. Factors associated with cell phone use in adolescents in the community of madrid (spain). *CyberPsychology & Behavior*, 12(2):131–137, 2009.

[TJRM18] Jean M Twenge, Thomas E Joiner, Megan L Rogers, and Gabrielle N Martin. Increases in depressive symptoms, suicide-related outcomes, and suicide rates among us adolescents after 2010 and links to increased new media screen time. *Clinical Psychological Science*, 6(1):3–17, 2018.

[TZW+19] Fangbiao Tao, Liwei Zou, Xiaoyan Wu, Shuman Tao, Honglv Xu, Yang Xie, and Yajuan Yang. Mediating effect of sleep quality on the relationship between problematic mobile phone use and depressive symptoms in college students. *Frontiers in Psychiatry*, 10:822, 2019.

[YR98] Kimberly S Young and Robert C Rogers. The relationship between depression and internet addiction. *Cyberpsychology & behavior*, 1(1):25–28, 1998.